

KHOA KỸ THUẬT VÀ CÔNG NGHỆ  
BỘ MÔN CÔNG NGHỆ THÔNG TIN



BÁO CÁO MÔN HỌC  
CÔNG NGHỆ PHẦN MỀM

**TÌM HIỂU VÀ XÂY DỰNG HỆ THỐNG  
QUẢN LÝ THIẾT BỊ DẠY HỌC TẠI TRƯỜNG  
TRUNG HỌC PHỔ THÔNG**

Giảng viên giảng dạy: **Nguyễn Bảo Ân**

Sinh viên thực hiện:

Lớp: **DA21TTC**

**Nguyễn Hoàng Thương**      **110121224**

**Lê Thị Ngọc Hân**      **110121249**

**Nguyễn Khánh Băng**      **110121248**

**Trà Vinh, tháng 06 năm 2024**

## NHẬN XÉT CỦA GIẢNG VIÊN GIẢNG DẠY

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

*Trà Vinh, ngày ..... tháng ..... năm .....*

**Giảng viên giảng dạy**

## LỜI CẢM ƠN



Nhóm em xin gửi lời cảm ơn chân thành và sự tri ân sâu sắc đối với các Thầy/Cô của Trường Đại học Trà Vinh, đặc biệt là Thầy Nguyễn Bảo Ân đã tận tình truyền đạt kiến thức về môn hệ quản trị cơ sở dữ liệu. Với vốn kiến thức được tiếp thu trong quá trình học không chỉ là nền tảng cho quá trình nghiên cứu bài báo cáo mà nó còn là hành trang quý báu để nhóm em áp dụng vào thực tế một cách vững chắc và tự tin.

Chúng em cũng xin chân thành cảm ơn Thầy Nguyễn Bảo Ân là người hướng dẫn giúp đỡ và cung cấp những kiến thức quý báu giúp chúng em hoàn thành tốt bài báo cáo của nhóm mình.

Do còn hạn chế về kiến thức cũng như những kinh nghiệm thực tế cho nên không tránh khỏi được những sai sót trong quá trình tìm hiểu và trình bày rất mong nhận được sự đóng góp ý kiến của thầy để bài báo cáo được hoàn chỉnh hơn.

Sau cùng, nhóm em xin kính chúc Thầy Nguyễn Bảo Ân thật nhiều sức khỏe, niềm tin để tiếp tục thực hiện sứ mệnh cao đẹp của mình là truyền đạt kiến thức cho thế hệ mai sau.

Xin chân thành cảm ơn!

# MỤC LỤC



Nội dung	Trang
CHƯƠNG 1: GIỚI THIỆU.....	1
CHƯƠNG 2: CƠ SỞ LÝ THUYẾT .....	3
2.1 Agile.....	3
2.1.1 Định nghĩa Agile .....	3
2.1.2 Agile Values .....	3
2.1.3 Agile Principles .....	4
2.1.4 Phương pháp Agile.....	5
2.2 Scrum .....	9
2.2.1 Định nghĩa Scrum.....	9
2.2.2 Thuật ngữ trong Scrum.....	10
2.2.3 Các vai trò quan trọng trong Scrum .....	11
2.2.4 Scrum cycles.....	11
2.2.5 Product Backlog .....	12
2.2.6 Sprint .....	14
2.3 Công nghệ, kiến trúc phần mềm được sử dụng .....	15
2.3.1 Jira .....	15
2.3.2 Figma.....	20
2.3.3 Docker .....	21
CHƯƠNG 3: XÁC ĐỊNH NHU CẦU .....	24
3.1 Đặc tả mục tiêu dự án .....	24
3.2 Xác định Personas.....	24
3.3 Xác định các User Stories .....	24
3.4 Xác định các Product Backlog.....	25

3.5 Xác định các nhu cầu phi tính năng .....	26
CHƯƠNG 4: LẬP KẾ HOẠCH SCRUM .....	27
4.1 Sắp xếp các Product Backlog .....	27
4.2 Thiết kế các Sprint trên Jira .....	27
CHƯƠNG 5: HIỆN THỰC HÓA KẾ HOẠCH .....	30
5.1 Thiết kế giao diện trên Figma .....	30
5.1.1 Giao diện đăng nhập .....	30
5.1.2 Giao diện dashbroad .....	30
5.1.3 Giao diện danh sách thiết bị .....	31
5.1.4 Giao diện danh sách mượn thiết bị .....	31
5.1.5 Giao diện danh sách trả thiết bị .....	32
5.1.6 Giao diện danh sách nhân viên .....	32
5.1.7 Giao diện danh sách nhập thiết bị .....	33
5.1.8 Giao diện in phiếu nhập .....	33
5.2 Mã nguồn .....	34
5.3 Burndown Chart .....	36
5.3.1 Burndown Chart Sprint 1 .....	36
5.3.2 Burndown Chart Sprint 2 .....	36
5.3.3 Burndown Chart Sprint 3 .....	37
5.3.4 Burndown Chart Sprint 4 .....	37
CHƯƠNG 6: KẾT LUẬN .....	38
6.1 Kết quả đạt được .....	38
6.2 Hạn chế .....	38
6.3 Hướng phát triển .....	38
DANH MỤC TÀI LIỆU THAM KHẢO .....	39

## DANH MỤC HÌNH ẢNH



Hình 2.1 Tư duy Agile .....	3
Hình 2.2 Giá trị cốt lõi của Agile .....	4
Hình 2.3 Nguyên tắc của tuyên ngôn Agile .....	4
Hình 2.4 Phương pháp Agile.....	6
Hình 2.5 Phát triển tăng trưởng (Incremental development) .....	7
Hình 2.6 Extreme Programming practices .....	8
Hình 2.7 Chu trình Scrum .....	11
Hình 2.8 Sprint activities.....	15
Hình 2.9 Giao diện Jira.....	16
Hình 2.10 Giao diện Figma .....	20
Hình 2.11 Giao diện Docker .....	21
Hình 4.1 Sprint 1 .....	29
Hình 4.2 Sprint 2 .....	29
Hình 4.3 Sprint 3 .....	29
Hình 4.4 Sprint 4 .....	29
Hình 5.1 Giao diện đăng nhập.....	30
Hình 5.2 Giao diện dashbroad.....	30
Hình 5.3 Giao diện danh sách thiết bị .....	31
Hình 5.4 Giao diện danh sách mượn thiết bị.....	31
Hình 5.5 Giao diện danh sách trả thiết bị .....	32
Hình 5.6 Giao diện danh sách nhân viên.....	32
Hình 5.7 Giao diện nhập thiết bị .....	33
Hình 5.8 Giao diện in phiếu nhập .....	33
Hình 5.9 Các file mã nguồn được upload lên Github .....	34

Hình 5.10 Lịch sử commit Github .....	34
Hình 5.11 Chạy các giao diện trên Docker .....	35
Hình 5.12 Burndown Chart Sprint 1 .....	36
Hình 5.13 Burndown Chart Sprint 2 .....	36
Hình 5.14 Burndown Chart Sprint 3 .....	37
Hình 5.15 Burndown Chart Sprint 4 .....	37

## DANH MỤC BẢNG BIỂU



Bảng 1. 1 Bảng kế hoạch Scrum trên Jira .....	28
---	----



## CHƯƠNG 1: GIỚI THIỆU

Hệ thống quản lý thiết bị cho trường trung học phổ thông là một giải pháp phần mềm nhằm mục đích tổ chức và theo dõi tất cả các thiết bị của trường một cách hiệu quả. Hệ thống này cung cấp nhiều tính năng quản lý thiết bị, tình trạng thiết bị, và quản lý sử dụng thiết bị, giúp đảm bảo rằng các thiết bị luôn ở trạng thái tốt nhất và được sử dụng hợp lý.

Các chức năng chính của hệ thống như sau:

- **Quản lý tương tác người dùng:**
  - + Đăng nhập và đăng xuất: Hệ thống sẽ quản lý quá trình đăng nhập và đăng xuất của người dùng. Khi đăng nhập thành công, người dùng sẽ có thể truy cập các chức năng của hệ thống.
- **Quản lý thiết bị:**
  - + Cập nhật thông tin thiết bị: Người dùng có thể thêm mới, chỉnh sửa hoặc xóa thông tin thiết bị. Thông tin này bao gồm tên thiết bị, loại thiết bị, nhà cung cấp, tình trạng hiện tại của thiết bị,....
  - + Tìm kiếm thiết bị: Hệ thống cho phép người dùng tìm kiếm thiết bị theo tên thiết bị.
- **Quản lý tình trạng thiết bị:**
  - + Nhập thêm thiết bị: Hệ thống hỗ trợ nhập thêm thiết bị mới vào kho. Điều này bao gồm việc ghi nhận các thông tin liên quan đến thiết bị mới như tên, loại, số lượng, và ngày nhập.
  - + Lập phiếu nhập hàng: Khi có thiết bị mới được nhập về, hệ thống sẽ tạo lập các phiếu nhập hàng để lưu trữ thông tin nhập kho.
  - + Loại bỏ thiết bị hư hỏng: Hệ thống cũng cho phép loại bỏ các thiết bị không còn sử dụng được hoặc đã hư hỏng, đảm bảo rằng kho thiết bị luôn được cập nhật với các thiết bị hoạt động tốt.
- **Quản lý sử dụng thiết bị:**
  - + Lập danh sách mượn thiết bị: Người dùng có thể lập danh sách mượn thiết bị cho các mục đích sử dụng khác nhau. Danh sách này bao gồm thông tin về người mượn, thiết bị được mượn, thời gian mượn,..v.v.

- + Lập danh sách trả thiết bị: Tương tự, khi thiết bị được trả lại, hệ thống sẽ cập nhật danh sách trả thiết bị, bao gồm thông tin về người trả, thời gian trả,...
- + Danh sách thiết bị chưa trả trong ngày: Hệ thống cung cấp danh sách các thiết bị chưa được trả trong ngày, giúp người quản lý dễ dàng theo dõi và nhắc nhở người mượn trả thiết bị đúng hạn.
- **Thông tin nhân viên quản lý thiết bị:**
  - + Cập nhật thông tin nhân viên: Hệ thống cho phép thêm mới, chỉnh sửa hoặc xóa thông tin nhân viên quản lý thiết bị. Thông tin này bao gồm tên, ngày sinh, giới tính,...
  - + Loại bỏ thông tin nhân viên: Khi nhân viên hết hạn hợp đồng hoặc bị sa thải, hệ thống sẽ loại bỏ thông tin của họ ra khỏi cơ sở dữ liệu.
- **Báo cáo thống kê:**
  - + Lập báo cáo: Hệ thống sẽ hỗ trợ nhân viên lập các báo cáo thống kê liên quan đến thiết bị và tình trạng thiết bị. Các báo cáo này có thể bao gồm số lượng thiết bị, tình trạng sử dụng, và các vấn đề phát sinh trong quá trình sử dụng.
- **Quyền hạn người dùng:**
  - + Nhân viên: Nhân viên có quyền đăng nhập vào hệ thống để thực hiện các chức năng như thêm, xóa, sửa thông tin thiết bị và lập báo cáo thống kê. Họ cũng có thể đăng xuất khỏi hệ thống sau khi hoàn thành công việc.

Hệ thống này không chỉ giúp quản lý hiệu quả các thiết bị mà còn giúp giảm thiểu rủi ro mất mát, hư hỏng thiết bị, và tăng cường việc sử dụng thiết bị một cách hợp lý trong trường học.

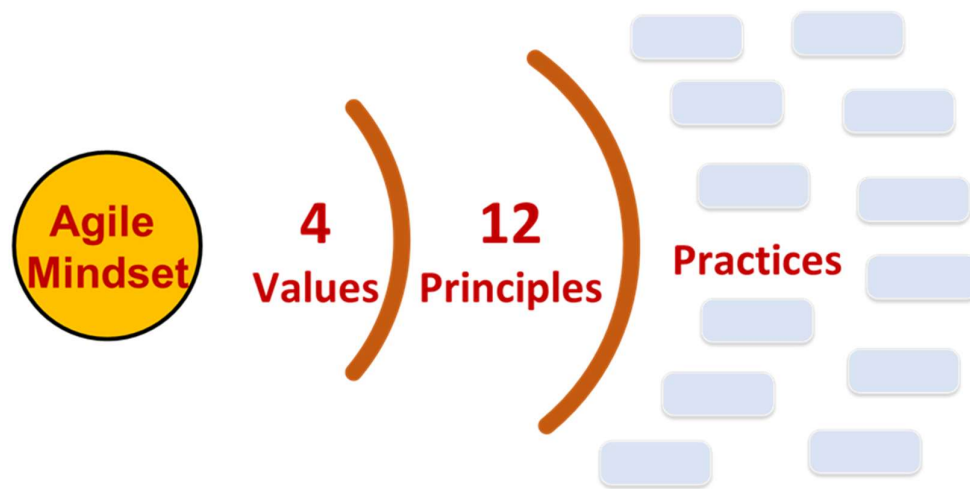
## CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

### 2.1 Agile

#### 2.1.1 Định nghĩa Agile

Agile Software Development là một phương thức phát triển phần mềm linh hoạt. Mục tiêu của Agile là đưa sản phẩm đến tay người dùng càng nhanh càng tốt. Agile không chỉ là một phương pháp luận mà còn là một triết lý dựa trên các nguyên tắc phân đoạn vòng lặp (iterative) và tăng trưởng (incremental). Hiện nay, Agile đã trở thành một phương thức quản lý dự án phổ biến với nhiều đại diện được gọi là các phương pháp “họ Agile”.

