



Phần 1: Giới thiệu về phát triển phần mềm GIS nguồn mở

Đỗ Thành Long
dtlong@opengis.vn

OPENGIS

Discover the world, Learn with maps


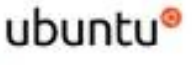















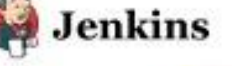























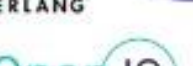
















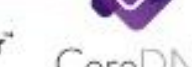



















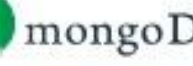








 <https://opengis.vn>

Nội dung

- ✓ Phần mềm mã nguồn mở là gì?
- ✓ Một số giấy phép nguồn mở phổ biến
- ✓ Tổng quan về phần mềm GIS nguồn mở và những ưu điểm của nó
- ✓ Giới thiệu các khái niệm và phương pháp phát triển phần mềm
- ✓ Thiết lập môi trường phát triển (IDE, thư viện, v.v.)
- ✓ Giới thiệu về hệ thống kiểm soát phiên bản (Git)

1. Phần mềm mã nguồn mở là gì?

Sample Open-Source Software for IBM Z and LinuxONE

Linux Distributions & Virtualization	Cloud & Container Services	Languages, Runtimes, Frameworks	DevOps/Automation	Big Data, Observability, Analytics
 Red Hat  ubuntu  SUSE  KVM	 docker  LXD  openstack  kubernetes  minikube	 python  Java  JS  R  Scala  Open Liberty	 CHEF  puppet  Jenkins  ANSIBLE  sonarqube	 Flink  Solr  splunk  kafka  Spark  fluentd  Ignite  Grafana
Linux Community Versions  debian  openSUSE  fedora  alpine  ClefOS  gentoo linux	 HELM  APACHE HTTP SERVER PROJECT  okd  OPENSIFT  docker Compose  MARATHON	 Ruby  node  GO  ERLANG  php  pypy  OpenJ9  RAILS  OpenJDK	 Maven  Gradle  SALTSTACK  Zowe  Travis CI  GitLab RUNNER	 logstash  elasticsearch  matomo  APACHE STORM  cassandra  kibana
Networking & Monitoring  NGINX  HAPROXY  Apache MESOS  Prometheus  ZABBIX  Apache ZooKeeper  CoreDNS  etcd  Caddy	 Sysdig  Terraform  kata containers  podman  KUBE ROUTER  Knative  Kong	 Perl  OCaml  Clojure  GCC  TensorFlow  BASH  Apache Tomcat  WildFly  JRuby	Middleware & others  Apache ACTIVEMQ  Istio  RabbitMQ  mule ESB  Apache Camel  APACHE GEODE  HIBERNATE  WORDPRESS  remine  Drupal	Databases & Storage  CouchDB  mongoDB  MariaDB  RethinkDB  redis  PostgreSQL  MySQL  GlusterFS  SCYLLA  Couchbase  influxdb  Memcached  CockroachDB

More: www.ibm.com/community/z/open-source-software/

1. Phần mềm mã nguồn mở là gì?

Phần
mềm mã
nguồn
mở

Mã nguồn được công khai

Cho phép cộng đồng xem, sửa đổi, phát triển tiếp

Phân phối theo các giấy phép mã nguồn mở

GPL (General Public License) hoặc MIT License



2. Một số giấy phép nguồn mở phổ biến

- Giấy phép GPL (General Public License)
- Giấy phép MIT (Massachusetts Institute of Technology)
- Giấy phép Apache
- Giấy phép BSD (Berkeley Software Distribution)

2. Một số giấy phép nguồn mở phổ biến



Giấy phép Công cộng GNU (tiếng Anh: *GNU General Public License*, viết tắt **GNU GPL** hay chỉ **GPL**) là giấy phép phần mềm tự do được sử dụng rộng rãi, đảm bảo cho người dùng cuối tự do chạy, nghiên cứu, sửa đổi và chia sẻ phần mềm

GPL yêu cầu rằng bất kỳ phần mềm sử dụng giấy phép này phải được phân phối kèm theo mã nguồn và giữ nguyên giấy phép khi phân phối lại

Giấy phép GNU hay GPL



2. Một số giấy phép nguồn mở phổ biến



Giấy phép MIT

Giấy phép MIT là một loại giấy phép sử dụng cho các **phần mềm mã nguồn mở** được phát triển dựa trên một loại giấy phép có nguồn gốc tại **Viện Công nghệ Massachusetts (MIT)**

Cho phép sử dụng, sao chép, sửa đổi và phân phối mã nguồn của phần mềm, bao gồm cả mã nguồn được sửa đổi

Sử dụng sử dụng và phát triển phần mềm mà không cần đáp ứng các yêu cầu đặc biệt



2. Một số giấy phép nguồn mở phổ biến



Giấy phép Apache

Giấy phép Apache là một giấy phép phần mềm tự do của **Quỹ Phần mềm Apache** (*Apache Software Foundation - ASF*).

Apache cung cấp tự do sử dụng, sao chép, sửa đổi và phân phối mã nguồn của phần mềm.

Nó cũng yêu cầu việc bảo lưu các thông báo về bản quyền và giấy phép trong phần mềm và tài liệu liên quan



2. Một số giấy phép nguồn mở phổ biến

BSD

Giấy phép BSD

(Berkeley Software Distribution)

Giấy phép BSD ban đầu được sử dụng cho tên gọi của nó, Berkeley Software Distribution (BSD), một hệ điều hành tương tự Unix

Giấy phép BSD cung cấp tự do sử dụng, sao chép, sửa đổi và phân phối mã nguồn của phần mềm.

Cho phép người sử dụng sử dụng mã nguồn mà không cần tuân theo các yêu cầu nguyên tắc hoặc giấy phép cụ thể



