

Họ tên: Trần Đình Văn

Msv: 20210970

Bùi Anh Minh

20210931

Nguyễn Văn Phụng Vĩ

20211104

Lớp: DCCNTT12.10.4

CÔNG NGHỆ PHẦN MỀM

TOOL

"Tool" (công cụ) trong công nghệ phần mềm là các chương trình, phần mềm hoặc ứng dụng được sử dụng để hỗ trợ quá trình phát triển, kiểm thử, triển khai, và quản lý phần mềm. Những công cụ này giúp tự động hóa nhiều công việc phức tạp, giảm thiểu lỗi do con người gây ra và tăng cường hiệu quả làm việc.

Dưới đây là một số tool được sử dụng phổ biến:

1. Git

Git là một hệ thống quản lý phiên bản phân tán (Distributed Version Control System - DVCS) phổ biến, được sử dụng để theo dõi các thay đổi trong mã nguồn phần mềm. Nó cho phép nhiều nhà phát triển làm việc cùng nhau trên cùng một dự án mà không lo về việc ghi đè hoặc mất mát mã nguồn.

Git được tạo ra bởi **Linus Torvalds** vào năm 2005, nhằm hỗ trợ phát triển nhân Linux. Hiện nay, Git đã trở thành công cụ chuẩn mực trong việc quản lý mã nguồn cho các dự án từ nhỏ đến lớn.

Các khái niệm cơ bản trong Git

- **Repository (Repo):** Là nơi lưu trữ tất cả các tập tin và lịch sử thay đổi của một dự án.
- **Commit:** Một đơn vị lưu trữ thay đổi trong Git. Mỗi commit lưu lại trạng thái của dự án tại một thời điểm.
- **Branch:** Một nhánh trong Git cho phép bạn phát triển song song các tính năng mà không ảnh hưởng đến nhánh chính.
- **Merge:** Hợp nhất các thay đổi từ một nhánh này sang nhánh khác.
- **Clone:** Tạo bản sao của một repository về máy cục bộ.
- **Push:** Đẩy các thay đổi từ máy cục bộ lên repository trên máy chủ (như GitHub, GitLab).
- **Pull:** Kéo các thay đổi từ repository trên máy chủ về máy cục bộ.

Cách sử dụng Git

a. Cài đặt Git

Bạn có thể tải Git từ git-scm.com và cài đặt theo hướng dẫn.

b. Thiết lập Git lần đầu sau khi cài đặt

Trước khi sử dụng Git, bạn cần cấu hình tên người dùng và email để gắn liền với các thay đổi của mình:

```
git config --global user.name "Tên của bạn"  
git config --global user.email "email@example.com"
```

c. Khởi tạo một Git Repository

Để bắt đầu theo dõi một dự án, bạn cần khởi tạo một repo Git trong thư mục dự án của mình:

```
cd ten-thu-muc-du-an  
git init
```

Lệnh này sẽ tạo một thư mục ẩn .git trong thư mục dự án, chứa toàn bộ thông tin và lịch sử của repo.

d. Thêm tập tin vào repo và thực hiện commit

Sau khi khởi tạo repo, bạn có thể thêm các tập tin vào và thực hiện commit:

```
git add <ten-tap-tin> # Hoặc git add . để thêm tất cả các tập tin thay  
đổi  
git commit -m "Thông điệp commit"
```

Lệnh git add thêm các tập tin vào khu vực chờ (staging area), sau đó lệnh git commit sẽ lưu các thay đổi vào repo kèm theo thông điệp mô tả ngắn gọn.

e. Làm việc với các nhánh (Branch)

Git cho phép làm việc với các nhánh khác nhau, giúp bạn phát triển song song nhiều tính năng mà không ảnh hưởng đến mã nguồn chính:

- Tạo nhánh mới:

```
git branch ten-nhanh-moi
```

- Chuyển sang nhánh mới:

git checkout ten-nhanh-moi

- Hoặc tạo và chuyển nhánh trong một lệnh:

git checkout -b ten-nhanh-moi

f. Gộp (merge) nhánh

Khi tính năng trên nhánh đã hoàn thiện, bạn có thể gộp nhánh đó vào nhánh chính (thường là master hoặc main):

git checkout master # Chuyển về nhánh chính
git merge ten-nhanh-moi

g. Đồng bộ với repository từ xa

Khi làm việc với GitHub hoặc GitLab, bạn có thể kết nối và đồng bộ với repository từ xa.