Agile là một cách tư duy được định nghĩa bởi 4 giá trị và được hướng dẫn bởi 12 nguyên tắc. Những giá trị và nguyên tắc này được biểu hiện qua nhiều thực hành khác nhau. Các Agile practitioners (người thực hành Agile) chọn phương pháp dựa trên nhu cầu cụ thể của họ.



Hình 2.1 Tư duy Agile

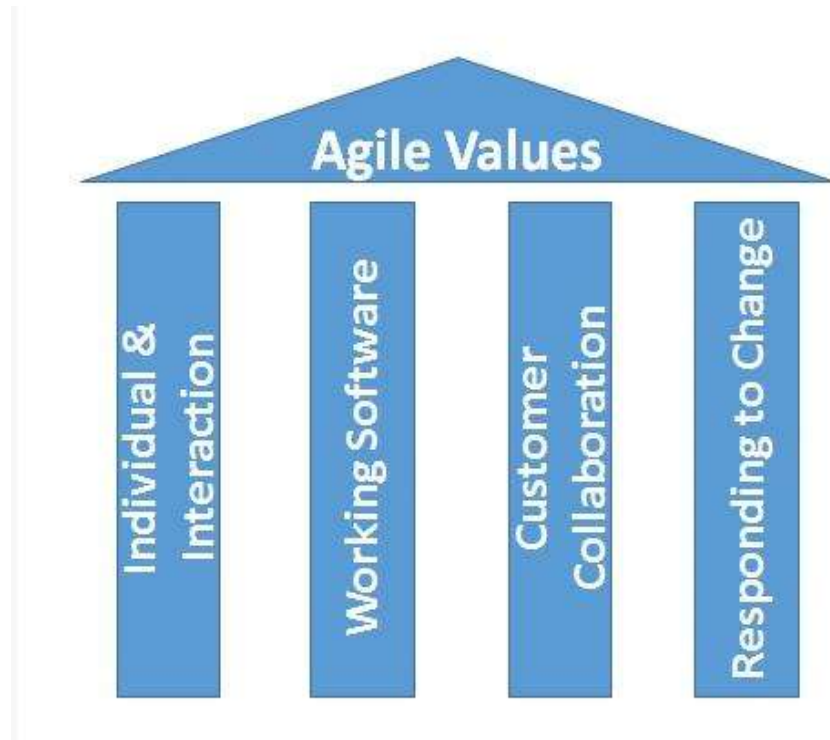
#### 2.1.2 Agile Values

❖ Agile có 4 giá trị cốt lõi:

- **Individuals and Interactions Over Processes and Tools**
  - Cá nhân và tương tác hơn là quy trình và công cụ.
- **Working Software Over Comprehensive Documentation**
  - Phần mềm hoạt động tốt hơn là tài liệu đầy đủ.
- **Customer Collaboration Over Contract Negotiation**
  - Hợp tác với khách hàng hơn là đàm phán hợp đồng.

- **Responding to Change Over Following a Plan**

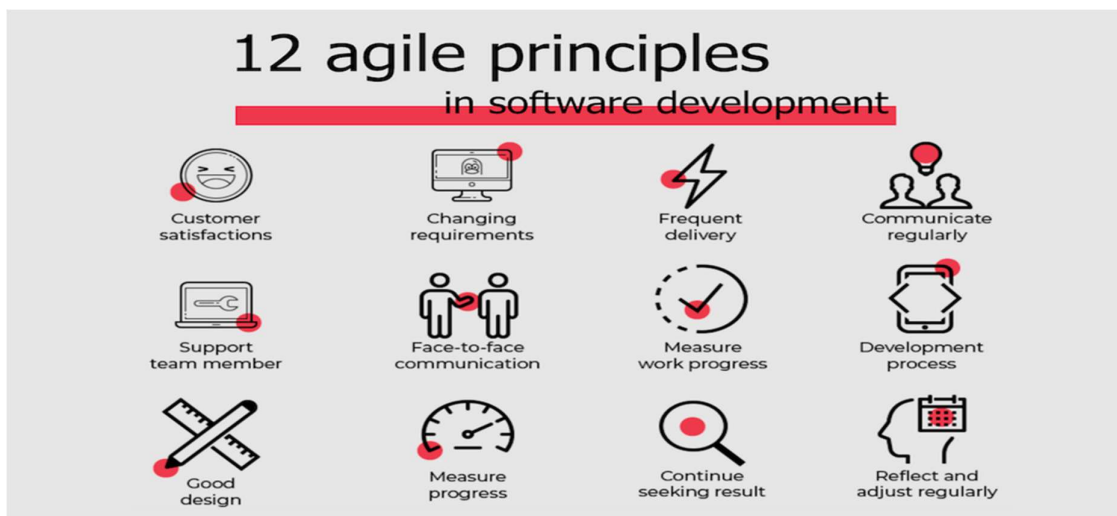
- Ứng phó, phản hồi với các thay đổi hơn là làm theo kế hoạch.



Hình 2.2 Giá trị cốt lõi của Agile

Những giá trị này nhấn mạnh sự linh hoạt, thích ứng, ưu tiên con người và kết quả thực tế hơn là các quy trình cứng nhắc và tài liệu chi tiết. Agile mindset thúc đẩy môi trường làm việc mà trong đó sự hợp tác và khả năng thích ứng với thay đổi được coi trọng để mang lại giá trị cao nhất cho khách hàng và các bên liên quan.

### 2.1.3 Agile Principles



Hình 2.3 Nguyên tắc của tuyên ngôn Agile

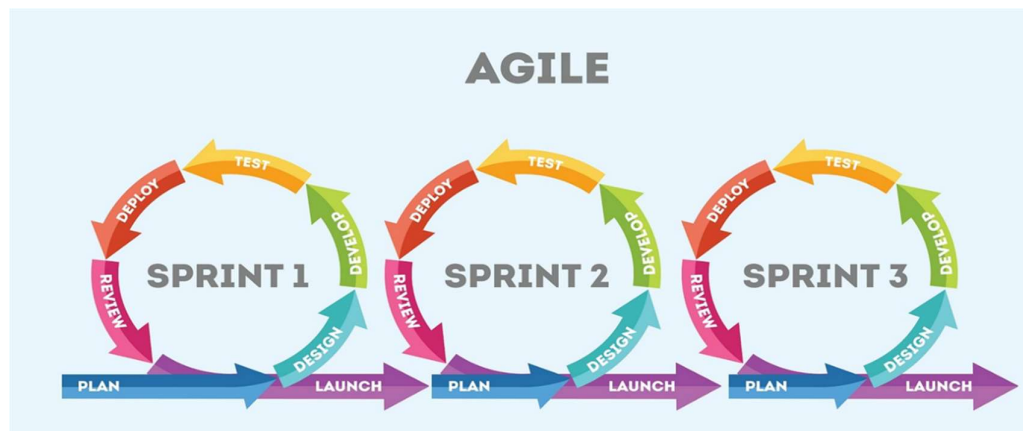
- **Customer satisfaction:** Ưu tiên sự hài lòng của khách hàng thông qua việc trao đổi và bàn giao liên tục, khách hàng sẽ cảm thấy hài lòng khi họ nhận được sản phẩm làm việc đều đặn thay vì chờ đợi thời gian kéo dài giữa các lần releases.
- **Changing requirements:** Chào đón việc thay đổi trong suốt quá trình phát triển. Tránh sự chậm trễ khi có yêu cầu thay đổi requirements.
- **Frequent delivery:** Chuyển giao sản phẩm thường xuyên. Chuyển giao phần mềm sử dụng được định kỳ, từ một vài tuần đến một vài tháng, với một thời gian ngắn hơn.
- **Communicate regularly:** Khách hàng và đội phát triển phải làm việc cùng nhau hàng ngày trong suốt dự án.
- **Support team member:** Xây dựng các dự án xoay quanh các cá nhân có động lực. Tạo cho họ một môi trường và hỗ trợ họ những thứ cần thiết và tin tưởng họ để công việc được hoàn thành.
- **Face-to-face communication:** Thúc đẩy các cuộc trò chuyện trực tiếp
- **Measure work progress:** Phần mềm làm việc được là thước đo của quá trình.
- **Development progress:** Các quy trình Agile thúc đẩy sự phát triển bền vững. Các nhà tài trợ cho dự án, các nhà phát triển và người dùng cuối có thể duy trì một tốc độ vô hạn định.
- **Good design:** Liên tục quan tâm đến kỹ thuật xuất sắc và thiết kế tốt giúp nâng cao tính linh hoạt.
- **Measure progress:** Tính đơn giản – nghệ thuật tối đa hoá khối lượng công việc chưa hoàn thành – là điều thiết yếu.
- **Continue seeking result:** Các kiến trúc, yêu cầu và thiết kế tốt nhất xuất hiện từ các nhóm tự tổ chức.
- **Reflect and adjust regularly:** Ở thời điểm kết thúc sprint, các team nên xem lại làm thế nào để hiệu quả hơn, sau đó cùng tự cải thiện ứng xử, cải tiến quy trình, kỹ thuật của mình sao cho phù hợp.

## 2.1.4 Phương pháp Agile

Phát triển dựa trên kế hoạch (plan-driven development) là một phương pháp phát triển phần mềm được sử dụng để hỗ trợ các hệ thống lớn, có tuổi thọ cao. Phương pháp này dựa trên các quy trình phát triển nghiêm ngặt và được kiểm soát chặt chẽ, bao gồm lập kế hoạch dự án chi tiết, xác định yêu cầu và mô hình hóa hệ thống.

Mặc dù phát triển dựa trên kế hoạch đảm bảo sự ổn định và tính năng của hệ thống, nhưng nó cũng có một số hạn chế. Phương pháp này đòi hỏi khoản dự phòng lớn và tài liệu nghiêm ngặt, do đó không hỗ trợ phát triển và chuyển giao phần mềm nhanh.

Để giải quyết những hạn chế này, các phương pháp Agile đã được phát triển vào những năm 1990. Các phương pháp Agile tập trung vào phần mềm hơn là tài liệu, phát triển theo chuỗi gia tăng và nhằm mục đích giảm thiểu tình trạng quan liêu của quy trình càng nhiều càng tốt.



Hình 2.4 Phương pháp Agile

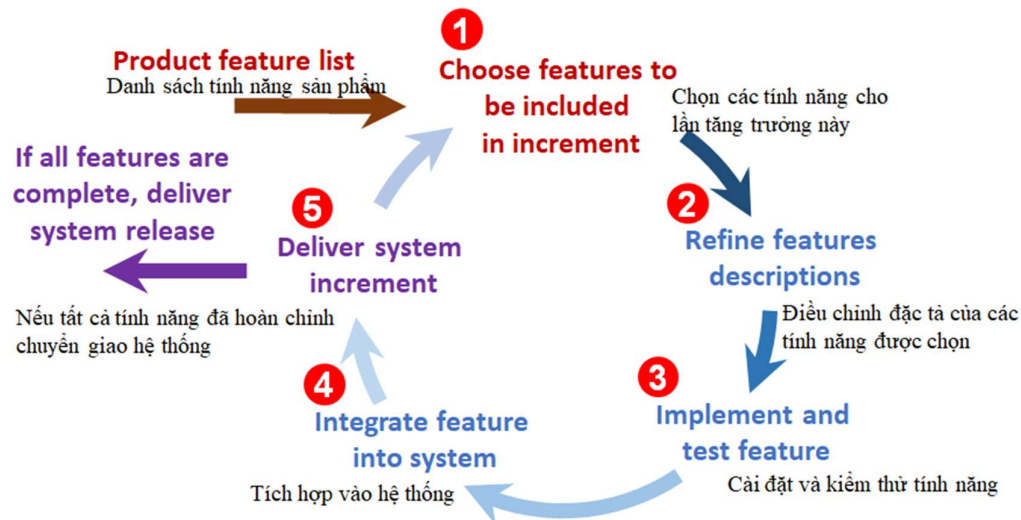
Tất cả các phương pháp linh hoạt đều dựa trên sự phát triển và phân phối tăng dần. Cách tốt nhất để hiểu sự phát triển gia tăng là coi sản phẩm phần mềm như một tập hợp các tính năng. Mỗi tính năng đều có tác dụng gì đó đối với người dùng phần mềm. Có thể có tính năng cho phép nhập dữ liệu, tính năng tìm kiếm dữ liệu đã nhập và tính năng định dạng và hiển thị dữ liệu. Mỗi phiên bản phần mềm sẽ triển khai một số lượng nhỏ các tính năng của sản phẩm.

#### ❖ Phát triển tăng trưởng (Incremental development)

Tất cả các phương pháp Agile đều xoay quanh các vòng lặp phát triển và chuyển giao. Quá trình phát triển sản phẩm tập trung vào các tính năng phần mềm, những tính năng này phục vụ cho nhu cầu của người dùng.

Trong phát triển gia tăng, chúng ta bắt đầu bằng việc xác định độ ưu tiên của các tính năng, từ đó triển khai các tính năng quan trọng nhất trước. Chi tiết của mỗi tính năng chỉ được định nghĩa trong từng vòng lặp phát triển cụ thể.

Sau khi tính năng được phát triển và chuyển giao, người dùng hoặc đại diện của họ có thể dùng thử và cung cấp phản hồi cho nhóm phát triển. Sau đó, chúng ta tiếp tục xác định và triển khai tính năng tiếp theo của hệ thống.