2. Một số giấy phép nguồn mở phổ biến (Thông tin thêm về dự án GNU)



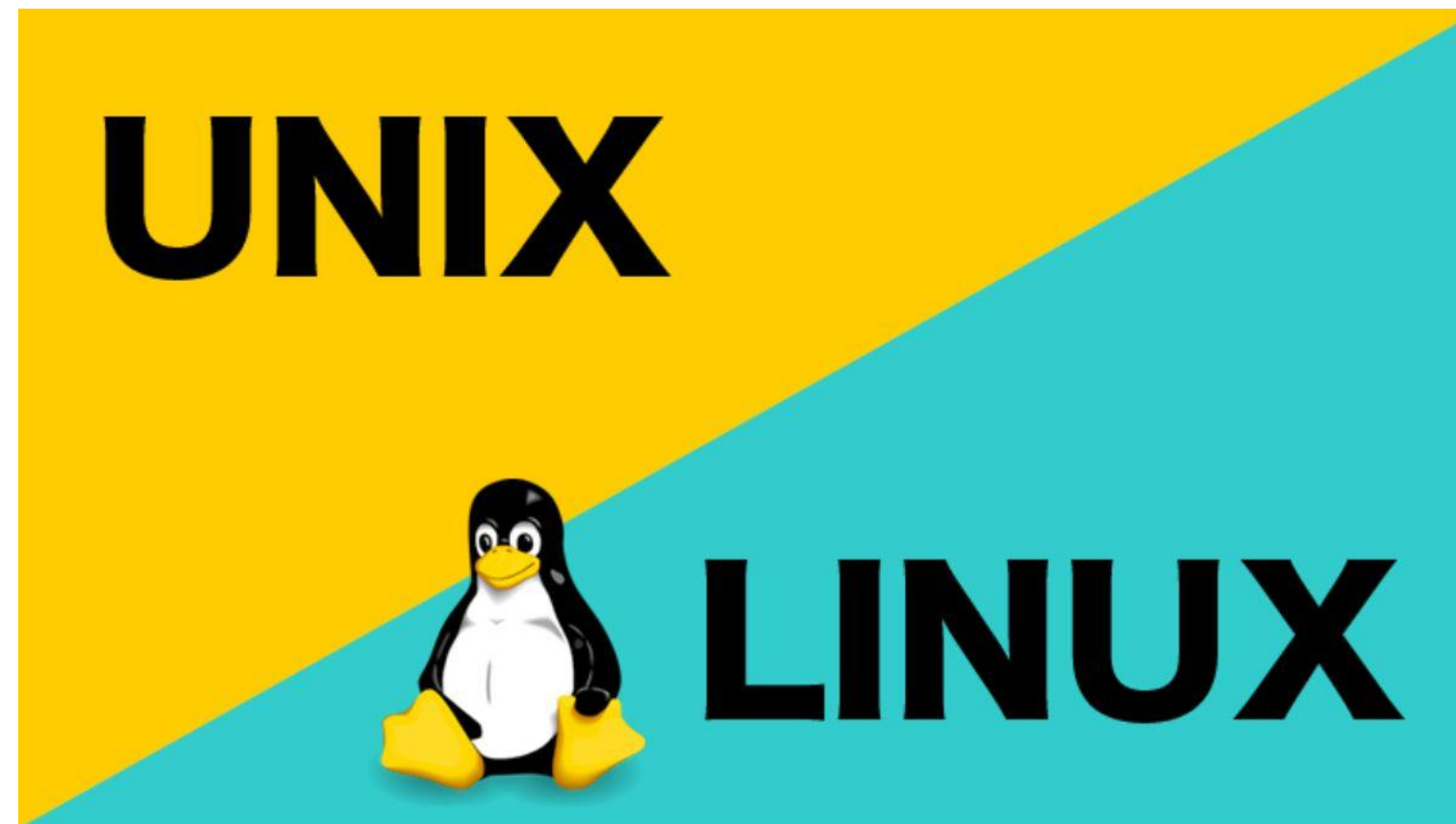
GNU

- Là một hệ điều hành* và bộ sưu tập phần mềm máy tính;
- GNU bao gồm toàn bộ phần mềm tự do, hầu hết được cấp phép theo General Public License (GPL) của GNU Project.

GNU là một kiểu viết tắt đệ quy của "**GNU's Not Unix!**"



2. Một số giấy phép nguồn mở phổ biến (Thông tin thêm về Unix, Linux)

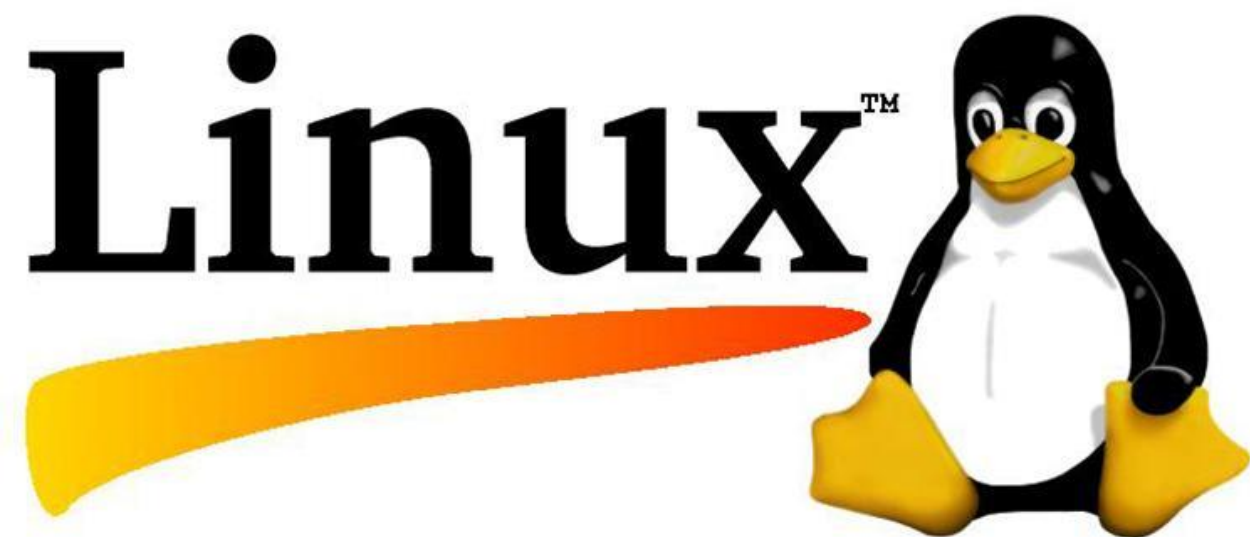


2. Một số giấy phép nguồn mở phổ biến (Thông tin thêm về Unix, Linux)



- Là hệ điều hành máy tính;
- **MÃ NGUỒN ĐÓNG**
- Hoạt động trên nhiều loại máy tính khác nhau: máy trạm, máy chủ và thiết bị nhúng
- Thiết kế đa người dùng
- Đa nhiệm
- Cung cấp môi trường, công cụ, tiện ích để phát triển phần mềm
- Tiền đề phát triển các biến thể như **Linux, FreeBSD, MacOS, Solaris, ..**

2. Một số giấy phép nguồn mở phổ biến (Thông tin thêm về Unix, Linux)



- Là hệ điều hành máy tính;
- Được phát triển bởi Linus Torvalds;
- **MÃ NGUỒN MỞ**
- Phát triển dựa trên các tiêu chuẩn và giao thức của Unix
- Dựa trên kernel Linux, một thành phần quan trọng của hệ điều hành, và sử dụng các thành phần từ dự án **GNU**



3. Tổng quan về phần mềm GIS nguồn mở



3. Tổng quan về phần mềm GIS nguồn mở



**GIS mã nguồn mở
(Open Source GIS)**

- Các phần mềm GIS được phát triển và phân phối dưới các giấy phép mã nguồn mở;
- Được công khai, cho phép người dùng xem, sửa đổi và phân phối lại theo các điều khoản của giấy phép mã nguồn mở



3. Tổng quan về phần mềm GIS nguồn mở

OGC[®]
Making location count.



3. Tổng quan về phần mềm GIS nguồn mở



Open Source Geospatial Foundation

- Là một tổ chức phi lợi nhuận toàn cầu
- Thành lập vào năm 2006

Mục tiêu thúc đẩy và hỗ trợ việc sử dụng phần mềm mã nguồn mở trong lĩnh vực GIS và địa lý

OSGeo quản lý và hỗ trợ nhiều dự án quan trọng trong lĩnh vực GIS mã nguồn mở, bao gồm **QGIS, GRASS GIS, GeoServer và PostGIS**.

OSGeo tổ chức các sự kiện, khóa học và hội thảo để tăng cường sự nhận thức và kiến thức về GIS mã nguồn mở.

3. Tổng quan về phần mềm GIS nguồn mở

OGC®

Making location count.

Open Geospatial Consortium

- Là một tổ chức tiêu chuẩn hóa quốc tế
- Thành lập vào năm 1994

Mục tiêu xác định và duy trì các tiêu chuẩn cho việc tương thích và tương tác giữa các hệ thống GIS khác nhau

Để làm gì?