- Kết nối repository từ xa:

git remote add origin https://github.com/username/repo.git

- Đẩy các thay đổi lên repository từ xa:

git push -u origin master # Lần đầu, sau đó chỉ cần git push

- Kéo các thay đổi từ repository từ xa về:

git pull origin master

h. Xem trạng thái và lịch sử commit

- Xem trạng thái các tập tin trong repo:

git status

- Xem lịch sử các commit:

git log

i. Clone một repo có sẵn

Nếu bạn muốn sao chép một dự án từ GitHub hoặc GitLab về máy, bạn có thể dùng lệnh git clone:

```
git clone https://github.com/username/repo.git
```

Quy trình làm việc phổ biến với Git

1. Clone repo từ repository từ xa về máy.
2. Tạo nhánh mới để phát triển tính năng hoặc sửa lỗi.
3. Thực hiện commit mỗi khi có thay đổi quan trọng.
4. Merge nhánh mới vào nhánh chính sau khi hoàn thiện tính năng.
5. Push các thay đổi lên repository từ xa.

Git giúp quản lý mã nguồn hiệu quả, cho phép cộng tác nhóm một cách trơn tru, đặc biệt là khi dự án có nhiều thành viên cùng làm việc.

2. Jira

Jira là một công cụ quản lý dự án và theo dõi lỗi (issue tracker) được phát triển bởi Atlassian. Ban đầu, Jira được thiết kế để theo dõi và quản lý lỗi trong quá trình phát triển phần mềm, nhưng hiện nay nó đã phát triển thành một nền tảng quản lý dự án toàn diện, phù hợp với nhiều phương pháp quản lý như Agile, Scrum, Kanban. Jira được sử dụng rộng rãi trong các công ty phát triển phần mềm và công nghệ nhờ khả năng tùy biến cao, hỗ trợ đa dự án, và khả năng tích hợp với nhiều công cụ khác.

- Các tính năng chính của Jira

Quản lý nhiệm vụ (Issue Management): Issue là đơn vị cơ bản trong Jira, có thể là lỗi (bug), nhiệm vụ (task), câu chuyện (story), hoặc epic. Mỗi issue có thông tin chi tiết như mô tả, ưu tiên, người chịu trách nhiệm, và trạng thái.

Quản lý dự án (Project Management): Jira cho phép bạn tạo và quản lý nhiều dự án cùng một lúc. Mỗi dự án có thể có quy trình làm việc (workflow) riêng biệt, theo dõi nhiệm vụ và tiến độ một cách linh hoạt.

Hỗ trợ Agile: Jira được biết đến với khả năng quản lý dự án Agile tốt, hỗ trợ các phương pháp Scrum và Kanban. Bạn có thể tạo backlog, quản lý sprint, theo dõi tiến độ qua các bảng Scrum hoặc Kanban.

Báo cáo và phân tích (Reporting and Analytics): Jira cung cấp nhiều loại báo cáo như burndown chart, velocity chart, và sprint report để theo dõi hiệu suất làm việc của đội ngũ.

Tích hợp với các công cụ khác: Jira tích hợp với nhiều công cụ phát triển khác như **Confluence** (công cụ wiki), **Bitbucket**, **GitHub**, **Slack**, và các công cụ CI/CD như **Jenkins**.

- Cách sử dụng công cụ Jira

a) Đăng ký và cài đặt Jira

- Bạn có thể đăng ký tài khoản Jira tại [Atlassian.com](https://atlassian.com) và chọn phiên bản phù hợp (cloud hoặc server).
- Sau khi đăng ký, bạn sẽ tạo không gian làm việc (workspace) và dự án đầu tiên để bắt đầu.

b) Tạo và cấu hình dự án mới

- Sau khi đăng nhập, bạn có thể tạo một **dự án mới** bằng cách chọn "Create Project". Bạn sẽ cần chọn loại dự án phù hợp với phương pháp quản lý mà bạn sử dụng, chẳng hạn như **Scrum**, **Kanban**, hoặc **dự án cơ bản**.
- Jira sẽ tự động tạo các bảng (board), workflow, và các tính năng đi kèm tương ứng với loại dự án mà bạn chọn.