Hình 2.5 Phát triển tăng trưởng (Incremental development)

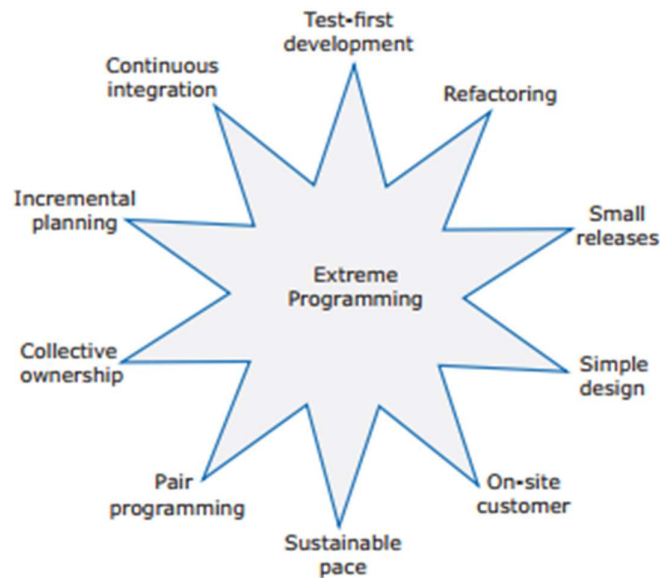
### ❖ Extreme programming

Extreme Programming (XP) là một trong những phương pháp phát triển phần mềm có ảnh hưởng lớn nhất, đã tạo ra sự thay đổi đáng kể trong văn hóa phát triển phần mềm. Phương pháp này được sáng lập bởi Kent Beck vào năm 1998. XP phát triển bằng cách thúc đẩy các thực tiễn tốt đã được công nhận, như phát triển lặp lại, và đẩy chúng lên mức 'cực đoan' để đạt được hiệu quả tối đa.

XP tập trung vào 12 kỹ thuật phát triển mới, mỗi kỹ thuật đều hướng đến việc phát triển, thay đổi và phân phối phần mềm một cách nhanh chóng và gia tăng. Những kỹ thuật này bao gồm lập trình cặp (pair programming), kiểm thử đơn vị liên tục (continuous unit testing), và tích hợp liên tục (continuous integration). Các kỹ thuật này giúp đảm bảo rằng phần mềm luôn ở trạng thái sẵn sàng và chất lượng cao.

Mục tiêu chính của XP là phát triển phần mềm với chất lượng cao nhất, đồng thời giảm thiểu chi phí, lỗi phần mềm và tăng cường năng suất. XP hướng đến việc tối đa hóa lợi nhuận đầu tư, điều này được thể hiện rõ ràng qua tên gọi "Extreme Programming" - tức là lập trình cực đoan.

XP đạt được mục tiêu này thông qua một tập hợp giá trị định hướng (values), nguyên tắc (principles) và kỹ thuật thực hành (Practices) đặc thù, tất cả đều theo định hướng của triết lý Agile. Các giá trị định hướng bao gồm giao tiếp, đơn giản, phản hồi, can đảm và tôn trọng. Những nguyên tắc cơ bản của XP giúp định hình các kỹ thuật thực hành, đảm bảo rằng mọi thành viên trong nhóm đều hiểu rõ và tuân thủ các phương pháp này để đạt được hiệu quả cao nhất.



*Hình 2.6 Extreme Programming practices*

- **Những kỹ thuật thực hành XP thường dùng:**
  - **Small releases (Chuyển giao nhỏ)**
    - Những chức năng hữu ích tối thiểu cung cấp giá trị kinh doanh được phát triển trước tiên.
    - Các bản phát hành của hệ thống diễn ra thường xuyên và tăng dần chức năng cho bản phát hành trước đó.
  - **Test-driven development (Phát triển hướng-kiểm thử)**
    - Thay vì viết mã sau đó kiểm tra mã đó, các nhà phát triển viết các đoạn mã kiểm thử (test code) trước.
    - Điều này giúp làm rõ mã thực sự nên làm gì và luôn có sẵn phiên bản “tested code”.
    - Unit-test cần được chạy để kiểm thử sau mỗi thay đổi.
    - Mã mới không nên 'phá vỡ' mã đã được cài đặt.



- **Continuous integration (Tích hợp liên tục)**

- Ngay sau khi một task được hoàn thành, nó sẽ được tích hợp vào hệ thống một phiên bản mới của hệ thống được tạo.
- Tất cả các bài unit-test từ tất cả các developers đều được chạy tự động và phải thành công trước khi phiên bản mới của hệ thống được chấp nhận.

- **Refactoring**

- Refactoring: cải tiến cấu trúc, trình bày, tính hiệu quả, bảo mật của một chương trình mà không làm thay đổi hành vi bên ngoài hệ thống.
- Mọi developers được yêu cầu phải refactor code ngay khi một yêu cầu cải tiến xuất hiện.
- Việc này giúp code đơn giản và dễ dàng bảo trì.

## **2.2 Scrum**

### **2.2.1 Định nghĩa Scrum**

Scrum là một phương pháp Agile tập trung vào lập kế hoạch và quản lý linh hoạt. Không giống như XP, nó không xác định các thực hành kỹ thuật sẽ được sử dụng. Nhóm phát triển có thể sử dụng bất kỳ phương pháp kỹ thuật nào mà họ cho là phù hợp với sản phẩm đang được phát triển.

Scrum là một phương pháp Agile cung cấp một khuôn khổ linh hoạt cho việc tổ chức và lập kế hoạch dự án. Không giống như Plan-driven development, Scrum cho phép điều chỉnh liên tục các kế hoạch dựa trên phản hồi thực tế và điều kiện thay đổi. Scrum không bắt buộc bất kỳ thực hành kỹ thuật cụ thể nào, thay vào đó, nó tập trung vào việc thiết lập các vai trò rõ ràng, các sự kiện định kỳ, và các tạo tác cụ thể để giúp nhóm phát triển phần mềm hoạt động hiệu quả hơn.

Trong Scrum, dự án được chia thành các chu kỳ ngắn gọi là "sprints", mỗi sprint thường kéo dài từ 2 đến 4 tuần. Cuối mỗi sprint, nhóm phát triển sẽ tạo ra một sản phẩm phần mềm có thể hoạt động được, và sẵn sàng để nhận phản hồi từ khách hàng hoặc các bên liên quan. Điều này giúp đảm bảo rằng sản phẩm phát triển luôn phù hợp với nhu cầu thực tế và có thể điều chỉnh kịp thời trước khi chuyển giao ra thị trường.

Scrum cung cấp sự minh bạch qua các cuộc họp hàng ngày (Daily Scrum), các cuộc họp lập kế hoạch (Sprint Planning), các buổi đánh giá sprint (Sprint Review), và các buổi hồi kết sprint (Sprint Retrospective). Những cuộc họp này giúp nhóm phát

triển duy trì sự tập trung, xác định các vấn đề kịp thời, và cải thiện quy trình làm việc liên tục.

Nhờ vào sự linh hoạt và khả năng thích ứng cao, Scrum trở thành một công cụ mạnh mẽ cho việc quản lý dự án phần mềm trong môi trường phát triển nhanh chóng và biến động. Nó không chỉ giúp quản lý có được các thông tin quan trọng về tiến độ và chi phí, mà còn đảm bảo sản phẩm cuối cùng đáp ứng tốt nhất yêu cầu của thị trường và khách hàng.

### 2.2.2 Thuật ngữ trong Scrum

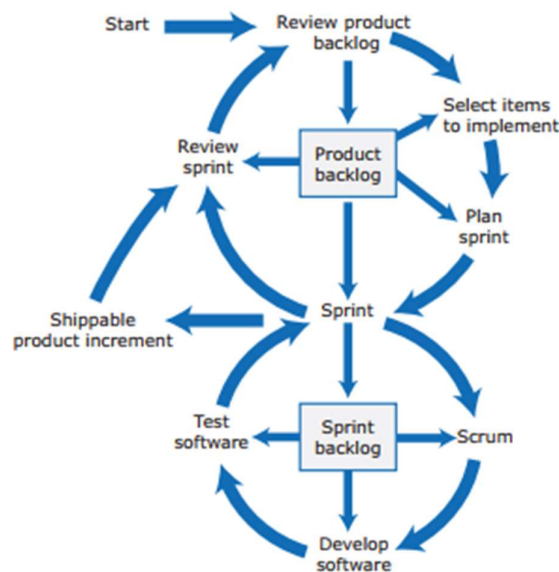
- **Product Backlog:** Danh sách công việc cần làm bao gồm các hạng mục như lỗi, tính năng và cải tiến sản phẩm mà nhóm Scrum chưa hoàn thành.
- **Sprint:** Một khoảng thời gian ngắn, thường từ hai đến bốn tuần, khi phát triển phần gia tăng của sản phẩm.
- **Daily Scrum:** Cuộc họp nhóm hàng ngày trong đó tiến độ được xem xét và công việc phải hoàn thành vào ngày hôm đó được thảo luận và thống nhất.
- **ScrumMaster:** Một huấn luyện viên nhóm, hướng dẫn nhóm sử dụng Scrum một cách hiệu quả.
- **Product Owner:** Một thành viên trong nhóm chịu trách nhiệm xác định các tính năng và thuộc tính của sản phẩm. Họ xem xét công việc đã hoàn thành và giúp kiểm tra sản phẩm.
- **Product:** Sản phẩm phần mềm đang được phát triển bởi nhóm Scrum.
- **Development Team:** Một nhóm nhỏ tự tổ chức từ năm đến tám người chịu trách nhiệm phát triển sản phẩm.
- **Potentially Shippable Product Increment:** Đầu ra của một lần chạy nước rút có chất lượng đủ cao để triển khai cho khách hàng sử dụng. Đây là một phiên bản của sản phẩm đã được hoàn thiện đến mức có thể được chuyển giao cho người dùng mà không cần thêm bất kỳ công việc nào khác.
- **Velocity (vận tốc):** Ước tính khối lượng công việc mà một nhóm có thể thực hiện trong một lần chạy nước rút. Velocity được tính bằng tổng số điểm của các user story hoặc công việc đã hoàn thành trong một sprint. Velocity được sử dụng để dự đoán khối lượng công việc có thể thực hiện trong các sprint tương lai và điều chỉnh kế hoạch dự án.

### 2.2.3 Các vai trò quan trọng trong Scrum

- Product Owner: Chịu trách nhiệm đảm bảo rằng nhóm phát triển luôn tập trung vào sản phẩm mà họ đang xây dựng thay vì chuyển hướng sang công việc thú vị về mặt kỹ thuật nhưng ít liên quan hơn. Trong quá trình phát triển sản phẩm, người quản lý sản phẩm thường đảm nhận vai trò Chủ sở hữu sản phẩm.
- Scrum Master: Là một chuyên gia Scrum có nhiệm vụ hướng dẫn nhóm sử dụng hiệu quả phương pháp Scrum. ScrumMaster không phải là người quản lý dự án thông thường mà là huấn luyện viên cho nhóm. Họ có thẩm quyền trong nhóm về cách sử dụng Scrum. Tuy nhiên, trong một số công ty sử dụng Scrum, ScrumMaster cũng có thể có một số trách nhiệm quản lý dự án.

### 2.2.4 Scrum cycles

Quy trình Scrum hoặc chu trình chạy nước rút được thể hiện trong Hình 2.8. Ý tưởng cơ bản của quy trình Scrum là phần mềm nên được phát triển theo một loạt “chạy nước rút”. Chạy nước rút là một hoạt động có thời lượng cố định (có khung thời gian), mỗi lần chạy nước rút thường kéo dài từ hai đến bốn tuần. Trong suốt một sprint, nhóm họp hàng ngày. (Scrum) để xem xét công việc đã hoàn thành cho đến nay và thống nhất về các hoạt động của ngày hôm đó. “Sprint backlog” được sử dụng để theo dõi công việc sẽ được thực hiện trong sprint đó.



Hình 2.7 Chu trình Scrum

- **Lập kế hoạch cho sprint:**
  - + Các hạng mục được chọn để thực hiện trong sprint tiếp theo.
  - + Các thành viên trong nhóm làm việc cùng nhau để lập kế hoạch cho sprint.
  - + Phân tích các mục đã chọn để tạo ra sprint backlog (danh sách các hoạt động sẽ hoàn thành trong sprint).
- **Triển khai sprint:**
  - + Nhóm triển khai nhiều mục tồn đọng của sprint nhất có thể trong khoảng thời gian cố định của sprint.
  - + Các hạng mục chưa hoàn thiện sẽ được trả lại vào danh sách tồn đọng của sản phẩm.
  - + Sprint không bao giờ được kéo dài để hoàn thành một hạng mục chưa hoàn thành.
- **Kết quả của sprint:**
  - + Sprint tạo ra mức tăng sản phẩm có thể chuyển giao được và có thể giao cho khách hàng hoặc bàn giao nội bộ.
  - + Các sản phẩm nội bộ, chẳng hạn như nguyên mẫu sản phẩm hoặc thiết kế kiến trúc, cung cấp thông tin cho các sprint trong tương lai.
  - + Nếu kết quả của sprint là một phần của sản phẩm cuối cùng thì nó phải hoàn thành.
  - + Trừ khi cần thay đổi chức năng phần mềm, nhóm sẽ không phải thực hiện thêm bất kỳ công việc nào trên phần mềm đó trong các sprint tương lai.
- **Cuộc họp đánh giá sau sprint:**
  - + Sau khi hoàn thành sprint, một cuộc họp đánh giá sẽ được tổ chức với sự tham gia của tất cả các thành viên trong nhóm.
  - + Nhóm thảo luận về những gì đã diễn ra tốt đẹp trong suốt sprint, những vấn đề nào phát sinh và cách giải quyết các vấn đề.
  - + Các thành viên trong nhóm cũng phản ánh về tính hiệu quả của các công cụ và phương pháp được sử dụng.
  - + Mục đích của cuộc họp này là để nhóm học hỏi lẫn nhau nhằm tránh các vấn đề và cải thiện năng suất trong các sprint sau này.