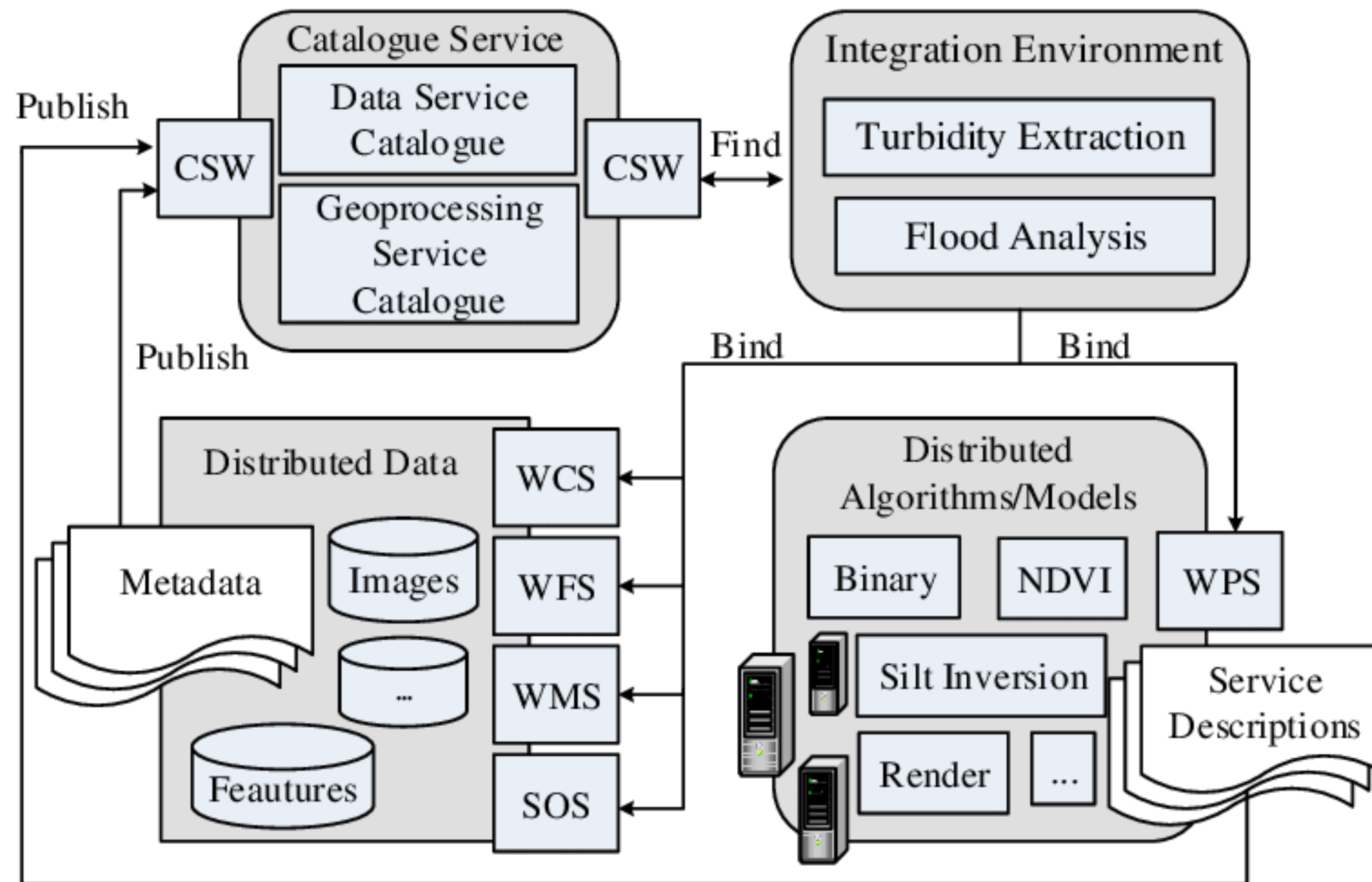


3. Tổng quan về phần mềm GIS nguồn mở

OGC Standards

Tiêu chuẩn mở trong GIS

- Đảm bảo tính tương thích và tương tác giữa các hệ thống và dữ liệu địa lý;
- Chia sẻ và sử dụng dữ liệu địa lý một cách linh hoạt và tiện lợi.



3. Tổng quan về phần mềm GIS nguồn mở

OGC

Standards

Tiêu chuẩn mở trong GIS

- **Geography Markup Language (GML):** GML là một tiêu chuẩn định dạng dữ liệu địa lý mã nguồn mở dựa trên XML. Nó cho phép mô tả và truyền tải dữ liệu địa lý dưới dạng văn bản có cấu trúc, bao gồm đối tượng địa lý, hệ thống tọa độ và thông tin thuộc tính.
- **Keyhole Markup Language (KML):** KML là một tiêu chuẩn định dạng dữ liệu địa lý mã nguồn mở được sử dụng để mô tả và trình bày dữ liệu địa lý trong các ứng dụng như Google Earth và Google Maps. Nó cho phép hiển thị các đối tượng địa lý, hình ảnh và thông tin thuộc tính.



OGC Standards

Tiêu chuẩn mở trong GIS

3. Tổng quan về phần mềm GIS nguồn mở

- **Web Map Service (WMS):** WMS là một tiêu chuẩn cho phép truy xuất và hiển thị bản đồ tĩnh dưới dạng hình ảnh trên web. Nó cho phép người dùng yêu cầu và tải xuống các lớp bản đồ từ máy chủ WMS thông qua các yêu cầu HTTP.
- **Web Feature Service (WFS):** WFS là một tiêu chuẩn cho phép truy xuất và chỉnh sửa dữ liệu địa lý không gian trên web. Nó cho phép người dùng truy vấn, tìm kiếm và cập nhật đối tượng địa lý thông qua giao thức HTTP.
- **Web Coverage Service (WCS):** WCS là một tiêu chuẩn cho phép truy xuất và truyền tải dữ liệu địa lý dưới dạng dữ liệu vùng phủ (coverage data) trên web. Nó cung cấp khả năng truy vấn và tải xuống các dữ liệu liên quan đến vùng phủ, chẳng hạn như hình ảnh vệ tinh hoặc dữ liệu radar.
- **Tile Map Service (TMS):** TMS là một tiêu chuẩn cho phép tải xuống và hiển thị các dữ liệu địa lý dưới dạng các ô vuông nhỏ gọi là tile. Nó cho phép tải xuống chỉ các phần của bản đồ cần thiết để hiển thị trên màn hình giúp tăng tốc độ hiển thị và tiết kiệm băng thông

4. Các khái niệm và phương pháp phát triển phần mềm



4. Các khái niệm và phương pháp phát triển phần mềm

Là một khung làm việc hoặc một quy trình chuẩn để tổ chức, lập kế hoạch và triển khai quá trình phát triển phần mềm.

Định nghĩa các bước, quy trình và vai trò của các thành viên trong đội làm việc phát triển phần mềm.

Có thể được tùy chỉnh và điều chỉnh phù hợp với yêu cầu và tình huống cụ thể của dự án phát triển phần mềm. Có nhiều mô hình phát triển phần mềm phổ biến như Waterfall, Agile, Spiral, Incremental, V-Model và DevOps.

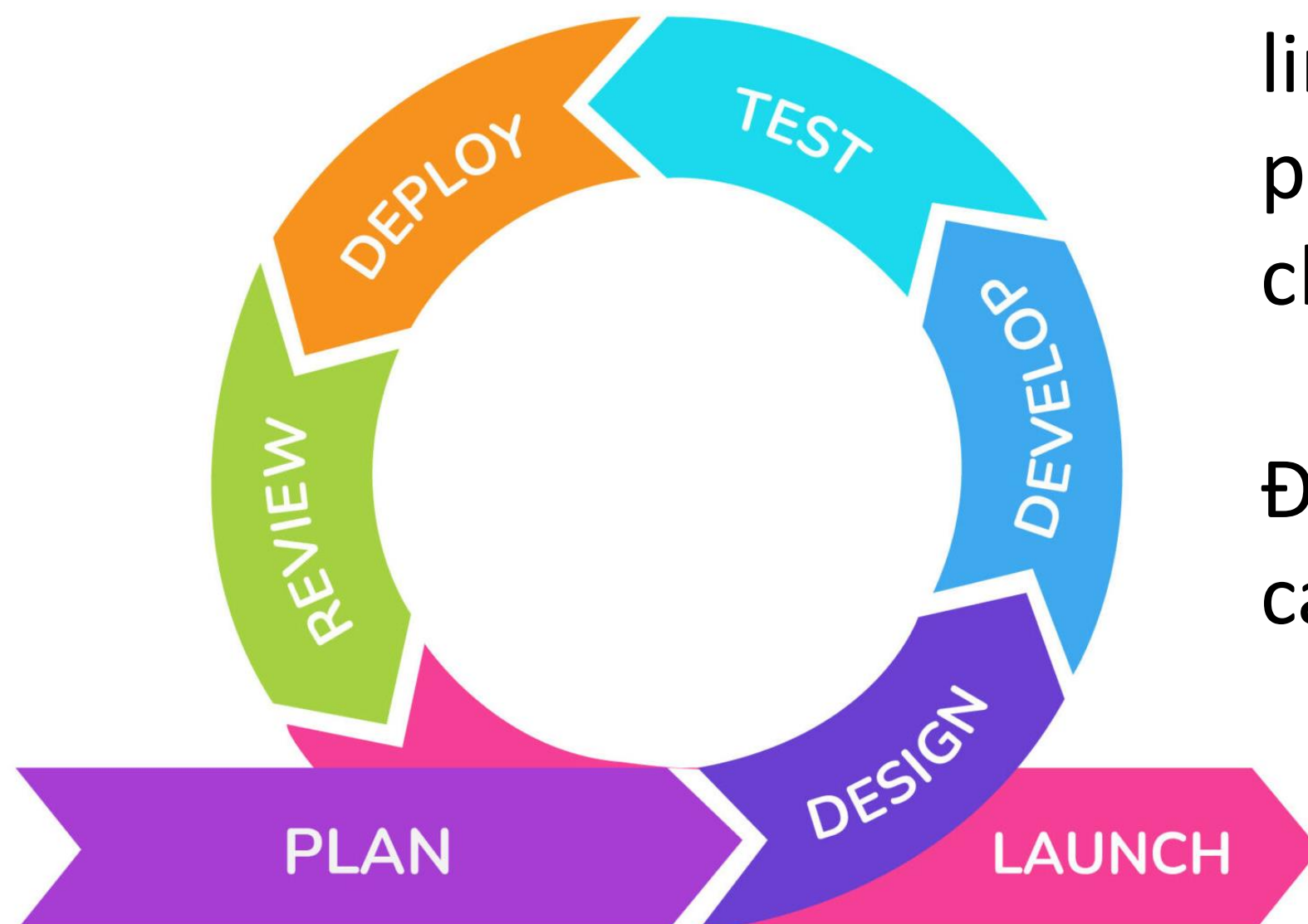
Mục tiêu của mô hình phát triển phần mềm là đảm bảo sự tổ chức, hiệu quả và chất lượng trong quá trình phát triển phần mềm.