c) Tạo và quản lý Issue

- Mỗi dự án trong Jira sẽ có một loạt các **issue** đại diện cho các nhiệm vụ, lỗi, hoặc yêu cầu mới.
- Bạn có thể tạo issue mới bằng cách nhấn vào "Create" trên thanh điều hướng.
- Khi tạo issue, bạn cần cung cấp thông tin như:
 - **Type:** Loại issue (task, bug, story, epic).
 - **Summary:** Tóm tắt ngắn gọn về nhiệm vụ.
 - **Description:** Mô tả chi tiết về công việc cần làm.
 - **Assignee:** Người chịu trách nhiệm hoàn thành.
 - **Priority:** Mức độ ưu tiên của nhiệm vụ.

d) Quản lý Backlog và lập kế hoạch Sprint (dành cho dự án Scrum)

- Trong dự án Scrum, **backlog** lưu trữ các issue chưa được thực hiện. Bạn có thể sắp xếp thứ tự ưu tiên cho các issue trong backlog dựa trên yêu cầu và chiến lược của dự án.

- Để bắt đầu một **Sprint**, bạn chọn các nhiệm vụ từ backlog và đưa chúng vào **Sprint** mới. Sau đó, Sprint sẽ được theo dõi thông qua bảng Scrum, nơi các nhiệm vụ được di chuyển qua các cột trạng thái như “To Do”, “In Progress”, và “Done”.

e) Quản lý tiến độ qua Kanban/Scrum Board

- **Scrum Board:** Trong mô hình Scrum, bảng này sẽ hiển thị trạng thái của các issue trong sprint, giúp theo dõi tiến độ của nhóm trong từng giai đoạn của sprint.
- **Kanban Board:** Trong mô hình Kanban, bảng này theo dõi toàn bộ quá trình làm việc với các cột trạng thái linh hoạt. Bạn có thể thêm hoặc điều chỉnh cột để phù hợp với quy trình làm việc của dự án.

f) Theo dõi tiến độ và báo cáo

- Jira cung cấp nhiều loại **báo cáo** giúp nhóm theo dõi tiến độ làm việc:
 - **Burndown Chart:** Hiển thị số lượng công việc còn lại trong sprint.
 - **Velocity Chart:** Đo lường lượng công việc hoàn thành trong mỗi sprint.
 - **Sprint Report:** Tóm tắt công việc hoàn thành và những gì chưa hoàn thành trong sprint.
- Các báo cáo này giúp đội ngũ đánh giá hiệu suất và cải tiến quy trình làm việc.

g) Tích hợp với các công cụ khác

- Bạn có thể tích hợp Jira với các công cụ phát triển như **GitHub**, **Bitbucket**, và các công cụ CI/CD để theo dõi các thay đổi mã nguồn và triển khai ứng dụng trực tiếp từ Jira.

h) Tùy chỉnh quy trình làm việc (Workflow)

- Mỗi dự án có thể có quy trình làm việc riêng, cho phép điều chỉnh các bước từ khi tạo đến khi hoàn thành một nhiệm vụ. Bạn có thể thay đổi quy trình làm việc bằng cách vào **Project Settings > Workflows**, thêm hoặc chỉnh sửa các trạng thái và quy tắc chuyển trạng thái.

Quy trình làm việc điển hình với Jira (Agile)

1. **Tạo backlog:** Đội ngũ tạo các issue (task, bug, story) và lưu trữ chúng trong backlog.
2. **Lập kế hoạch sprint:** Chọn các nhiệm vụ từ backlog để đưa vào sprint mới.
3. **Thực hiện sprint:** Theo dõi tiến độ của các issue qua bảng Scrum hoặc Kanban.
4. **Báo cáo và phân tích:** Sử dụng các báo cáo như burndown chart hoặc velocity chart để đánh giá tiến độ và hiệu suất.
5. **Đánh giá sprint và cải tiến:** Sau mỗi sprint, nhóm sẽ họp để đánh giá kết quả và cải thiện quy trình làm việc cho các sprint tiếp theo.

Jira là một công cụ rất mạnh mẽ và linh hoạt trong việc quản lý dự án, đặc biệt là đối với các dự án phát triển phần mềm theo phương pháp Agile. Với khả năng tùy chỉnh cao, tích hợp mạnh mẽ với nhiều công cụ khác và hỗ trợ quản lý công việc một cách trực quan, Jira giúp các nhóm làm việc hiệu quả và cải thiện năng suất.