## 2.2.5 Product Backlog

Product Backlog là danh sách các công việc cần làm để hoàn thành quá trình phát triển sản phẩm. Danh sách này chứa các hạng mục tồn đọng của sản phẩm (Product Backlog Items - PBI). Các hạng mục này bao gồm:

- Tính năng sản phẩm cần triển khai: Các tính năng cụ thể mà sản phẩm cần có để đáp ứng yêu cầu của người dùng và khách hàng.
- Yêu cầu của người dùng: Các nhu cầu và mong muốn của người dùng cuối, những gì họ mong đợi từ sản phẩm.
- Hoạt động phát triển thiết yếu: Các công việc cần thiết để đảm bảo sản phẩm hoạt động đúng và hiệu quả, như viết mã, kiểm thử, tài liệu hóa.
- Cải tiến kỹ thuật mong muốn: Các nâng cấp và cải tiến nhằm cải thiện hiệu suất, bảo mật, hoặc khả năng bảo trì của sản phẩm.

➤ Ví dụ: Product Backlog Items (PBIs)

- Là một giáo viên, tôi muốn có thể định cấu hình nhóm công cụ có sẵn cho từng lớp học. (tính năng).
- Là phụ huynh, tôi muốn có thể xem bài làm của con mình và các đánh giá do giáo viên của chúng thực hiện. (tính năng).
- Là một giáo viên dạy trẻ nhỏ, tôi muốn có một giao diện hình ảnh dành cho những trẻ có khả năng đọc hạn chế. (yêu cầu người dùng).
- Xây dựng tiêu chí đánh giá phần mềm nguồn mở làm cơ sở cho các bộ phận của hệ thống này. (hoạt động phát triển).
- Tái cấu trúc mã giao diện người dùng để cải thiện tính dễ hiểu và hiệu suất. (cải tiến kỹ thuật).
- Thực hiện mã hóa tất cả dữ liệu cá nhân của người dùng (cải tiến kỹ thuật).

❖ **Trạng thái của Product Backlog Item states**

Trong quy trình quản lý Product Backlog trong Scrum, các hạng mục (PBI) thường được phân loại theo mức độ sẵn sàng khác nhau trước khi được đưa vào Sprint. Dưới đây là các trạng thái khác nhau của PBI:

- **Sẵn sàng để xem xét:**
  - Mô tả: Đây là các ý tưởng cấp cao và mô tả tính năng sơ bộ. Các hạng mục này có thể được xem xét để đưa vào sản phẩm, nhưng vẫn đang ở giai đoạn dự kiến.

- Đặc điểm: Có thể thay đổi hoàn toàn hoặc không được đưa vào sản phẩm cuối cùng.
  - Ví dụ: Một tính năng mới được đề xuất bởi khách hàng hoặc một ý tưởng từ nhóm phát triển để cải thiện sản phẩm.
- **Sẵn sàng để tinh chỉnh:**
- Mô tả: Nhóm đã đồng ý rằng đây là một hạng mục quan trọng cần được triển khai trong quá trình phát triển hiện tại.
  - Đặc điểm: Có một định nghĩa hợp lý rõ ràng về những gì được yêu cầu, nhưng cần thêm công việc để hiểu và tinh chỉnh hạng mục.
  - Ví dụ: Một tính năng đã được xác định cụ thể về yêu cầu chức năng nhưng cần thêm phân tích để xác định chi tiết kỹ thuật và công việc cần thiết.
- **Sẵn sàng triển khai:**
- Mô tả: PBI đã có đủ thông tin chi tiết để nhóm ước tính nỗ lực liên quan và triển khai hạng mục.
  - Đặc điểm: Sự phụ thuộc vào các mục khác đã được xác định, và hạng mục có thể được đưa vào Sprint Planning.
  - Ví dụ: Một tính năng đã hoàn thành phân tích chi tiết, có đầy đủ tài liệu và không có rào cản kỹ thuật nào để triển khai.

## 2.2.6 Sprint

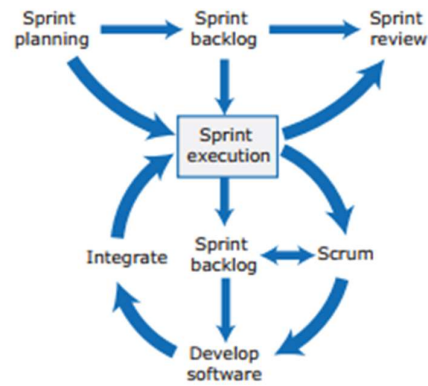
Sprint là các hoạt động có thời gian cố định (thường là 2–4 tuần) trong đó phần gia tăng của sản phẩm được phát triển. Các phần gia tăng phải 'có khả năng chuyển giao được'.

Trong quy trình Scrum, mỗi lần chạy nước rút (Sprint) bao gồm ba hoạt động cơ bản:

- **Lập kế hoạch Sprint:** Trong giai đoạn này, nhóm dự án chọn các mục công việc cần hoàn thành trong Sprint và nếu cần, sẽ tinh chỉnh chúng để tạo ra một Sprint Backlog. Quá trình lập kế hoạch này không nên kéo dài quá một ngày khi bắt đầu chạy nước rút.
- **Thực hiện Sprint:** Nhóm làm việc để triển khai các hạng mục tồn đọng của Sprint đã được chọn. Nếu không thể hoàn thành hết các hạng mục sprint backlog, thời gian cho sprint sẽ không được kéo dài. Thay vào đó, các hạng mục chưa hoàn thành

sẽ được trả lại vào danh sách tồn đọng của sản phẩm và được xếp hàng cho lần chạy nước rút trong tương lai.

- **Đánh giá Sprint:** Công việc được thực hiện trong Sprint được nhóm và (có thể) các bên liên quan bên ngoài xem xét. Nhóm phản ánh về những gì đã diễn ra tốt đẹp và những gì đã sai trong suốt sprint, nhằm cải thiện quy trình làm việc.



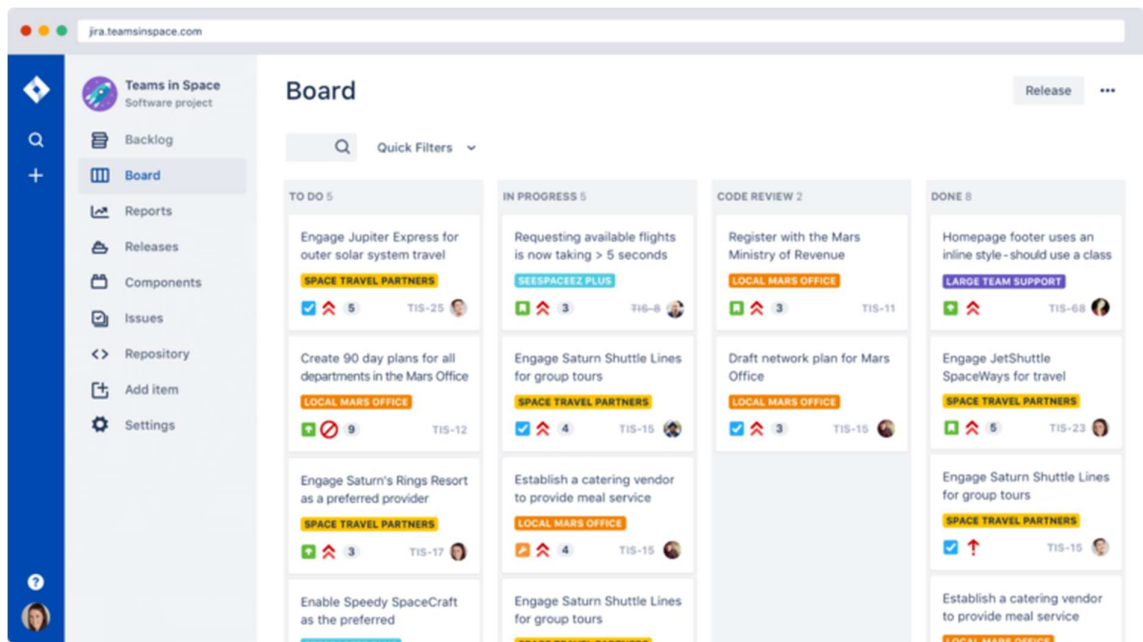
Hình 2.8 Sprint activities

## 2.3 Công nghệ, kiến trúc phần mềm được sử dụng

### 2.3.1 Jira

Jira là phần mềm quản lý công việc hàng đầu cho các nhóm phát triển phần mềm. Kiểm soát các tác vụ, các lỗi phát sinh cũng như chỉ định các công việc là các hoạt động quan trọng trong quản lý dự án. Thông thường, các nhà quản lý dự án sẽ gặp nhiều khó khăn trong việc xử lý và kiểm soát một lượng lớn thông tin. Để giải quyết vấn đề này, Jira đã ra đời để giúp tối ưu quy trình và các dự án được hoạt động trơn tru hơn.

Jira là một công cụ đám mây để theo dõi các công việc và quản lý dự án. Công cụ này được thiết kế để tăng cường phối hợp nhóm trong phát triển phần mềm Agile. Ngoài ra, Jira cũng cung cấp một bộ theo dõi bug và tiến độ toàn diện trong toàn bộ vòng đời phát triển phần mềm.



Hình 2.9 Giao diện Jira

#### ❖ Những tính năng cơ bản của Jira

- Quản lý và theo dõi tiến độ của dự án.
- Quản lý các tasks, bugs, sự cải tiến, những tính năng mới hoặc bất kỳ vấn đề xảy ra.
- Tạo ra và lưu trữ lại những bộ lọc có cấu hình cao xuyên suốt mọi vấn đề trong hệ thống.
- Chia sẻ bộ lọc với người sử dụng khác hoặc đăng ký và nhận kết quả qua hệ thống email định kỳ.
- Xây dựng quy trình làm việc tương thích với từng yêu cầu của dự án và quy trình của doanh nghiệp.
- Bảng dashboard cung cấp cho người sử dụng một không gian riêng. Nhóm xem mọi thông tin liên quan đến cá nhân.
- Cung cấp nhiều loại báo cáo thống kê với những biểu đồ khác nhau, phù hợp với nhiều loại hình dự án và đối tượng người sử dụng.
- Dễ dàng tích hợp với các hệ thống ứng dụng khác (như Email, Excel, RSS,...).
- Có thể chạy trên hầu hết các nền tảng, hệ điều hành và cơ sở dữ liệu.

#### ❖ Ưu điểm của Jira



- Jira giúp bạn lập biểu đồ ý tưởng của mình và truyền đạt chúng cho nhóm của bạn thông qua Roadmap. Tính năng này cho phép bạn chia sẻ các kế hoạch lớn và phân công nhiệm vụ cá nhân hiệu quả.
- Ứng dụng này có tất cả các tính năng bạn cần để giúp các team Agile/Scrum đi đúng hướng. Đối với các user story riêng lẻ, bạn có thể tạo các báo cáo như biểu đồ tổng hợp và đối với quản lý nhóm, bạn cũng có thể theo dõi khối lượng công việc cực kì chi tiết.
- Khả năng tích hợp mạnh mẽ . Ứng dụng cho phép tích hợp dễ dàng với các phần mềm bên thứ 3 khác. Ví dụ: với Hipchat và Slack để nhận thông báo và giao tiếp. Hơn 3000 add-ons/plugins có sẵn, giúp customize Jira với các nhu cầu và nhiệm vụ khác nhau.
- Jira cho phép người dùng tùy chỉnh trải nghiệm của họ bằng cách tạo và tùy chỉnh các phần tử khác nhau, chẳng hạn như bảng, biểu mẫu, tiến trình, báo cáo, trường và hơn thế nữa.
- Phần mềm được sử dụng cho một số nhiệm vụ khác nhau của các loại người dùng khác nhau. Cho dù bạn là nhà phát triển, người quản lý, người quản lý dự án hay kỹ sư, bạn sẽ có thể sử dụng Jira để đáp ứng nhu cầu kinh doanh của mình.

#### ❖ **Nhược điểm của Jira**

- Ứng dụng Jira trên điện thoại không phải là cách tốt nhất để làm việc với Jira khi bạn đang di chuyển. Nhiều người dùng nói rằng mặc dù mobile app của Jira đem đến một trải nghiệm tốt, nhưng họ thích làm việc bằng ứng dụng trên web hơn.
- Tích hợp với các hệ thống khác là khá phức tạp. Di chuyển một dự án từ Microsoft Team Foundation Server sang Jira là một quá trình rất tốn thời gian. Nếu bạn gặp vấn đề như thế này và cần tư vấn và hỗ trợ thì hãy liên hệ đội ngũ chuyên gia Atlassian của Biplus để được hỗ trợ tốt nhất.
- Không thể tải xuống báo cáo. Không thể tải xuống báo cáo Jira dưới dạng hình ảnh. Sử dụng các phương pháp khác làm thay đổi độ phân giải của hình ảnh, dẫn đến chất lượng hình ảnh kém. Jira đã tải lên kích thước tệp giới hạn để giữ cho công cụ này load nhanh hơn. Do đó, không được phép tải lên ảnh, video hoặc tài liệu có kích thước lớn hơn 10MB.
- Cấu hình của một số phần trong Jira đôi lúc hơi khó hiểu.

#### ❖ **Các thuật ngữ trong Jira**

Jira là phần mềm với khá nhiều thuật ngữ công nghệ. Đôi khi chúng có thể là phần khó nhất của lộ trình học tập khi bắt đầu với Jira (và phương pháp Agile).