Mô hình phát triển phần mềm



4. Các khái niệm và phương pháp phát triển phần mềm

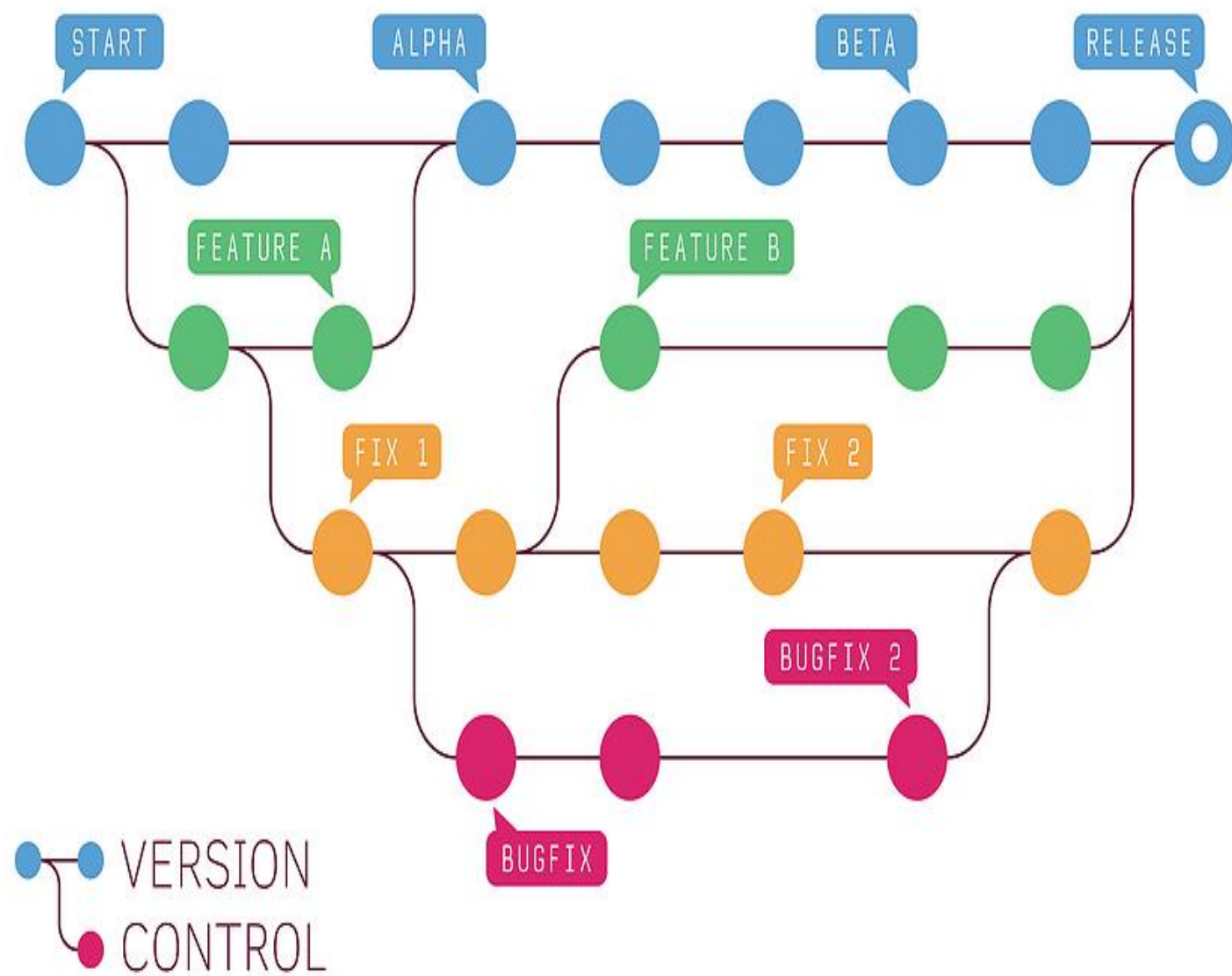
AGILE



Đây là một phương pháp phát triển phần mềm linh hoạt và tập trung vào việc phát triển sản phẩm một cách linh hoạt và phản hồi nhanh chóng.

Đảm bảo sự linh hoạt và khả năng thích ứng với các yêu cầu và thay đổi trong quá trình phát triển.

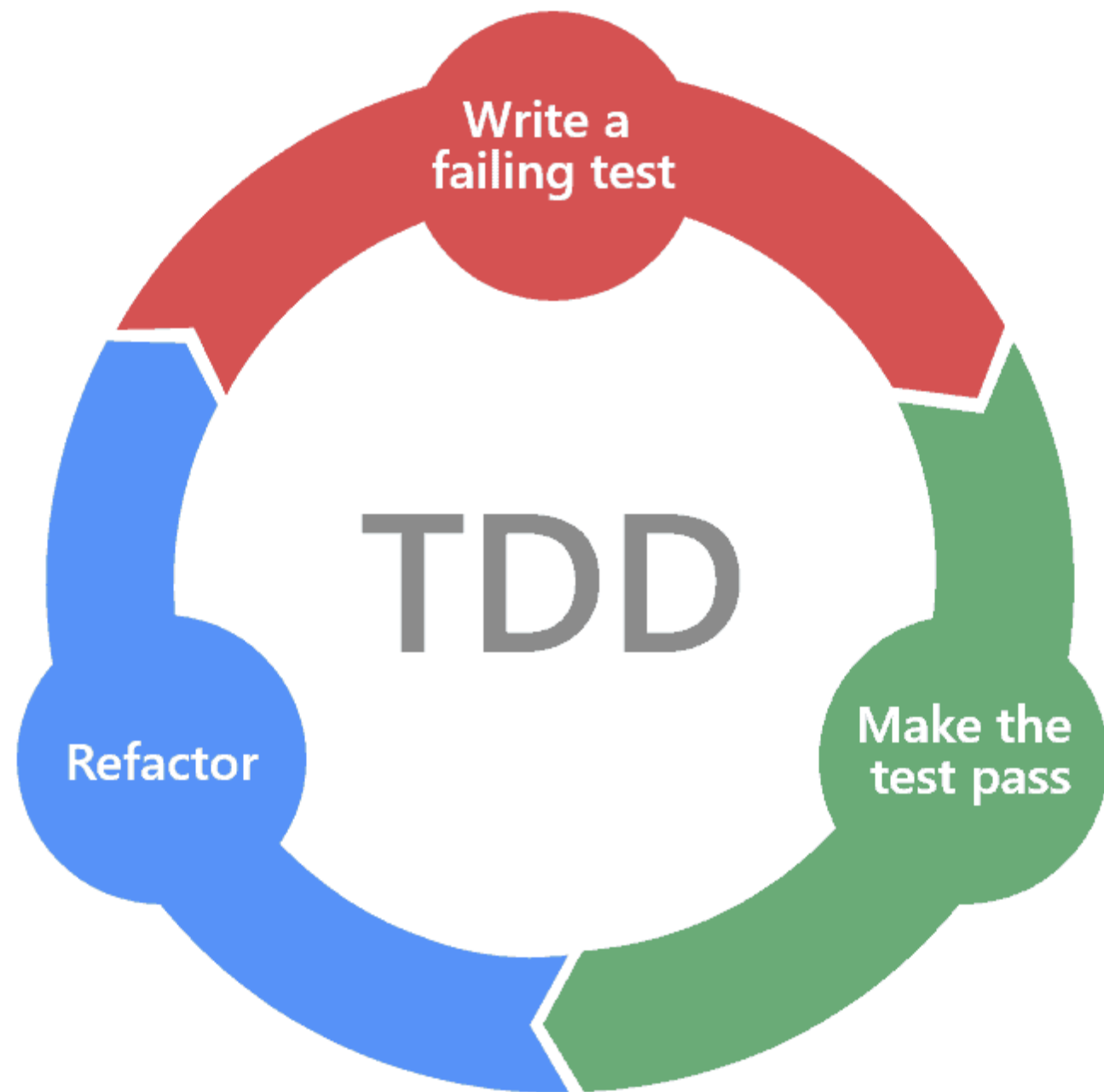
4. Các khái niệm và phương pháp phát triển phần mềm



Hệ thống quản lý phiên bản giúp theo dõi và quản lý các phiên bản khác nhau của mã nguồn phần mềm.

