1. **Storage (Lưu trữ):**

Storage là quá trình hoặc nơi mà dữ liệu được lưu giữ và duy trì. Trong lĩnh vực công nghệ thông tin, có nhiều loại lưu trữ khác nhau, bao gồm:

Lưu trữ đĩa cứng (Hard Disk Drive - HDD): Lưu trữ dữ liệu trên các đĩa cứng có cơ cấu cơ học.

Lưu trữ đĩa rắn (Solid State Drive - SSD): Lưu trữ dữ liệu trên các chip flash không cơ học, nhanh hơn và bền hơn HDD.

Lưu trữ đám mây (Cloud Storage): Lưu trữ dữ liệu trên các máy chủ từ xa và có thể truy cập từ mọi nơi qua internet.

Lưu trữ mã nguồn (Source Code Repository): Lưu trữ mã nguồn của phần mềm để quản lý phiên bản và chia sẻ giữa các nhà phát triển.

1. **3rd Party (Bên thứ ba):**

3rd Party (bên thứ ba) ám chỉ đến các tổ chức, nhà cung cấp, hoặc dịch vụ không phải là phần của tổ chức chính. Trong ngữ cảnh công nghệ thông tin, 3rd Party thường liên quan đến các phần mềm, dịch vụ, công nghệ, hoặc các tài nguyên bên ngoài mà một tổ chức sử dụng để mở rộng hoặc bổ sung cho chức năng của họ. Ví dụ, trong lĩnh vực phát triển phần mềm, có thể sử dụng các thư viện bên thứ ba để giảm thời gian phát triển.

Libraries (Thư viện):

Trong lĩnh vực phát triển phần mềm, một thư viện là một tập hợp các mã đã được viết trước, thường chứa các hàm và phương thức đã được triển khai sẵn. Người lập trình có thể sử dụng thư viện này để thực hiện các chức năng cụ thể mà không cần viết lại mã từ đầu. Thư viện giúp giảm thời gian và công sức phát triển, đồng thời cải thiện tính tái sử dụng mã nguồn.

1. **Behavior (Hành vi):**

Behavior (hành vi) thường được sử dụng để miêu tả cách một hệ thống, ứng dụng, hoặc phần mềm hoạt động và phản ứng trong các tình huống cụ thể. Hành vi của một phần mềm có thể được định nghĩa bởi các quy tắc, luật lệ, hoặc tập hợp các chức năng và chức năng sẽ được thực hiện trong các tình huống khác nhau.

Components (Các thành phần):

Các thành phần (components) là các phần cơ bản của một hệ thống, ứng dụng hoặc phần mềm. Chúng được xây dựng để hoạt động cùng nhau và thực hiện các chức năng cụ thể. Các thành phần có thể bao gồm các đối tượng, hàm, lớp, giao diện, cấu trúc dữ liệu, và nhiều yếu tố khác.

1. **State (Trạng thái):**

Trạng thái (state) trong lập trình đề cập đến tình trạng hiện tại của một hệ thống, ứng dụng hoặc thành phần trong một thời điểm cụ thể. Nó thường xuyên thay đổi khi các sự kiện xảy ra và có thể ảnh hưởng đến hành vi của hệ thống hoặc ứng dụng.

Management (Quản lý):

Management (quản lý) là quá trình quản lý, giám sát và điều hành các hoạt động và tài nguyên của một tổ chức, dự án hoặc hệ thống để đảm bảo hoạt động hiệu quả và đạt được mục tiêu. Trong ngữ cảnh phát triển phần mềm, quản lý dự án là quá trình lập kế hoạch, phân công, theo dõi tiến độ và đảm bảo chất lượng của dự án phát triển.

**5. Quality Assurance (Đảm bảo chất lượng):**

Quality Assurance (QA) là quá trình kiểm tra, đánh giá và đảm bảo rằng sản phẩm hoặc dịch vụ đáp ứng các tiêu chuẩn chất lượng đã đề ra. Trong lĩnh vực phát triển phần mềm, QA bao gồm các hoạt động kiểm thử, đảm bảo tuân thủ quy trình và chuẩn mực phát triển, và xác nhận tính chính xác và hiệu suất của phần mềm.

Version Control (Kiểm soát phiên bản):

Kiểm soát phiên bản (Version Control) là một hệ thống quản lý mã nguồn cho phép theo dõi các thay đổi và phát hành của mã nguồn trong quá trình phát triển phần mềm. Nó cho phép các nhà phát triển làm việc cùng nhau trên mã nguồn mà không gây xung đột và dễ dàng quay lại phiên bản trước đó nếu cần thiết.

**6. Version (Phiên bản):**

Phiên bản (Version) là một bản sao cụ thể của một phần mềm, ứng dụng hoặc hệ thống tại một thời điểm cụ thể. Phiên bản thường được đánh số hoặc đặt tên để phân biệt và quản lý sự phát triển của phần mềm. Khi có một phiên bản mới, nó thường bao gồm các cải tiến, sửa lỗi và/hoặc tính năng mới so với phiên bản trước đó.

Control (Kiểm soát):

Kiểm soát (Control) ám chỉ việc quản lý và điều khiển các hoạt động, quy trình hoặc nguồn lực trong một tổ chức hoặc hệ thống. Kiểm soát đảm bảo rằng các hoạt động được thực hiện đúng cách, tuân thủ quy trình và đạt được các tiêu chuẩn đã đề ra.

**7. Firebase:**

Firebase là một nền tảng phát triển ứng dụng di động và web của Google. Nó cung cấp một loạt các công cụ và dịch vụ giúp các nhà phát triển xây dựng ứng dụng di động, web và phần mềm với hiệu suất cao và tính năng phong phú. Firebase cung cấp lưu trữ dữ liệu theo thời gian thực, xác thực người dùng, phân tích, thông báo đẩy và nhiều tính năng khác giúp tối ưu hóa trải nghiệm người dùng.

**8. Native (Ứng dụng Native):**

Ứng dụng Native là những ứng dụng được xây dựng sử dụng các công nghệ và ngôn ngữ lập trình đặc biệt dành cho một nền tảng cụ thể. Ví dụ, ứng dụng Native cho iPhone sẽ được viết bằng Objective-C hoặc Swift, trong khi ứng dụng Native cho điện thoại thông minh Android sẽ được viết bằng Java hoặc Kotlin. Ứng dụng Native thường có hiệu suất tốt và tích hợp sâu vào hệ điều hành, nhưng việc phát triển và duy trì nhiều phiên bản cho các nền tảng khác nhau có thể tốn kém.

Integration (Tích hợp):

Tích hợp (Integration) đề cập đến quá trình kết hợp hai hoặc nhiều hệ thống, ứng dụng hoặc dịch vụ với nhau để làm việc một cách hài hòa và hiệu quả. Tích hợp cho phép dữ liệu và chức năng được chia sẻ và sử dụng chung giữa các hệ thống khác nhau, giúp tăng cường hiệu suất và tối ưu hóa quá trình làm việc.