- **Backlog** – đây chính là danh sách các công việc cần làm. Thông thường, backlog trong Jira sẽ chứa danh sách các story, issue và user story cho một sản phẩm hoặc sprint.
- **Board** – bảng biểu: Công cụ mà các nhóm sử dụng để hình dung các đơn vị công việc đang được thực hiện trong quy trình làm việc cụ thể. Nó có thể được điều chỉnh cho các phong cách phát triển phần mềm theo Agile khác nhau (ví dụ: bảng Scrum hiển thị các mục công việc chuyển từ product backlog sang sprint backlog trong khi bảng Kanban thường có quy trình làm việc ba bước: To do, In Progress và Done).
- **Burndown chart** – Biểu đồ Burndown: cho thấy khối lượng công việc thực tế và ước tính phải thực hiện trong một sprint. Đồng thời cho phép theo dõi tốc độ "burn" việc của nhóm.
- **Control chart** – Biểu đồ kiểm soát: Biểu đồ kiểm soát có thể hiển thị thời gian chu kỳ hoặc thời gian dẫn đầu cho sản phẩm, phiên bản hoặc sprint của bạn.
- **Cycle time**: Thời gian chu kỳ: Thời gian chu kỳ là thời gian dành cho việc giải quyết một issue (vấn đề) – thường là thời gian tính từ khi bắt đầu giải quyết một vấn đề đến khi hoàn thành công việc, nhưng cũng bao gồm bất kỳ thời gian nào khác dành cho việc giải quyết vấn đề. Ví dụ: nếu một vấn đề được mở lại, khắc phục và hoàn thành một lần nữa, thì thời gian cho công việc bổ sung này được thêm vào thời gian chu kỳ.
- **Daily stand-up** – họp hàng ngày – Daily là một cuộc họp nhỏ kéo dài 15 phút mỗi ngày để các nhóm đồng bộ các công việc đang được thực hiện.
- **Epic**: Epic ghi lại một khối lượng lớn công việc cần được chia nhỏ thành một số story nhỏ hơn. Có thể mất vài sprint để hoàn thành một epic. Hệ thống phân cấp cho các đơn vị công việc trong phần mềm Jira như sau: Project > Epics/Components > Stories > Tasks > Subtasks.
- **Filter** – Bộ lọc: Bộ lọc xác định những gì hiển thị trên mỗi bảng của bạn. Sử dụng trình tạo truy vấn đơn giản (JQL – Jira Query Language) của Jira, bạn có thể tùy chỉnh chính xác vấn đề (issue) nào được hiển thị trên bảng của bạn.
- **Issue**: Một issue chỉ đơn giản là một đơn vị công việc trong Jira sẽ được theo dõi thông qua một quy trình làm việc, từ khi được tạo đến khi hoàn thành. Nó có thể bao

gồm đại diện cho một đơn vị công việc, như một nhiệm vụ đơn giản hoặc một bug, đến một hạng mục công việc lớn hơn (parent work) cần được theo dõi, như một story hoặc một epic.

- **Kanban:** Kanban là một hệ thống để hình dung luồng công việc và giới hạn công việc đang thực hiện. Kanban không được định hướng theo hướng sprint, giống như phương pháp luận phát triển Scrum, vì nó theo hướng tiếp diễn liên tục.

- **Scrum:** Scrum là một phương pháp luận phát triển Agile trong đó sản phẩm được xây dựng trong một chuỗi các lần lặp có độ dài cố định được gọi là sprint. Nó cung cấp cho các nhóm một khuôn khổ để vận chuyển phần mềm theo nhịp đều đặn.

- **Scrum of Scrums:** Scrum of Scrums là một phương tiện mở rộng Scrum tới các dự án lớn, nhiều nhóm. Một kỹ thuật để mở rộng quy mô Scrum, các dự án đa đội – theo truyền thống gọi là program management.

- **Sprint:** là một khoảng thời gian ngắn (lý tưởng là từ hai đến bốn tuần) trong đó nhóm phát triển triển khai và cung cấp một phần của sản phẩm hoặc một chức năng có thể sử dụng được.

- **Sprint planning:** Một cuộc họp lập kế hoạch nhóm xác định những gì cần hoàn thành trong sprint sắp tới.

- **Sprint retrospective:** Cuộc họp nhằm xem xét lại những gì đã làm, đánh giá tốt hoặc không tốt cùng với các kế hoạch hành động để làm cho sprint tiếp theo tốt hơn.

- **Story:** Story hoặc User story là một yêu cầu hệ thống phần mềm được thể hiện bằng một vài câu ngắn, lý tưởng là sử dụng ngôn ngữ phi kỹ thuật.

- **Story point:** Story point là một ước tính về mức độ phức tạp tương đối của một story.

- **Swimlane:** Phân loại các công việc để xem xét công việc nào nên tiến hành trước.

- **Subtask:** Một nhiệm vụ phụ có thể là “tập con” của bất kỳ loại vấn đề nào, tùy thuộc vào Issue Type Scheme (Sơ đồ loại issue) của dự án.

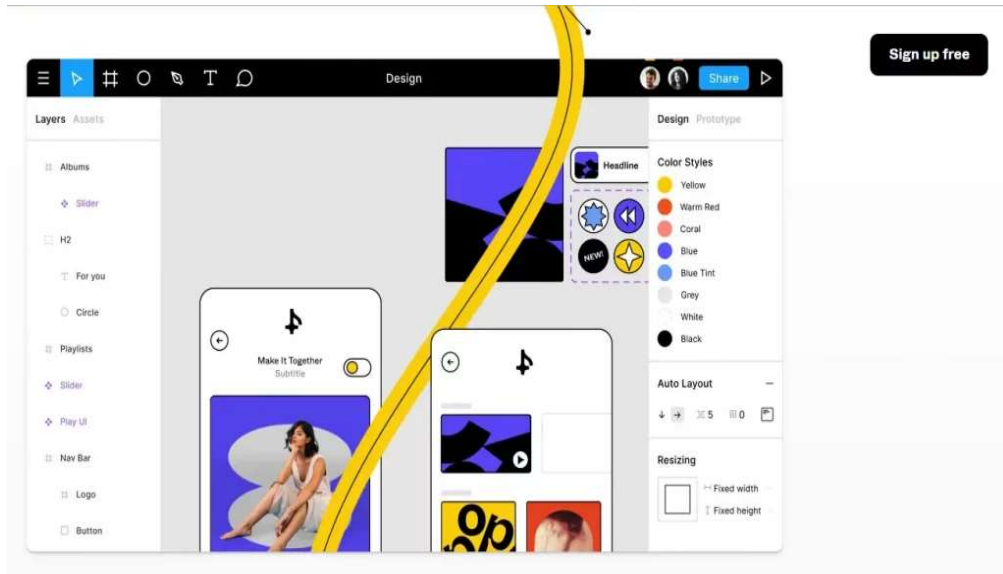
- **Task:** Một task (tác vụ) là một loại issue có sẵn trong Jira.

- **Velocity:** Velocity là thước đo mức độ công việc mà nhóm có thể xử lý trong một khoảng thời gian cụ thể, tức là nhóm có thể hoàn thành bao nhiêu product backlog của sản phẩm trong một lần sprint. Velocity có thể được tính toán dựa trên story point, giá trị kinh doanh, giờ, số lượng issue hoặc bất kỳ trường số (numeric field) nào bạn chọn.

- **Workflow** – quy trình làm việc là logic thúc đẩy chuyển động của một story / subtask dọc theo hành trình phát triển của chúng trên bảng scrum hoặc kanban.

### 2.3.2 Figma

Figma là một phần mềm thiết kế đồ họa và giao diện người dùng trực tuyến mạnh mẽ. Nó được sử dụng để tạo ra các bản thiết kế UI/UX cho ứng dụng web và di động một cách sáng tạo và không giới hạn.



Hình 2.10 Giao diện Figma

#### ❖ Ưu điểm nổi bật của Figma

Là một công cụ thiết kế đồ họa và giao diện người dùng trực tuyến mạnh mẽ và được sử dụng rộng rãi trong cộng đồng thiết kế. Nó mang lại nhiều ưu điểm nổi bật so với các công cụ thiết kế truyền thống, hãy cùng điểm qua những ưu điểm đó.

**Thiết kế đa nền tảng và đa thiết bị:** Figma là một phần mềm thiết kế đa nền tảng. Có thể sử dụng trên các nền tảng khác nhau như Windows, MacOS hay Linux. Điều này giúp cho các thiết kế viên có thể làm việc và chia sẻ dự án một cách dễ dàng và linh hoạt hơn.

**Công cụ tạo và chỉnh sửa vector tuyệt vời:** Cung cấp rất nhiều các công cụ thiết kế hữu ích để giúp cho việc thiết kế trở nên dễ dàng và chuyên nghiệp hơn. Có thể tạo ra các biểu tượng, các hình ảnh vector, các hiệu ứng chữ và rất nhiều thứ khác. Ngoài ra, Figma cũng hỗ trợ tính năng thử nghiệm giao diện người dùng trực tiếp trên ứng dụng hay website mà bạn đang thiết kế.

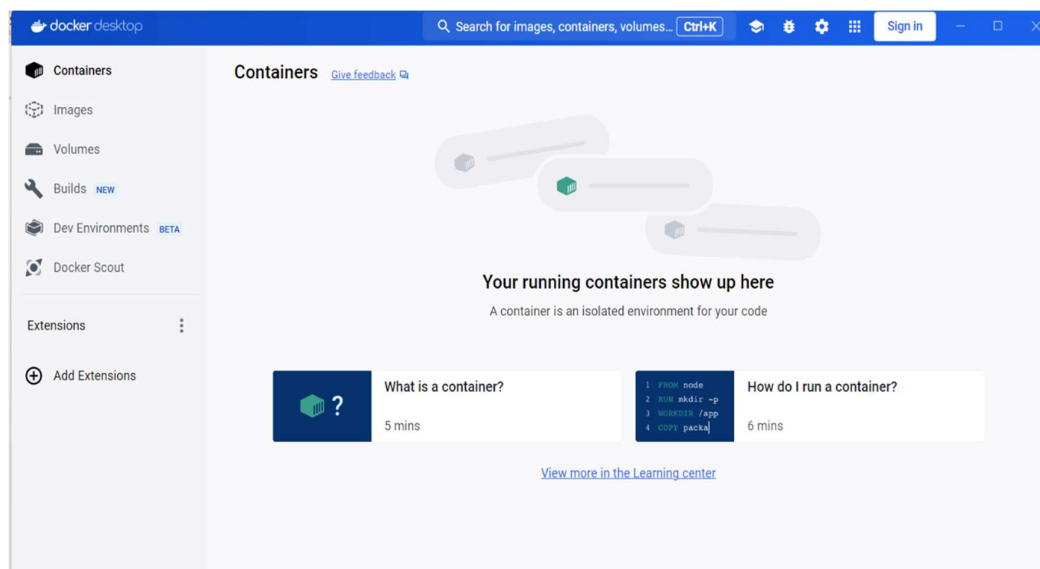
**Cộng tác và chia sẻ thiết kế dễ dàng:** Figma cho phép nhiều người cùng làm việc trên cùng một dự án. Chúng ta có thể chia sẻ dự án với đội ngũ thiết kế và các thành viên khác trong công ty để họ có thể theo dõi và đóng góp ý kiến. Tính năng này giúp cho quá trình làm việc của đội ngũ thiết kế trở nên hiệu quả và nhanh chóng hơn.

**Thanh công cụ tiện lợi:** Thanh công cụ tiện lợi giúp người dùng có thể truy cập nhanh chóng đến các công cụ và tính năng quan trọng. Thanh công cụ này cung cấp các công cụ thiết kế đồ họa, các công cụ làm việc với định dạng văn bản, hiệu ứng, phác thảo, và nhiều hơn nữa.

**Dễ dàng sử dụng cho người mới bắt đầu:** Figma là một công cụ thiết kế dựa trên đám mây. Các dữ liệu của chúng ta được lưu trữ và quản lý trên đám mây, giúp cho việc lưu trữ và chia sẻ dữ liệu trở nên tiện lợi và an toàn hơn. Đồng thời, tính năng này cũng giúp cho chúng ta có thể làm việc từ bất cứ đâu mà không cần phải cài đặt phần mềm trên máy tính. Giao diện dễ sử dụng và thân thiện với người dùng, đặc biệt là với những người mới bắt đầu với thiết kế đồ họa. Các tính năng và công cụ được sắp xếp một cách rõ ràng và dễ dàng để truy cập, giúp cho người dùng có thể tập trung vào việc tạo ra các thiết kế chất lượng cao.

### 2.3.3 Docker

Docker là một công cụ đóng gói phần mềm và triển khai ứng dụng trong các container. Các ứng dụng chạy trong các container được gọi là các container Docker. Docker giúp các lập trình viên chỉ cần chạy các container lên mà không cần phải lo về



Hình 2.11 Giao diện Docker

việc cài các thư viện hoặc môi trường, điều này giúp các lập trình viên phát triển ứng dụng một cách nhanh chóng, đảm bảo tính nhất quán giữa môi trường phát triển và môi trường triển khai.

#### ❖ Các thành phần cơ bản

**Docker daemon:** Là một tiến trình nền chạy trên một máy tính Docker host ( máy cài Docker ), và quản lý các hoạt động Docker như tạo và quản lý các container, images, networks và volumes.

**Docker client:** Là một ứng dụng dòng lệnh hoặc giao diện người dùng đồ họa (GUI) để tương tác với Docker daemon và thực hiện các hoạt động Docker. Docker client sử dụng Docker API để giao tiếp với Docker daemon.

**Docker images:** Là một gói đóng gói của một ứng dụng và các tài nguyên cần thiết để chạy ứng dụng đó trong một container. Một image có thể được tạo từ một Dockerfile hoặc tải từ một kho chứa image trên internet như Docker Hub.