Nó cho phép lưu trữ, quản lý và theo dõi các thay đổi được thực hiện trong quá trình phát triển và giúp đảm bảo tính nhất quán và an toàn của mã nguồn.

4. Các khái niệm và phương pháp phát triển phần mềm

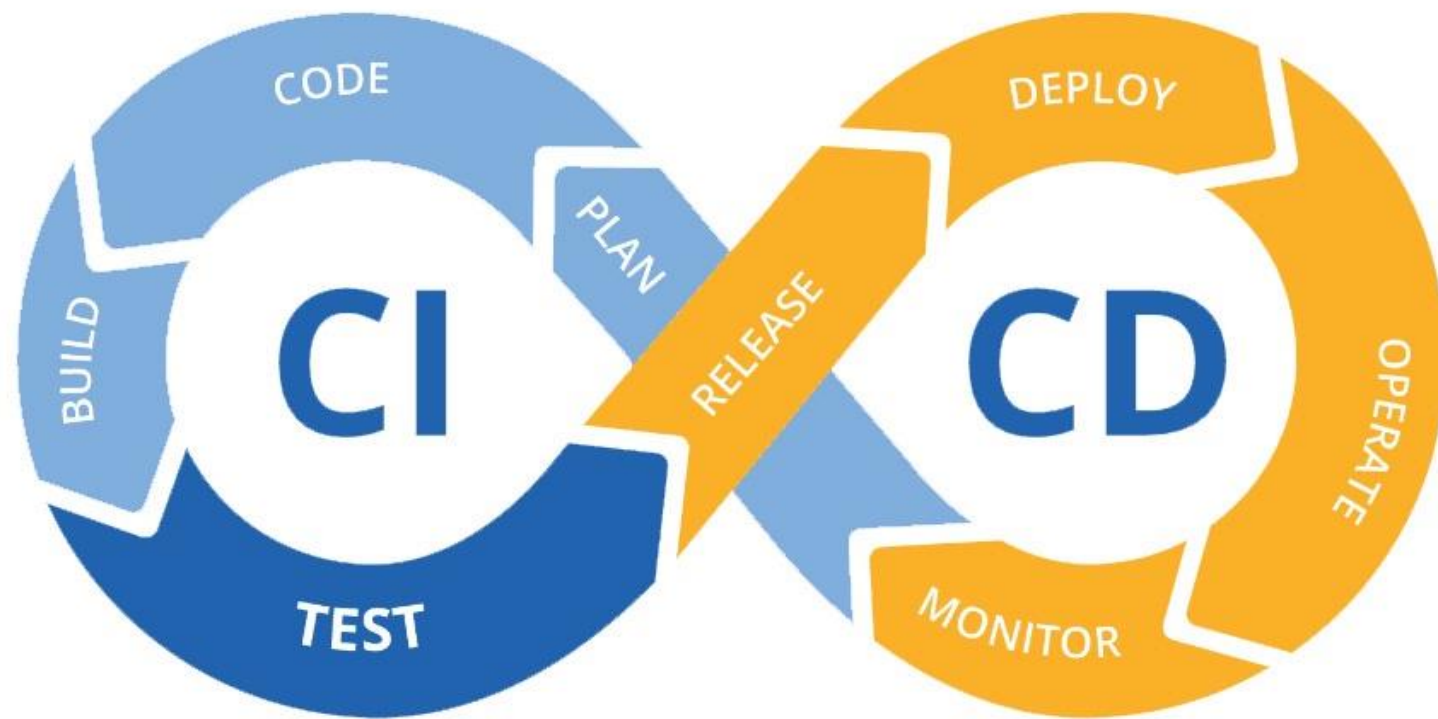


Đây là một phương pháp phát triển phần mềm mà trong đó việc viết các bài kiểm tra (test) được thực hiện trước khi triển khai mã nguồn.

Đảm bảo tính ổn định và chất lượng của phần mềm bằng cách tập trung vào việc xác định yêu cầu và viết các bài kiểm tra phù hợp.



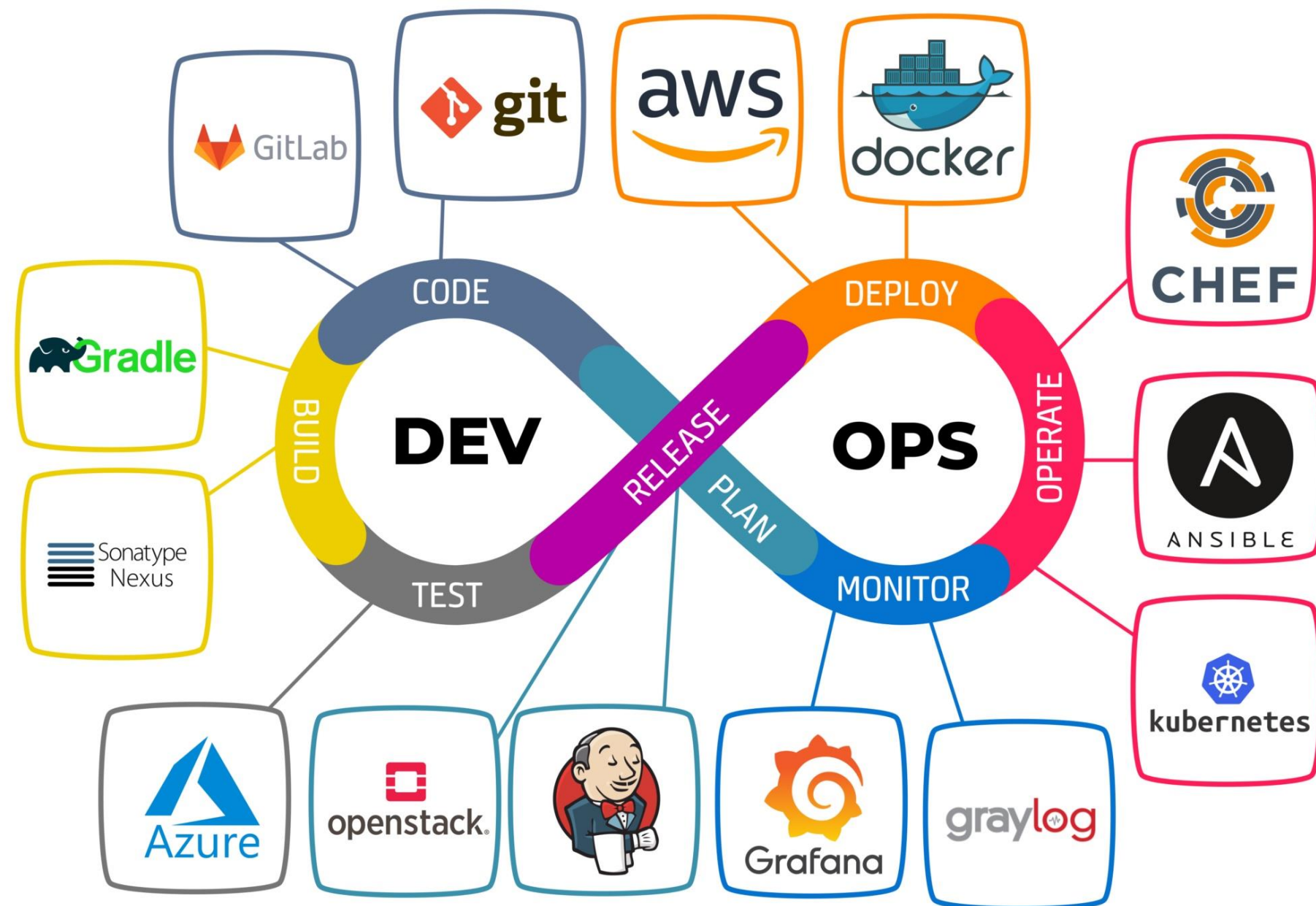
4. Các khái niệm và phương pháp phát triển phần mềm



- Continuous Integration (CI)
- Continuous Delivery (CD)
- Continuous Deployment (CD)

Đây là một quy trình tự động hóa trong phát triển phần mềm, trong đó các thành viên trong nhóm phát triển đồng bộ hóa và tích hợp mã nguồn của họ vào một nhánh chung. Nó giúp giảm thiểu xung đột và xác định sớm các lỗi tích hợp trong quá trình phát triển

4. Các khái niệm và phương pháp phát triển phần mềm



DevOps là viết tắt của Development và Operations

Là một phương pháp kết hợp quy trình phát triển phần mềm và quản lý hệ thống.