**Docker container:** Là một môi trường đóng gói độc lập, chứa tất cả các thành phần cần thiết để chạy một ứng dụng trong một môi trường cô lập, container được tạo ra từ image sau khi đã đóng gói

**Docker network:** cho phép các container tương tác với nhau và với các dịch vụ khác. Một Docker network được tạo ra để tạo một mạng ảo cho các container chạy trên cùng một máy Docker host

**Docker volumes:** Cho phép các container lưu trữ và truy cập dữ liệu được sử dụng bởi các ứng dụng. Volumes giúp dữ liệu được bảo vệ và giữ cho đồng bộ giữa các container.

#### ❖ Các lệnh cơ bản

Đây là 1 số câu lệnh phổ biến và hay sử dụng trong quá trình sử dụng Docker:

- ***docker run:*** Lệnh để chạy một container từ một image.
- ***docker pull:*** Lệnh để tải một image từ Docker Registry về máy tính.
- ***docker push:*** Lệnh để đẩy một image lên Docker Registry.
- ***docker build:*** Lệnh để tạo một image từ Dockerfile.
- ***docker images:*** Lệnh để hiển thị danh sách các image có sẵn trong Docker.

- ***docker ps***: Lệnh để hiển thị danh sách các container đang chạy.
- ***docker stop***: Lệnh để dừng một container đang chạy.
- ***docker rm***: Lệnh để xóa một container đã dừng hoặc bị lỗi.
- ***docker rmi***: Lệnh để xóa một image.
- ***docker exec***: Lệnh để thực thi một lệnh trong một container đang chạy.

## CHƯƠNG 3: XÁC ĐỊNH NHU CẦU

### 3.1 Đặc tả mục tiêu dự án

Xây dựng một hệ thống phần mềm quản lý thiết bị hoàn chỉnh và hiệu quả cho trường trung học phổ thông.

Mục tiêu là cung cấp một nền tảng để tổ chức và theo dõi tất cả các thiết bị trong trường, từ việc quản lý thông tin, tình trạng, đến việc quản lý sử dụng và bảo trì.

Sử dụng Jira để quản lý dự án, theo dõi tiến độ, và giao việc cho các thành viên trong nhóm phát triển. Jira cũng có thể được sử dụng để quản lý các yêu cầu từ người dùng và theo dõi các vấn đề kỹ thuật trong quá trình phát triển.

Sử dụng Figma để thiết kế giao diện người dùng của hệ thống quản lý thiết bị. Figma cho phép đồng bộ hóa và chia sẻ các thiết kế với nhóm phát triển và người dùng, từ đó tạo ra một giao diện người dùng hấp dẫn và dễ sử dụng.

Sử dụng Docker để đóng gói ứng dụng thành các container độc lập và di động. Điều này giúp dễ dàng triển khai ứng dụng trên nhiều môi trường khác nhau mà không cần lo lắng về sự phụ thuộc vào môi trường cụ thể.

### 3.2 Xác định Personas

Personas là 'người dùng tưởng tượng' nơi bạn tạo chân dung nhân vật của một loại người dùng mà bạn nghĩ có thể sử dụng sản phẩm của mình.

Các nhóm người dùng tiềm năng như:

- Ban giám hiệu trường: Người quản lý toàn bộ hoạt động của trường, có trách nhiệm cao trong việc đảm bảo tài nguyên được sử dụng hiệu quả.
- Nhân viên quản lý thiết bị: Người chịu trách nhiệm trực tiếp về việc quản lý, bảo trì và sử dụng các thiết bị trong trường.

### 3.3 Xác định các User Stories

- Quản lý tương tác người dùng: Đăng nhập và đăng xuất: Người dùng có thể đăng nhập vào hệ thống để truy cập các chức năng và thông tin.
- Quản lý thiết bị:
  - + Cập nhật thông tin thiết bị: Người dùng có thể thêm, sửa đổi hoặc xóa thông tin về các thiết bị.



- + Tìm kiếm thiết bị: Người dùng có thể tìm kiếm thông tin về các thiết bị dựa trên các tiêu chí như tên, loại, tình trạng.
- Quản lý tình trạng thiết bị:
  - + Nhập thêm thiết bị: Người dùng có thể thêm mới các thiết bị vào hệ thống và ghi nhận thông tin về chúng.
  - + Lập phiếu nhập hàng: Hệ thống tự động tạo phiếu nhập hàng khi có thiết bị mới được thêm vào.
- Quản lý sử dụng thiết bị:
  - + Lập danh sách mượn thiết bị: Người dùng có thể tạo danh sách các thiết bị được mượn và ghi nhận thông tin về người mượn và thời gian mượn.
  - + Lập danh sách trả thiết bị: Người dùng có thể cập nhật thông tin khi thiết bị được trả lại.

### **3.4 Xác định các Product Backlog**

- Quản lý tương tác người dùng:
  - + Tạo chức năng đăng nhập và đăng xuất.
  - + Thiết kế giao diện người dùng cho trang đăng nhập.
  - + Xây dựng chức năng xác thực người dùng.
- Quản lý thiết bị:
  - + Tạo giao diện để hiển thị danh sách các thiết bị.
  - + Tạo chức năng thêm mới thiết bị.
  - + Tạo chức năng sửa thông tin thiết bị.
  - + Tạo chức năng xóa thiết bị.
  - + Tạo chức năng tìm kiếm thiết bị.
- Quản lý nhập thiết bị:
  - + Tạo giao diện để hiển thị danh sách nhập các thiết bị.
  - + Tạo chức năng thêm mới thiết bị.
  - + Tạo chức năng sửa thông tin thiết bị.
  - + Tạo chức năng xóa thiết bị.
  - + Tạo chức năng tìm kiếm thiết bị.
- Quản lý sử dụng thiết bị:
  - + Tạo giao diện để hiển thị danh sách mượn và trả thiết bị.

- + Tạo chức năng lập danh sách mượn thiết bị.
- + Tạo chức năng lập danh sách trả thiết bị.
- + Tạo chức năng thêm mới thiết bị.
- + Tạo chức năng sửa thông tin thiết bị.
- + Tạo chức năng xóa thiết bị.
- + Tạo chức năng tìm kiếm thiết bị.
- Thông tin nhân viên quản lý thiết bị:
  - + Tạo giao diện để hiển thị danh sách nhân viên quản lý thiết bị.
  - + Tạo chức năng thêm mới thông tin nhân viên.
  - + Tạo chức năng sửa thông tin nhân viên.
  - + Tạo chức năng xóa thông tin nhân viên.
  - + Tạo chức năng tìm kiếm thiết bị.
- Báo cáo thống kê:
  - + Thiết kế giao diện cho các báo cáo thống kê.
  - + Tạo chức năng tạo báo cáo số lượng thiết bị nhập.

### **3.5 Xác định các nhu cầu phi tính năng**

- Bảo mật: Đảm bảo rằng dữ liệu trong hệ thống được bảo vệ an toàn, chỉ có người dùng có quyền truy cập được phép.
- Hiệu suất: Đảm bảo rằng hệ thống có khả năng xử lý dữ liệu lớn và có thể hoạt động mượt mà dưới tải trọng cao.
- Tính linh hoạt và mở rộng: Thiết kế hệ thống sao cho có thể mở rộng và tích hợp với các hệ thống khác một cách dễ dàng khi cần thiết.
- Tính ổn định và độ tin cậy: Đảm bảo rằng hệ thống luôn hoạt động ổn định và có khả năng khôi phục dữ liệu sau khi có sự cố.
- Trải nghiệm người dùng: Đảm bảo trải nghiệm người dùng sản phẩm sẽ được cung cấp giao diện dễ sử dụng, thân thiện và đáp ứng nhu cầu của người dùng.

## CHƯƠNG 4: LẬP KẾ HOẠCH SCRUM

### 4.1 Sắp xếp các Product Backlog

Dựa vào các Product Backlog đã xác định ở trên sắp xếp chúng theo trình tự thực hiện ưu tiên để cho quá trình phát triển dự án được hiệu quả và tối ưu.

- Thiết kế giao diện đăng nhập
- Thiết kế giao diện trang chủ
- Quản lý danh sách thiết bị
- Chức năng mượn thiết bị
- Chức năng trả thiết bị
- Chức năng nhập thiết bị
- Chức năng quản lý nhân viên
- Kiểm tra sửa lỗi
- Bảo trì và hoàn thiện
- Giao cho khách hàng test

### 4.2 Thiết kế các Sprint trên Jira

Phân công thực hiện các Sprint với 03 vai trò: Scrum Master, Product Owner, QA.

Scrum Master lập bảng phân công thực hiện các product backlog cho từng thành viên cụ thể như sau:

Issue	Person	Est. Story Points	Sprint	Start	End
Viết code tạo cơ sở dữ liệu	Thương	3	Sprint 1	29/04/2024	30/04/2024
Viết code nhập dữ liệu cho các bảng dữ liệu	Hân	4	Sprint 1	01/05/2024	02/05/2024
Thiết kế giao diện đăng nhập	Băng	2	Sprint 1	03//04/2024	04/05/2024

Issue	Person	Est. Story Points	Sprint	Start	End
Thiết kế giao diện trang dashboard	Hân	3	Sprint 2	05/05/2024	06/05/2024
Quản lý danh sách thiết bị	Thương	3	Sprint 2	07/04/2024	09/05/2024
Chức năng mượn thiết bị	Băng	4	Sprint 2	10/04/2024	12/04/2024
Chức năng trả thiết bị	Thương	5	Sprint 3	13/05/2024	15/05/2024
Chức năng nhập thiết bị	Băng	5	Sprint 3	16/05/2024	18/05/2024
Lập báo cáo	Hân	5	Sprint 3	19/05/2024	21/05/2024
Quản lý nhân viên	Thương	4	Sprint 4	22/05/2024	24/05/2024
Kiểm thử tích hợp	Hân	6	Sprint 4	25/05/2024	26/05/2024
Sửa lỗi và bảo trì	Băng	3	Sprint 4	27/05/2024	29/05/2024

Bảng 1. 1 Bảng kế hoạch Scrum trên Jira

<input type="checkbox"/>	Type	# Key	Summary	Status	Sprint	Assignee	Due
<input type="checkbox"/>		SCRUM-1	Viết code tạo cơ sở dữ liệu	DONE	SCRUM Sprint 1	SM Scrum Master	
<input type="checkbox"/>		SCRUM-2	Viết code nhập dữ liệu cho các bảng	DONE	SCRUM Sprint 1	PO Product Owner	
<input type="checkbox"/>	+	SCRUM-3	Thiết kế giao diện đăng nhập	DONE	SCRUM Sprint 1	QA	

Hình 4.1 Sprint 1

	SCRUM-9	Thiết kế giao diện trang dashboard	DONE	SCRUM Sprint 2	PO Product Owner
	SCRUM-10	Thiết kế trang quản lý danh sách thiết bị	DONE	SCRUM Sprint 2	SM Scrum Master
+	SCRUM-11	Thiết kế trang chức năng mượn thiết bị	DONE	SCRUM Sprint 2	QA

Hình 4.2 Sprint 2

	SCRUM-6	Thiết kế trang chức năng trả thiết bị	DONE	SCRUM Sprint 3	SM Scrum Master
	SCRUM-7	Thiết kế trang chức năng nhập thiết bị	DONE	SCRUM Sprint 3	QA
+	SCRUM-12	Thiết kế lập báo cáo	DONE	SCRUM Sprint 3	PO Product Owner

Hình 4.3 Sprint 3

	SCRUM-13	Thiết kế trang quản lý nhân viên	DONE	SCRUM Sprint 4	SM Scrum Master
	SCRUM-14	Kiểm thử tích hợp	DONE	SCRUM Sprint 4	PO Product Owner
+	SCRUM-15	Sửa lỗi và bảo trì	DONE	SCRUM Sprint 4	QA

Hình 4.4 Sprint 4

## CHƯƠNG 5: HIỆN THỰC HÓA KẾ HOẠCH

### 5.1 Thiết kế giao diện trên Figma

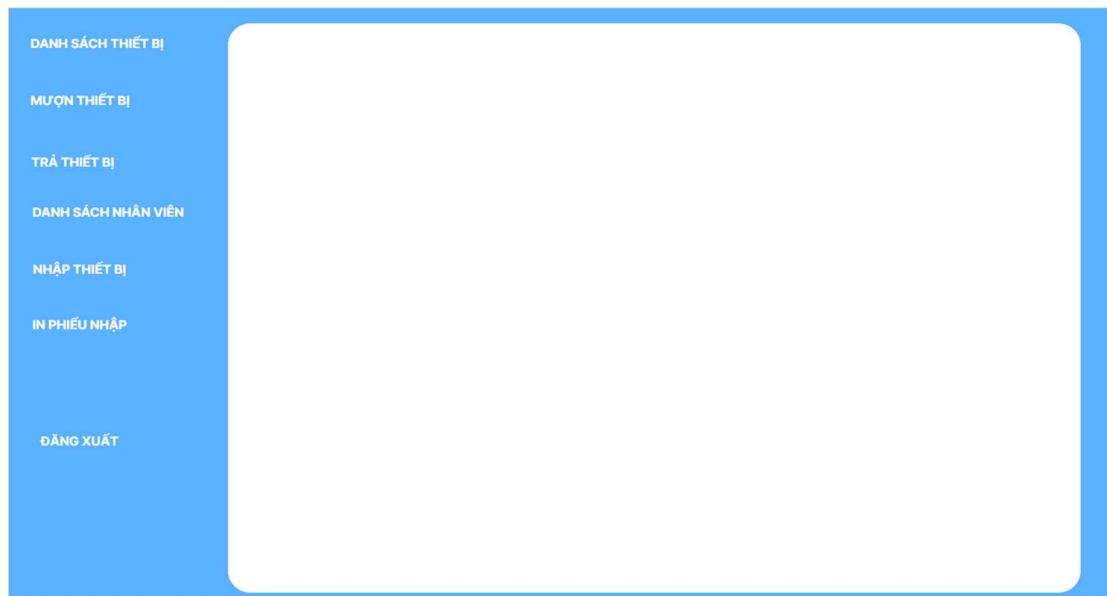
#### 5.1.1 Giao diện đăng nhập



The image shows a login form titled "ĐĂNG NHẬP" (Login) centered on a white background with a blue and pink gradient border. The form includes a "Tài khoản" (Account) field with the text "nhanvien123", a "Mật khẩu" (Password) field with the text "123", and a checkbox labeled "Hiện thị mật khẩu" (Show password) which is checked. Below the fields are two buttons: "ĐĂNG NHẬP" (Login) and "THOÁT" (Exit).