Nhấn mạnh sự hợp tác và tương tác giữa các nhóm phát triển và vận hành để đạt được sự linh hoạt và tương thích tốt hơn giữa phần mềm và hạ tầng.

4. Các khái niệm và phương pháp phát triển phần mềm

CODE REVIEW



Created by Gan Khoon Lay
from Noun Project

```
function register()
{
    if (!empty($_POST)) {
        $msg = '';
        if ($_POST['user_name']) {
            if ($_POST['user_password_new']) {
                if ($_POST['user_password_new'] === $_POST['user_password_
                if (strlen($_POST['user_password_new']) > 5) {
                    if (strlen($_POST['user_name']) < 65 && strlen($_P
                    if (preg_match('/^[a-z\d]{2,64}$/i', $_POST['u
                        $user = read_user($_POST['user_name']);
                        if (!isset($user['user_name'])) {
                            if ($_POST['user_email']) {
                                if (strlen($_POST['user_email']) <
                                    if (filter_var($_POST['user_en
                                        create_user();
                                        $_SESSION['msg'] = 'You ar
                                        header('Location: ' . $_SE
                                        exit();
                                    } else $msg = 'You must provid
                                } else $msg = 'Email must be less
                            } else $msg = 'Email cannot be empty';
                        } else $msg = 'Username already exists';
                    } else $msg = 'Username must be only a-z, A-Z,
                } else $msg = 'Username must be between 2 and 64 c
            } else $msg = 'Password must be at least 6 characters'
        } else $msg = 'Passwords do not match';
    } else $msg = 'Empty Password';
} else $msg = 'Empty Username';
$_SESSION['msg'] = $msg;
}
return register_form();
}
```

Đây là quá trình kiểm tra và đánh giá mã nguồn phần mềm bởi các thành viên khác trong nhóm phát triển.

Nó giúp đảm bảo tính chất lượng và tuân thủ các tiêu chuẩn mã hóa, cũng như cung cấp phản hồi và cải thiện chất lượng của mã nguồn.

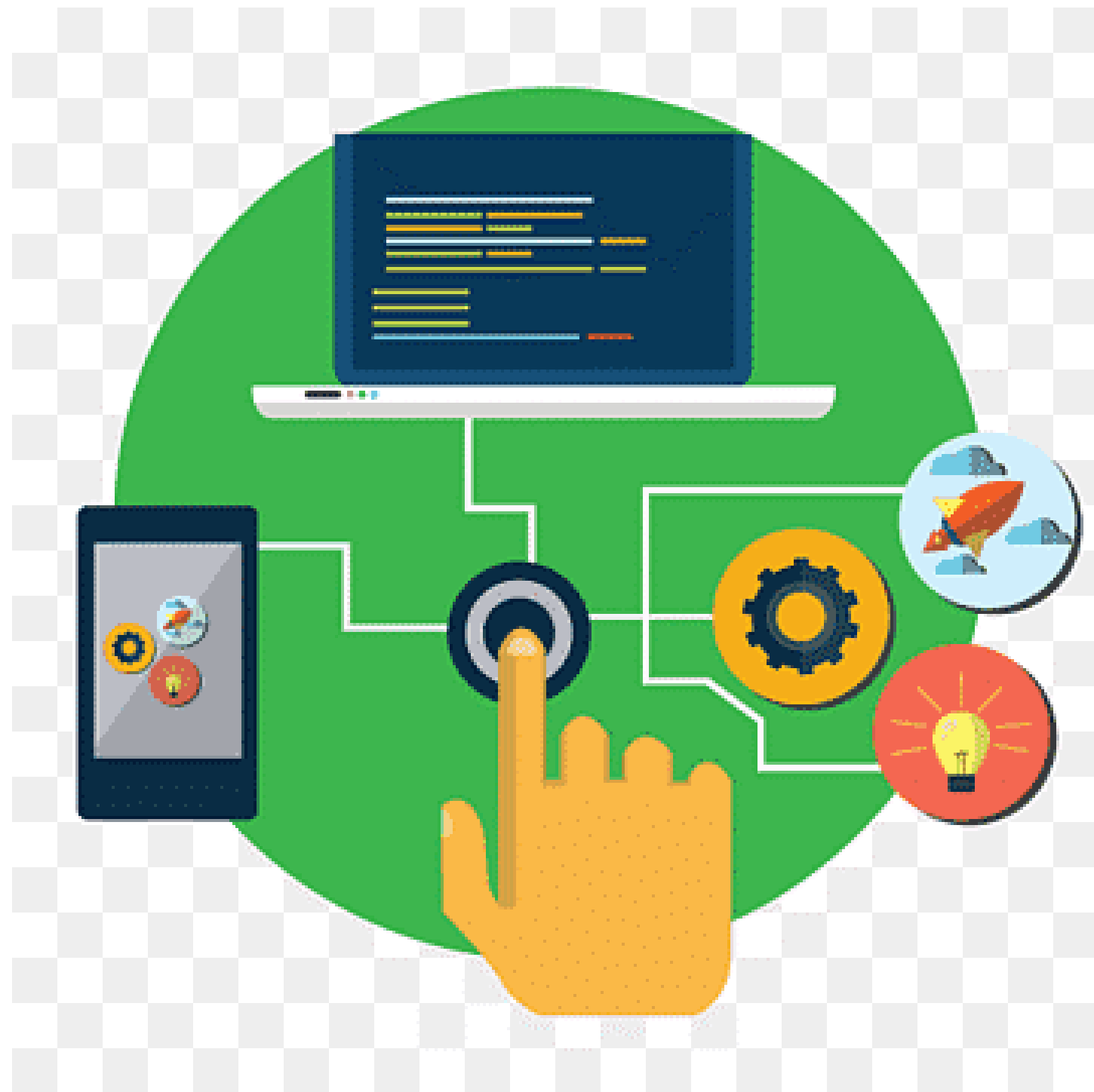
4. Các khái niệm và phương pháp phát triển phần mềm

- **Quy trình phát triển phần mềm:** Đây là quy trình toàn diện để phát triển phần mềm từ những ý tưởng ban đầu đến việc triển khai và duy trì sản phẩm. Các quy trình phát triển phần mềm phổ biến bao gồm mô hình thác nước, mô hình phát triển linh hoạt (Agile), và mô hình phát triển phần mềm hướng đối tượng (OOP).
- **Phân tích yêu cầu:** Đây là quá trình xác định và hiểu rõ các yêu cầu và mong muốn của khách hàng để xây dựng phần mềm. Phân tích yêu cầu bao gồm việc thu thập thông tin, phân loại và ưu tiên yêu cầu, và xác định phạm vi và ràng buộc của sản phẩm.
- **Thiết kế phần mềm:** Quá trình thiết kế phần mềm nhằm tạo ra kiến trúc và kế hoạch chi tiết cho phần mềm. Nó bao gồm thiết kế giao diện người dùng, thiết kế cơ sở dữ liệu, thiết kế kiến trúc phần mềm và các thành phần khác.

4. Các khái niệm và phương pháp phát triển phần mềm

- **Lập trình:** Đây là quá trình viết mã nguồn của phần mềm bằng ngôn ngữ lập trình. Lập trình bao gồm việc triển khai thiết kế, kiểm thử, và tạo ra các chức năng và tính năng của phần mềm.
- **Kiểm thử và Debugging:** Quá trình kiểm thử và debugging nhằm đảm bảo rằng phần mềm hoạt động đúng và không có lỗi. Kiểm thử có thể bao gồm kiểm thử đơn vị (unit testing), kiểm thử tích hợp (integration testing), kiểm thử hệ thống (system testing), và kiểm thử chấp nhận (acceptance testing).
- **Triển khai:** Quá trình triển khai và triển khai nhằm cài đặt và triển khai phần mềm vào môi trường thực tế. Nó bao gồm việc cấu hình hệ thống, chuyển giao sản phẩm cho khách hàng và hỗ trợ trong quá trình triển khai.
- **Bảo trì và nâng cấp:** Sau khi phần mềm đã được triển khai, quá trình bảo trì và nâng cấp nhằm duy trì và cải thiện phần mềm. Nó bao gồm việc sửa lỗi, cập nhật tính năng mới và tối ưu hóa hiệu suất.

5. Thiết lập môi trường phát triển (IDE, thư viện, v.v.)



Mục đích?

Làm như thế nào?



5. Thiết lập môi trường phát triển (IDE, thư viện, v.v.)

Hiệu suất làm việc: Môi trường phát triển phần mềm được tinh chỉnh và cấu hình để đáp ứng yêu cầu của quá trình phát triển. Điều này giúp tăng hiệu suất làm việc của nhóm phát triển và làm việc hiệu quả hơn.

Quản lý phiên bản: Môi trường phát triển phần mềm bao gồm công cụ quản lý phiên bản như Git, SVN, Mercurial,... giúp theo dõi và kiểm soát phiên bản của mã nguồn. Điều này giúp đảm bảo rằng các thay đổi và cải tiến được quản lý một cách cẩn thận và tiện lợi.

Kiểm thử và gỡ lỗi: Môi trường phát triển phần mềm cung cấp công cụ kiểm thử và gỡ lỗi để giúp phát hiện và khắc phục các lỗi và vấn đề trong quá trình phát triển. Điều này giúp đảm bảo tính ổn định và chất lượng của ứng dụng.

Mục đích



5. Thiết lập môi trường phát triển (IDE, thư viện, v.v.)

Quản lý dự án: Môi trường phát triển phần mềm cung cấp các công cụ quản lý dự án để theo dõi tiến độ, quản lý tài nguyên và phối hợp công việc. Điều này giúp đảm bảo rằng dự án được quản lý một cách hiệu quả và hoàn thành thành công.

Tích hợp và triển khai: Môi trường phát triển phần mềm cho phép tích hợp các công cụ và quy trình triển khai để tạo ra các phiên bản ứng dụng chính xác và triển khai chúng một cách dễ dàng và an toàn.

Tạo tài liệu: Môi trường phát triển phần mềm cung cấp các công cụ hỗ trợ tạo và quản lý tài liệu dự án. Điều này giúp đảm bảo rằng tài liệu liên quan đến dự án được tạo ra và duy trì một cách dễ dàng.

Mục đích



5. Thiết lập môi trường phát triển (IDE, thư viện, v.v.)

Máy tính và phần cứng: Đảm bảo máy tính và phần cứng của bạn đáp ứng yêu cầu của quá trình phát triển phần mềm. Điều này bao gồm bộ vi xử lý mạnh mẽ, bộ nhớ đủ, đĩa cứng lưu trữ đủ lớn và màn hình đủ rộng để làm việc hiệu quả.

Hệ điều hành: Lựa chọn hệ điều hành phù hợp với các công nghệ và công cụ phát triển mà bạn sử dụng. Ví dụ: Windows, macOS, Linux.

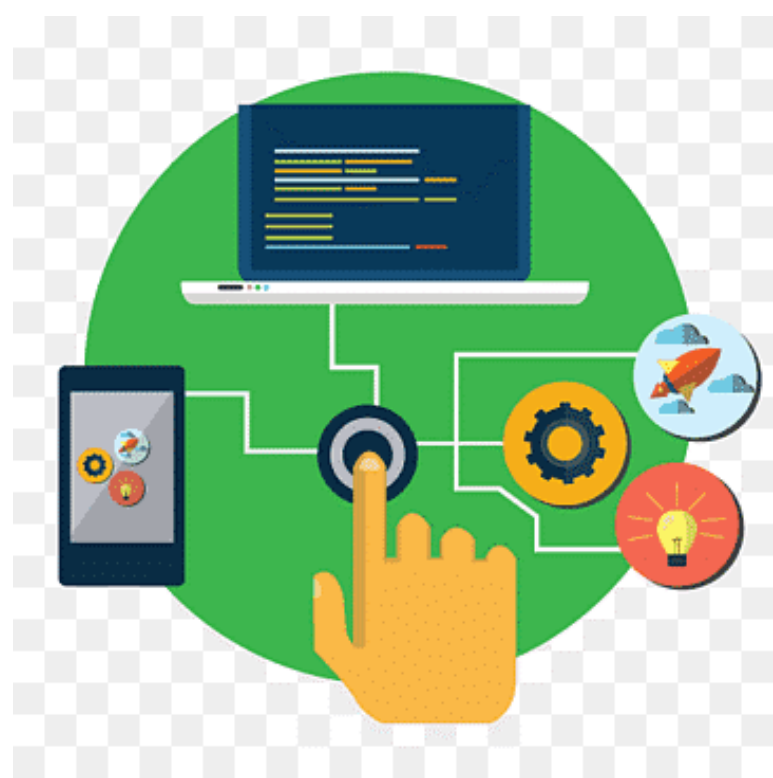
Trình biên dịch và ngôn ngữ lập trình: Cài đặt các trình biên dịch và ngôn ngữ lập trình phù hợp với dự án của bạn. Ví dụ: Java Development Kit (JDK) cho Java, Visual Studio cho C#, Python Interpreter cho Python.

Môi trường phát triển tích hợp (IDE): Sử dụng một IDE phù hợp để lập trình và quản lý mã nguồn. Ví dụ: Eclipse, IntelliJ IDEA, Visual Studio Code.

**Các yếu tố
xem xét**



5. Thiết lập môi trường phát triển (IDE, thư viện, v.v.)



Các yếu tố xem xét

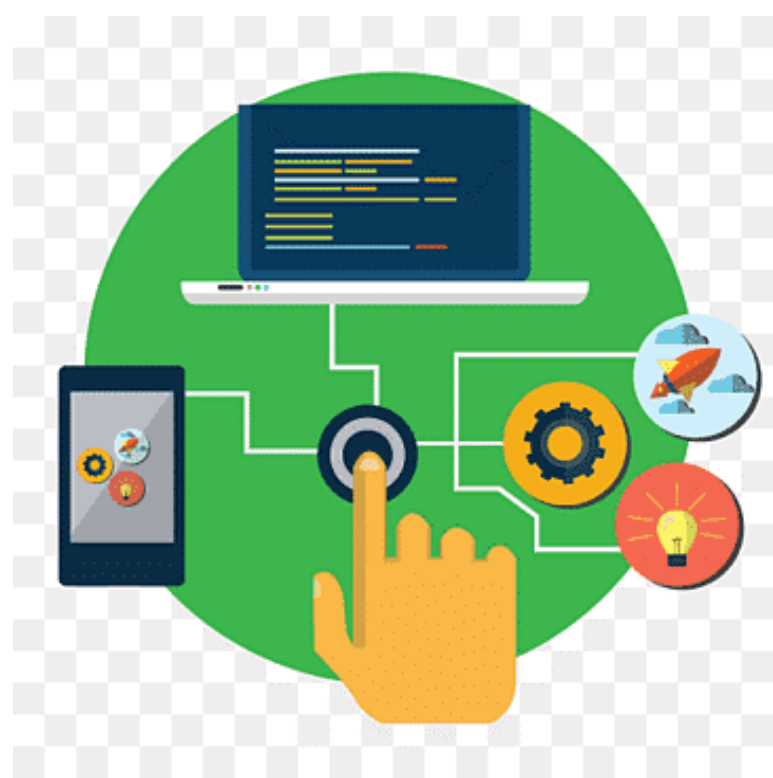
Công cụ quản lý mã nguồn: Sử dụng công cụ quản lý mã nguồn như Git để theo dõi và kiểm soát phiên bản của mã nguồn. Ví dụ: GitHub, GitLab.

Cơ sở dữ liệu: Cài đặt các hệ quản trị cơ sở dữ liệu (RDBMS hoặc NoSQL) phù hợp với dự án của bạn. Ví dụ: MySQL, PostgreSQL, MongoDB.

Công cụ kiểm thử: Sử dụng các công cụ kiểm thử phần mềm để đảm bảo tính ổn định và chất lượng của ứng dụng. Ví dụ: JUnit, Selenium, Postman.