Hình 5.1 Giao diện đăng nhập

#### 5.1.2 Giao diện dashbroad



The image shows a dashboard layout with a blue sidebar on the left and a large white main content area on the right. The sidebar contains a list of menu items: "DANH SÁCH THIẾT BỊ" (Device List), "MƯỢN THIẾT BỊ" (Borrow Device), "TRẢ THIẾT BỊ" (Return Device), "DANH SÁCH NHÂN VIÊN" (Staff List), "NHẬP THIẾT BỊ" (Enter Device), "IN PHIẾU NHẬP" (Print Receipt), and "ĐĂNG XUẤT" (Logout).

Hình 5.2 Giao diện dashbroad

### 5.1.3 Giao diện danh sách thiết bị

DANH SÁCH THIẾT BỊ  
MƯỢN THIẾT BỊ  
TRẢ THIẾT BỊ  
DANH SÁCH NHÂN VIÊN  
NHẬP THIẾT BỊ  
IN PHIẾU NHẬP  
ĐĂNG XUẤT

DANH SÁCH THIẾT BỊ

Mã thiết bị

Nhà cung cấp

Tên thiết bị

Bảo hành

Ngày nhập mm/dd/yyyy

Nguyên giá

Số lượng

Tình trạng

Ghi chú

NHẬP LẠI

THÊM

XÓA

SỬA

Mã thiết bị

Tên thiết bị

Ngày nhập

Số lượng mua

Nhà cung cấp

Bảo hành

Giá mua

Tình trạng

Ghi chú

01	Máy chiếu	01/01/2023	7	Nhà cung cấp A	2 năm	7.000.000 đ	Tốt	
02	Remote	02/01/2023	7	Nhà cung cấp B	2 năm	3.500.000 đ	Bình thường	
03	Cổng kết nối	03/01/2023	5	Nhà cung cấp A	2 năm	1.500.000 đ	Bình thường	
04	Ổ điện	04/01/2023	10	Nhà cung cấp C	2 năm	3.000.000 đ	Bình thường	
05	Nam châm	05/01/2023	30	Nhà cung cấp D	2 năm	240.000 đ	Bình thường	

Hình 5.3 Giao diện danh sách thiết bị

### 5.1.4 Giao diện danh sách mượn thiết bị

DANH SÁCH THIẾT BỊ  
MƯỢN THIẾT BỊ  
TRẢ THIẾT BỊ  
DANH SÁCH NHÂN VIÊN  
NHẬP THIẾT BỊ  
IN PHIẾU NHẬP  
ĐĂNG XUẤT

DANH SÁCH MƯỢN THIẾT BỊ

Mã phiếu mượn

Giới tính Nam

Số điện thoại

Người mượn

Số lượng

Tên thiết bị

Ngày mượn mm/dd/yyyy

Tên phòng

Mã NV NV001

Ca mượn

CCCD

Ghi chú

NHẬP LẠI

THÊM

XÓA

SỬA

Mã phiếu mượn

Người mượn

Ngày mượn

Ca mượn

Giới tính

Số lượng

Tên phòng

CCCD

Số ĐT

Tên thiết bị

Mã NV

Ghi chú

001	Gv.Nguyễn Thị Xuân	2023-11-01	Sáng	Nữ	2	Phòng lý thuyết 1	085412345678	0343 823 756	1.Máy chiếu, 1.Remote	NV001	
002	Gv.Bùi Quốc Tuấn	2023-11-01		Nam	2	Phòng lý thuyết 2	089765436543	0982 741 232	1.Máy chiếu, 1.Remote	NV001	
003	Gv.Nguyễn Gia Nghi	2023-11-01	Chiều	Nam	3	Phòng lý thuyết 3	085673290136	0762 258 313	1.Máy chiếu, 1.Remote, 1.Loa	NV001	
004	Gv.Trần Văn Minh	2023-11-02	Sáng	Nam	16	Phòng lý thuyết 4	085934713168	0128 356 440	1.Ổ điện, 15.Nam châm	NV002	

Hình 5.4 Giao diện danh sách mượn thiết bị

5.1.5 Giao diện danh sách trả thiết bị

DANH SÁCH THIẾT BỊ

MƯỢN THIẾT BỊ

TRẢ THIẾT BỊ

DANH SÁCH NHÂN VIÊN

NHẬP THIẾT BỊ

IN PHIẾU NHẬP

ĐĂNG XUẤT

DANH SÁCH TRẢ THIẾT BỊ

Mã phiếu trả

Giới tính

Số điện thoại

Người trả

Số lượng

Tên thiết bị

Ngày trả

Tên phòng

Mã NV

Ca trả

CCCD

Ghi chú

NHẬP LẠI

THÊM

XÓA

SỬA

Mã phiếu trả	Người trả	Ngày trả	Ca trả	Giới tính	Số lượng	Tên phòng	CCCD	Số ĐT	Tên thiết bị	Mã NV	Ghi chú
PT001	Gv.Nguyễn Thị Xuân	2023-11-01	Sáng	Nữ	2	Phòng lý thuyết 1	085412345678	0343 823 756	1.Máy chiếu, 1.Remote	NV001	
PT002	Gv.Bùi Quốc Tuấn	2023-11-01	Sáng	Nam	2	Phòng lý thuyết 2	089765436543	0982 741 232	1.Máy chiếu, 1.Remote	NV001	
PT003	Gv.Nguyễn Gia Nghi	2023-11-01	Chiều	Nam	3	Phòng lý thuyết 3	085673290136	0762 258 313	1.Máy chiếu, 1.Remote, 1.Loa	NV001	
PT004	Gv.Trần Văn Minh	2023-11-02	Sáng	Nam	16	Phòng lý thuyết 4	085934713168	0128 356 440	1.Ổ điện, 15.Nam châm	NV002	

Hình 5.5 Giao diện danh sách trả thiết bị

5.1.6 Giao diện danh sách nhân viên

DANH SÁCH THIẾT BỊ

MƯỢN THIẾT BỊ

TRẢ THIẾT BỊ

DANH SÁCH NHÂN VIÊN

NHẬP THIẾT BỊ

IN PHIẾU NHẬP

ĐĂNG XUẤT

DANH SÁCH NHÂN VIÊN

Mã nhân viên

Ngày sinh

Mật khẩu

Tên nhân viên

CCCD

Quyền

Địa chỉ

Giới tính

Số điện thoại

Tài khoản

NHẬP LẠI

THÊM

XÓA

SỬA

Mã NV	Tên NV	Ngày sinh	Giới tính	Địa chỉ	Số ĐT	CCCD	Tài khoản	Mật khẩu	Quyền
NV001	Nguyễn Thị An	02/03/1995	Nữ	Khóm 1, Phường 1, TP.Trà Vinh, Tỉnh Trà Vinh	0123456789	089359753217	an	123	NhanVien
NV002	Trần Văn Bình	04/05/1995	Nam	Khóm 2, Phường 1, TP.Trà Vinh, Tỉnh Trà Vinh	0987654321	089731579241	bình	123	NhanVien
NV003	Lê Minh Cường	05/01/1996	Nam	Khóm 3, Phường 1, TP.Trà Vinh, Tỉnh Trà Vinh	0369871245	080638593165	cuong	123	NhanVien
NV004	Phạm Thị Dung	09/12/1997	Nữ	Khóm 4, Phường 1, TP.Trà Vinh, Tỉnh Trà Vinh	0743217890	083815730825	Dung	123	NhanVien

Hình 5.6 Giao diện danh sách nhân viên



### 5.1.7 Giao diện danh sách nhập thiết bị

DANH SÁCH THIẾT BỊ

MƯỢN THIẾT BỊ

TRẢ THIẾT BỊ

DANH SÁCH NHÂN VIÊN

NHẬP THIẾT BỊ

IN PHIẾU NHẬP

ĐĂNG XUẤT

PHIẾU NHẬP THIẾT BỊ

Mã phiếu nhập

Nhà cung cấp

Tên thiết bị

Giá nhập

Ngày nhập

Mã NV

Số lượng nhập

Ghi chú

NHẬP LẠI

THÊM

XÓA

SỬA

Mã phiếu nhập	Tên thiết bị	Ngày nhập	Số lượng mua	Nhà cung cấp	Giá nhập	Mã NV	Ghi chú
PN001	Máy chiếu	10/11/2023	3	Nhà cung cấp A	3.000.000 đ	NV001	
PN002	Tivi	15/11/2023	4	Nhà cung cấp B	16.000.000 đ	NV001	
PN003	Loa	14/11/2023	2	Nhà cung cấp C	1.000.000 đ	NV002	
PN004	Ổ điện	17/10/2023	3	Nhà cung cấp A	300.000 đ	NV002	

Hình 5.7 Giao diện nhập thiết bị

### 5.1.8 Giao diện in phiếu nhập

DANH SÁCH THIẾT BỊ

MƯỢN THIẾT BỊ

TRẢ THIẾT BỊ

DANH SÁCH NHÂN VIÊN

NHẬP THIẾT BỊ

IN PHIẾU NHẬP

ĐĂNG XUẤT

Mã phiếu nhập

PN001

LỌC

PHIẾU NHẬP THIẾT BỊ

Mã phiếu nhập	Tên thiết bị	Ngày nhập	Số lượng mua	Nhà cung cấp	Giá nhập	Mã NV	Ghi chú
PN001	Máy chiếu	10/11/2023	3	Nhà cung cấp A	3.000.000 đ	NV001	
PN002	Tivi	15/11/2023	4	Nhà cung cấp B	16.000.000 đ	NV001	
PN003	Loa	14/11/2023	2	Nhà cung cấp C	1.000.000 đ	NV002	
PN004	Ổ điện	17/10/2023	3	Nhà cung cấp A	300.000 đ	NV002	

Hình 5.8 Giao diện in phiếu nhập













## 5.2 Mã nguồn

- Các file mã nguồn được upload lên Github như sau:

 03Thuong	Cập nhập file chạy docker	aea8ab5 · now	🕒 18 Commits
 DS_NHANVIEN	Thiết kế trang danh sách nhân viên		2 weeks ago
 INPHIEUNHAP	Thiết kế trang in phiếu nhập		3 weeks ago
 Login	Thiết kế giao diện đăng nhập		last month
 MUON_TB	Thiết kế trang chức năng mượn thiết bị		last month
 QLTB/QLTB	Source code (c#) quản lý thiết bị		last week
 QUANLY_THIETBI_THPT	Cập nhập file chạy docker		now
 TRA_TB	Thiết kế trang chức năng trả thiết bị		last month
 danhsachtb	Thiết kế trang danh sách thiết bị		last month
 dashbroad	Thiết kế dashroad		last month
 nhaptb	Thiết kế trang chức năng nhập thiết bị		3 weeks ago
 README.md	Initial commit		2 months ago
 SourceCode_ql_thietbi_thpt.sql	Viết code nhập dữ liệu cho các bảng		last month

Hình 5.9 Các file mã nguồn được upload lên Github

- Lịch sử commit mã nguồn lên Github:

Thiết kế trang danh sách nhân viên	fe48478	 
03Thuong committed 2 weeks ago		
Commits on May 21, 2024		
Thiết kế trang in phiếu nhập	b44aa62	 
NgocHan221 committed 3 weeks ago		
Commits on May 18, 2024		
Thiết kế trang chức năng nhập thiết bị	ea674e6	 
bang2003 committed 3 weeks ago		
Commits on May 15, 2024		
Thiết kế trang chức năng trả thiết bị	48975c2	 
03Thuong committed last month		
Commits on May 12, 2024		
Thiết kế trang chức năng mượn thiết bị	b7c834e	 
bang2003 committed last month		
Commits on May 9, 2024		
Thiết kế trang danh sách thiết bị	f56bb49	 
03Thuong committed last month		

Hình 5.10 Lịch sử commit Github

- Chạy các giao diện trên Docker:
- + Tạo file docker-compose.yml với nội dung sau:

```
version: '3'

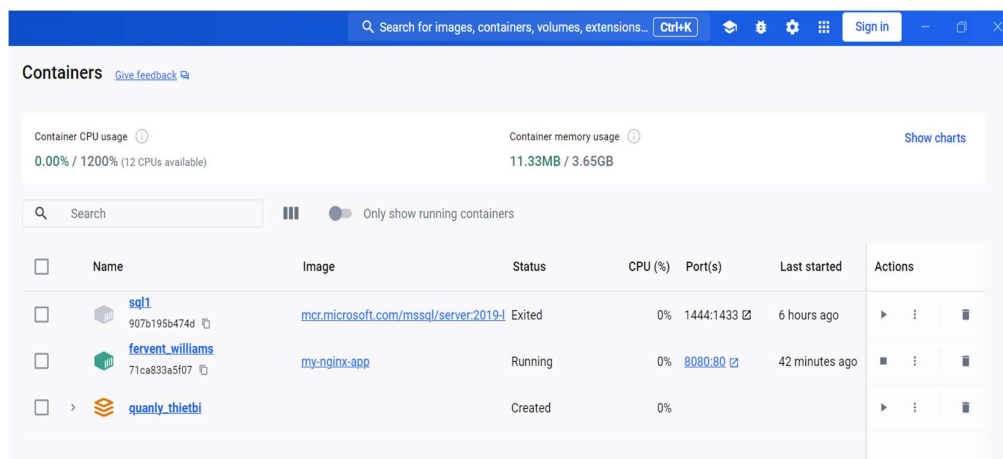
services:
  web:
    build: .
    ports:
      - "8080:80"
```

- + Tạo Dockerfile với nội dung sau:

```
# Sử dụng hình ảnh chính thức của Nginx từ Docker Hub
FROM nginx:alpine

# Sao chép các tệp HTML của bạn vào container Nginx
COPY ./html /usr/share/nginx/html

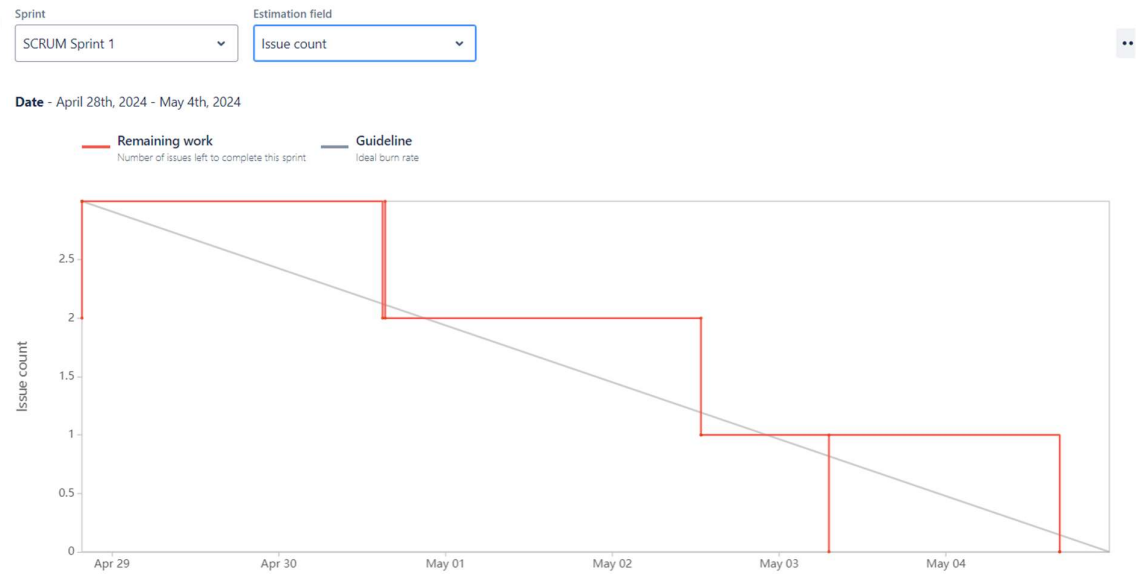
# Mở cổng 80 ra bên ngoài
EXPOSE 80
```



Hình 5.11 Chạy các giao diện trên Docker

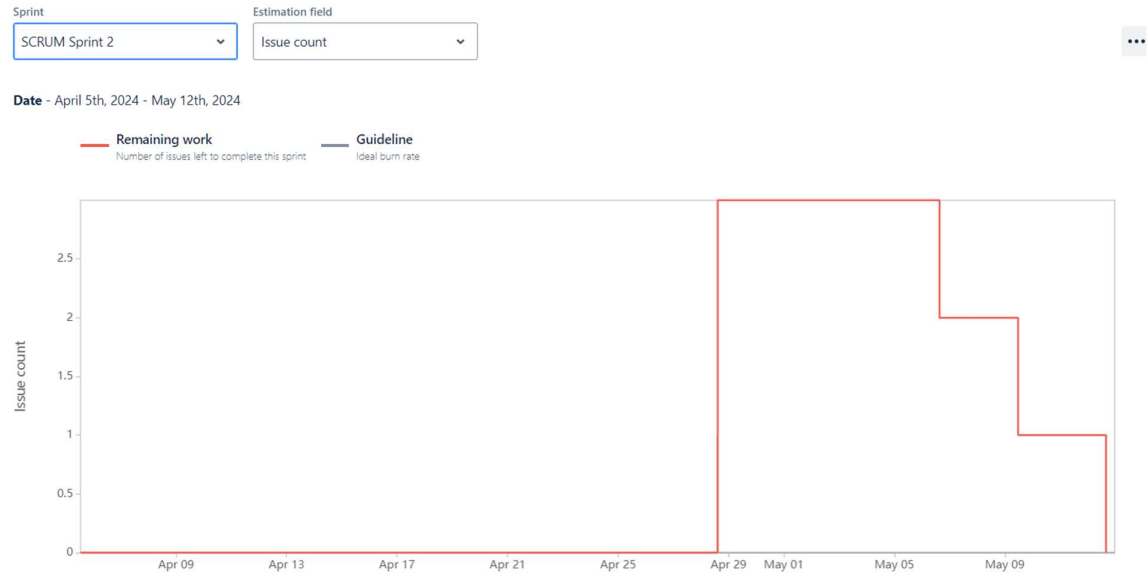
## 5.3 Burndown Chart

### 5.3.1 Burndown Chart Sprint 1



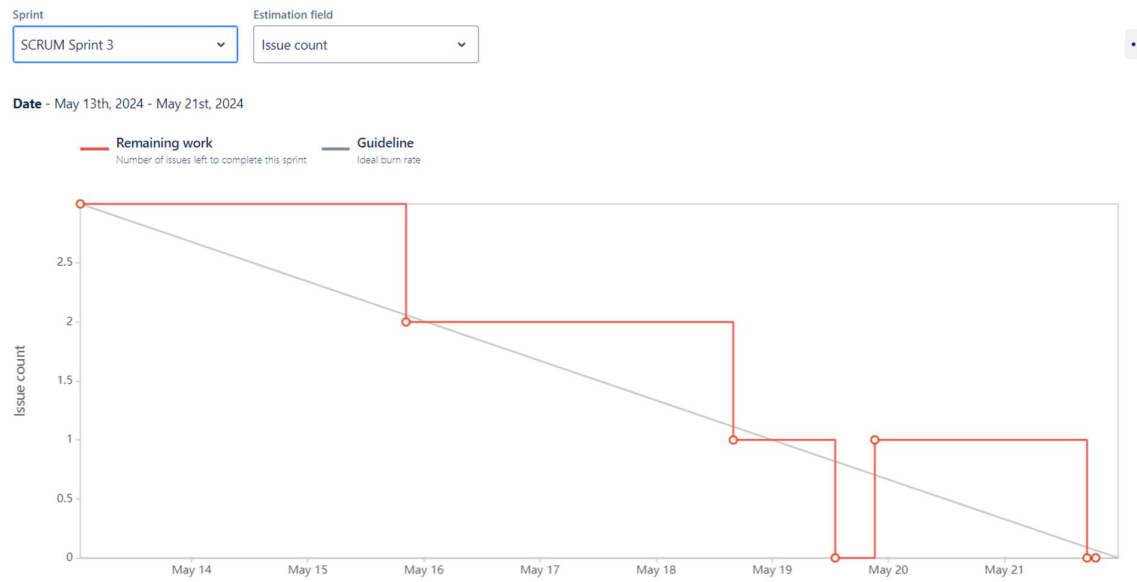
Hình 5.12 Burndown Chart Sprint 1

### 5.3.2 Burndown Chart Sprint 2



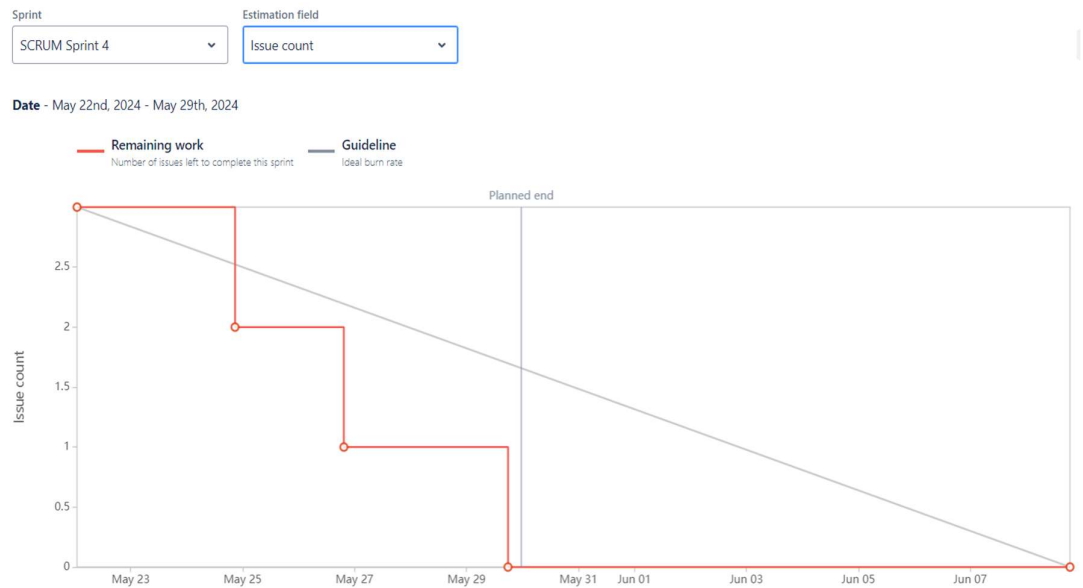
Hình 5.13 Burndown Chart Sprint 2

### 5.3.3 Burndown Chart Sprint 3



Hình 5.14 Burndown Chart Sprint 3

### 5.3.4 Burndown Chart Sprint 4



Hình 5.15 Burndown Chart Sprint 4

## CHƯƠNG 6: KẾT LUẬN

### 6.1 Kết quả đạt được

Sau một quá trình làm việc tích cực và phối hợp hiệu quả theo quy trình phát triển phần mềm Agile, nhóm chúng tôi đã hoàn thành “Tìm hiểu và xây dựng hệ thống quản lý thiết bị dạy học tại trường trung học phổ thông”. Dưới sự hướng dẫn và điều phối của Scrum Master, chúng tôi đã thiết lập và duy trì một quy trình làm việc rõ ràng và hiệu quả trên Jira và GitHub.

Scrum Master đã thiết lập dự án trên Jira, tạo bảng phân công chi tiết và điều phối dự án một cách hợp lý. Sự quan sát và quản lý tiến độ thông qua burndown chart đã giúp nhóm luôn nắm rõ tình hình và điều chỉnh kế hoạch kịp thời. GitHub repository cũng được sử dụng một cách hiệu quả để lưu trữ và quản lý mã nguồn.

Các thành viên trong nhóm đã tuân thủ nghiêm ngặt các cam kết về thời gian, cập nhật trạng thái dự án trên Jira và đảm bảo mã nguồn được commit đúng hạn. Việc quan sát và cập nhật tiến độ dựa trên burndown chart đã giúp các thành viên nhận thức rõ ràng về tiến độ và trách nhiệm của mình.

Kết quả đạt được là một sản phẩm phần mềm với giao diện đơn giản sử dụng.

### 6.2 Hạn chế

Dự án đã hoàn thành đúng tiến độ đặt ra tuy nhiên vẫn còn có một số sai sót:

- Việc quản lý thời gian, quản lý tiến độ còn chậm trễ
- Các chức năng chưa được tối ưu hóa
- Giao diện chưa thật sự hoàn hảo
- Chưa kết hợp được với cơ sở dữ liệu SQL

### 6.3 Hướng phát triển

Liên tục đánh giá và cải tiến quy trình Agile của nhóm. Tổ chức các buổi retrospectives để nhận diện và khắc phục các vấn đề gặp phải, từ đó tối ưu hóa hiệu suất làm việc của cả nhóm.

Tiếp tục hoàn thiện dự án, kết nối cơ sở dữ liệu, xử lý tối ưu hóa các chức năng. Chỉnh sửa giao diện đẹp hơn, đầy đủ tính năng hơn.

## DANH MỤC TÀI LIỆU THAM KHẢO

1. Slide bài giảng môn Công nghệ phần mềm của Thầy Nguyễn Bảo Ân, Khoa Kỹ thuật và Công nghệ, Trường Đại học Trà Vinh.
2. Sách “Engineering Software Products: An Introduction to Modern Software Engineering” của Ian Sommerville.
3. Thuận Jira. (2022). "Jira là gì? Tổng quan về Jira." Truy cập từ Viblo. URL: <https://viblo.asia/p/jira-la-gi-tong-quan-ve-jira-gGJ590XplX2>.
4. "Figma là gì?" Truy cập từ Tenten.vn. URL: <https://tenten.vn/tin-tuc/figma-la-gi/#:~:text=Figma%20l%C3%A0%20m%E1%BB%99t%20ph%E1%BA%A7n%20m%E1%BB%81m,d%E1%BB%B1%20%C3%A1n%20thi%E1%BA%BFt%20k%E1%BA%BF%20kh%C3%A1c>.
5. Grey (2023). "Docker và các kiến thức căn bản" Truy cập từ Viblo. URL: <https://viblo.asia/p/docker-va-cac-kien-thuc-can-ban-BQyJKjEb4Me>.