5. Thiết lập môi trường phát triển (IDE, thư viện, v.v.)



Công cụ quản lý dự án: Sử dụng công cụ quản lý dự án để theo dõi tiến độ, quản lý tài nguyên và phối hợp công việc. Ví dụ: Jira, Trello, Asana, Notion,...

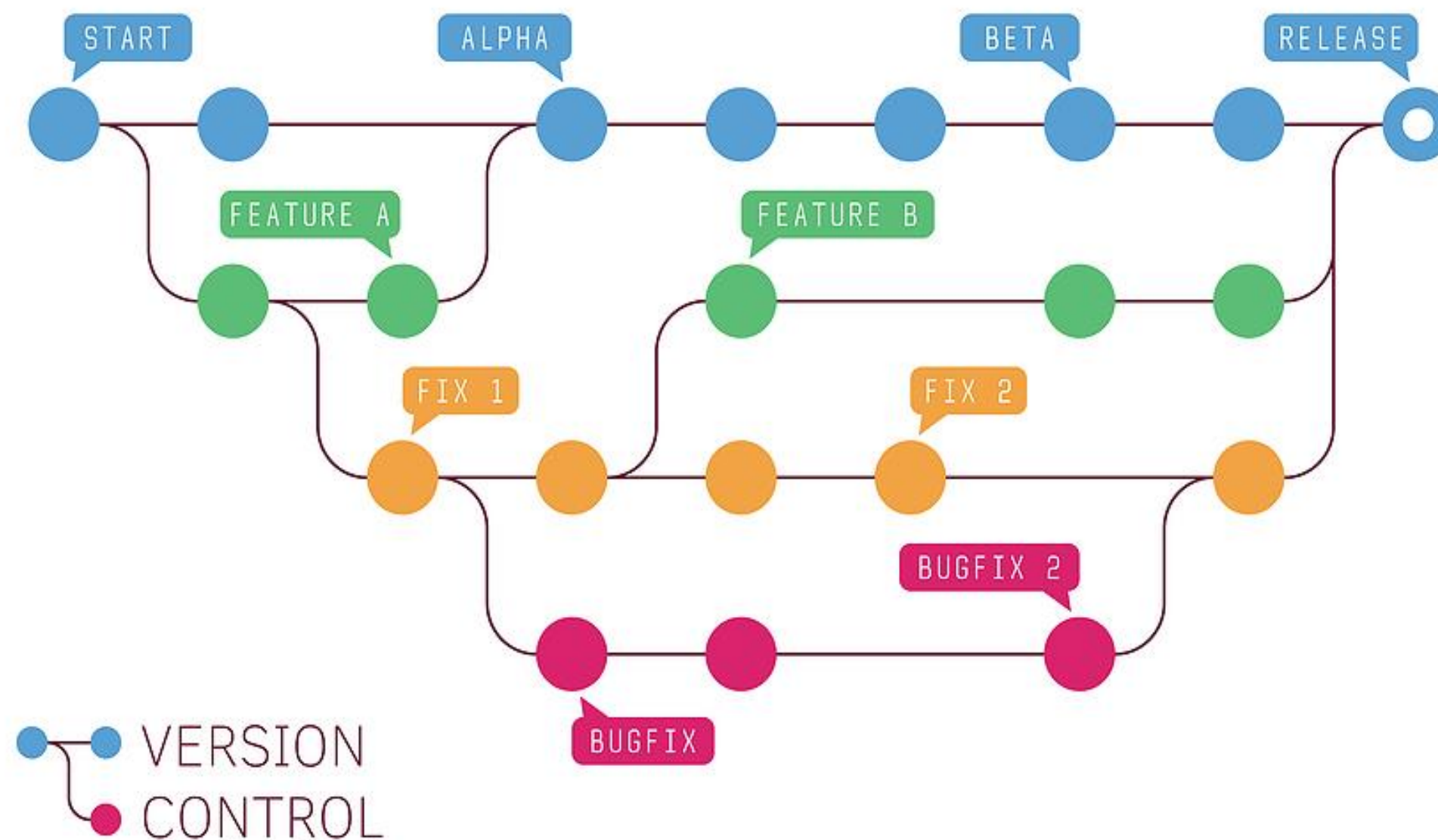
Công cụ hỗ trợ phát triển: Sử dụng các công cụ phát triển phần mềm như trình biên dịch, trình biên tập mã nguồn, trình gỡ lỗi,...

Tài liệu: Tạo và quản lý tài liệu dự án, bao gồm tài liệu yêu cầu, tài liệu thiết kế, tài liệu hướng dẫn,...

Các yếu tố xem xét



6. Giới thiệu về hệ thống kiểm soát phiên bản (Git)



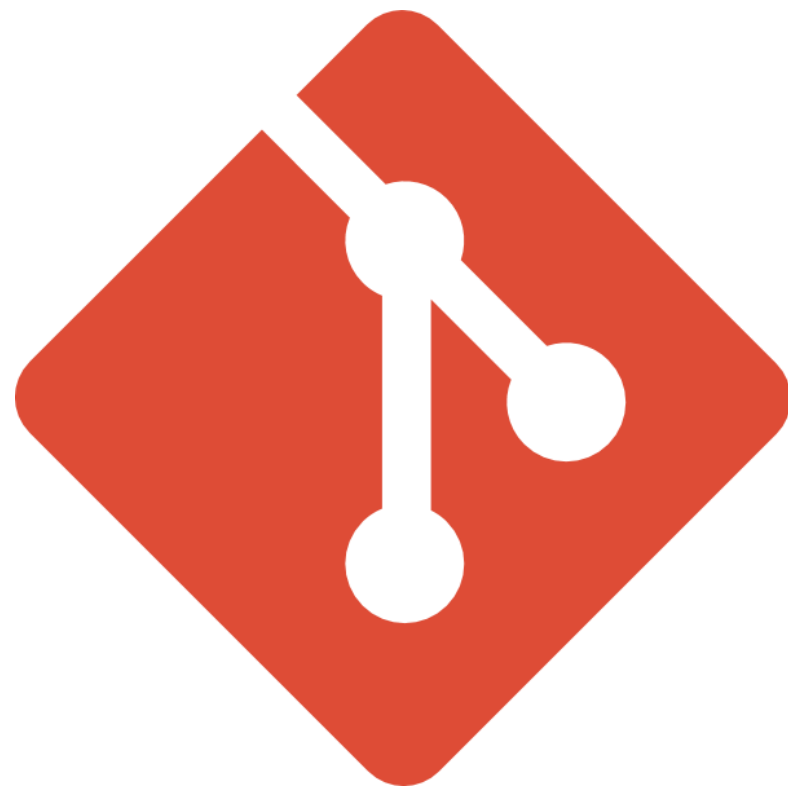
6. Giới thiệu về hệ thống kiểm soát phiên bản (Git)

- Được phát triển bởi **Linus Torvalds** năm 2005;



Git hoạt động bằng cách lưu trữ dữ liệu trong các "**repository**" (kho chứa) cục bộ trên máy tính của mỗi thành viên trong nhóm phát triển. Mỗi repository có thể chứa toàn bộ lịch sử thay đổi của mã nguồn, các nhánh (**branches**) khác nhau và các phiên bản đã được lưu trữ. Mỗi thành viên trong nhóm có thể làm việc trên repository cục bộ và thực hiện các thay đổi mà không ảnh hưởng đến các thành viên khác. Khi đã hoàn thành và kiểm tra, các thay đổi có thể được đưa vào repository chính (trên máy chủ) và được chia sẻ với toàn bộ nhóm.

6. Giới thiệu về hệ thống kiểm soát phiên bản (Git)



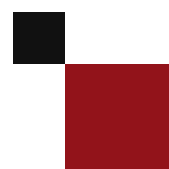
Các công cụ quản lý phiên bản trực quan như GitHub và GitLab cung cấp giao diện người dùng thân thiện và tính năng mở rộng cho việc quản lý repository Git. Chúng cung cấp tính năng như theo dõi và gán công việc, kiểm tra mã, kiểm soát quyền truy cập và hợp tác trong nhóm.

Git đã trở thành một tiêu chuẩn trong việc quản lý phiên bản phần mềm và hợp tác phát triển. Nó mang lại tính linh hoạt, mạnh mẽ và tiện lợi cho việc theo dõi và quản lý mã nguồn trong các dự án phần mềm.





OPENGIS



THANK YOU

