**Cơ sở dữ liệu: Những lỗ hổng sâu bên trong**

Dành được nhiều sự quan tâm đặc biệt, vì thế cơ sở dữ liệu sẽ không trở thành yếu tố rủi ro trong công nghệ thông tin. Budgeting DRPs, backups, top of the line DBAs – thế nhưng, người ta vẫn đặt ra một mối đe dọa lớn. Tại sao vậy?

**Trở thành Agile**

Trong một môi trường cạnh tranh khốc liệt, nối đối thủ của bạn cung cấp các sản phẩm phù hợp, nhanh hơn với chất lượng tốt hơn thì cuối cùng bạn sẽ mất thị phần. Đó chính là lý do tại sao các công ti, tập đoàn cần phải trở lên rất linh hoạt. Họ cần phải kiểm soát thông tin tốt hơn, quyết đoán hơn, tăng tốc chuyển giao phần mềm, tối ưu và kiểm soát tốt chất lượng sản phẩm. Phát triển Agile được ra đời nhằm đối phó với những thách thức này, việc các tổ chức nâng cao vị thế nhằm di chuyển nhanh hơn, xử lý linh động với các yêu cầu thay đổi, đồng thời đảm bảo. Sự cần thiệt có khái về sản phẩm Agile (Môi trường sản xuất, các trang web…) và các liên kết phát triển cùng với các hoạt động, đã làm tăng DevOps.

Các chu kỳ phát triển phần mềm phổ biết không thay đổi nhiều, trong khi đó Agile và DevOps nổi lên như là hiện tượng của quá trình phát triển phần mềm, chứ không phải là một cuộc cách mạng. Tuy nhiên, các chu kì phổ biến hiện nay ngày càng trở nên quan trọng để thực hiện và theo dõi nếu bạn đang đi trên con đường cao tốc của thế giới Agile. Hơn nữa, tự động hóa là rất quan trọng để nâng cao hiệu quả tổng thể và giảm thiểu rủi ro vốn có trong thay trong môi trường đầy biến động.

**Những yêu cầu đặc biệt chú ý của cơ sở dữ liệu**

Tự động hóa trên cơ sở lặp và liên tục sẽ đưa việc chiển khai trở lên an toàn hơn và ít mắc lỗi hơn. Sự thành công của thay dổi thường xuyên không thể dựa vào khả năng của một người ghi nhớ được tất cả những gì thực hiện, hoặc khả năng của họ xác định phạm vi đầy đủ về những thay đổi hiện nay có thể ảnh hưởng. Đêìu này quan trọng là loại bỏ càng nhiều công đoạn thủ công càng tốt.

Tuy nhiên, không giống các thành phần khác của phần mềm và mã hay mã biên dịch, một cơ sở dữ liệu không phải một tập các tập tin. Nó không thể chỉ được sao chép và dán từ quá trình phát triển sang để kiểm thử và sản phẩm, bởi vì nó là một bộ chứa các tài nguyên có giá trị nhất – dữ liệu nghiệp vụ của bạn, mà phải được lưu trữ. Trong hầu hết các trường hợp, phát triển cơ sở dữ liệu cũng dược thực hiện theo một cách rất khác so với mã hóa ứng dụng (.Net hoặc Java), như các nhà phát triển và DBA (Database Administration) đang truy cập và thay đổi một nguồn tài nguyên chia sẻ, một trung tâm dữ liệu, chứ không phải là một bản sao trên máy trạm.

Sự khác biệt giữa phát triển cơ sở dữ liệu và mã ứng dụng là là rất lớn, trong thời gian đó, các hố sâu được hình thành.

Trong khi các nhà phát triển đang tiến lên, áp dụng các công cụ tuyệt vời và hiện thực hóa Agile, tự động hóa và liên tục phân phối, thì phát triển cơ sợ dữ liệu được thực hiện bằng cách sử dụng một quá trình ít bị kiểm soát. Mang tính nghệ thuật hơn là kỹ thuât, phát triển cơ sở dữ liệu thường tách biệt với vòng đời của phát triển phần mềm. Cá công cụ, quy trình và hoạt động tốt nhất không được phân phối.

**Nên làm thế nào để việc thay đổi cơ sở dữ liệu đươc giải quyết?**

Chúng ta cần phải dừng nhìn vào cơ sở dữ liệu như một thứ gì đó khác nhau, chỉ vì nó được xây dựng khác nhau, hãy chắc chắn rằng các đội, các nhà phát triển, các DBA và phát triển cơ sở dữ liệu đang theo cùng theo dõi một quá trình. Phần mềm đã được chứng minh thực tiễn tốt nhất nên được theo dõ và áp dụng cho cơ sở dữ liệu, như thể họ là phát triển mã.

Để giảm thiểu những rủi ro liên quan tới cơ sở dữ liệu gây ra trong quá trình phát triển, dưới đây là một số đề xuất giải quyết:

* Agile – Rút ngắn các chu kỳ phát triển phần mềm và thực hiện những thay đổi trong cơ sở dữ liệu như là một phần của phát triển dựa trên nhiệm vụ trên. Điều này sẽ đảm bảo rằng những thay đổi về mã và thay đổi của cơ sở dữ liệu nằm trong tầm kiểm soát, sau đó có thể được triển khai với nhau dứa trên yêu cầu thay đổi.
* Việc cung cấp các môi trước phát triển đã được thực hiện hiệu quả bởi Chef, Puppet và Jenkins. Nhưng về cơ sở dữ liệu thì sao? Làm thế nào chúng ta có thể cung cấp các môi trường làm việc với những cơ sở dữ liệu lớn? Hoặc thậm chí là những cơ sở dữ liệu nhỏ? Cơ sở dữ liệu vẫn còn là một rảo cản với Agile
* Việc mặt lạ hóa nội dung có thể trở thành một thách thức thật sự, khi bạn cố gắng xây dựng một sự môi trường phát triển tốt, kiểm tra, UAT (User Acceptence Tesing) hoặc môi trường trước khi sản xuất. Nhưng những gì xảy ra nếu bạn quan tâm cung cấp cho các nhà phát triển tự do của bạn truy cập vào một bản sao của sản phẩm? Được số CC của bạn ngồi trên một máy tính xách tay ngẫu nhiên? Rò rì? Một khia cạnh khác, sau khi biến dữ liệu hoặc không có dữ liệu ở tất cả sẽ không đại diện cho một môi trường thử nghiệm tốt.
* Quản lý thay đổi cơ sở dữ liệu – Thực hiện các công cụ cơ sở dữ liệu để đảm bảo những thay đổi về cơ sở dữ liều đều được quản lý nhưu một phần của chu kỳ phát triển hệ thống. Ngoài ra, hãy chắc chắn rằng mỗi thay đổi tương quan với một yêu cầu thay đổi này giúp quản lý cơ sở dữ liệu như một phần của sự phát triển hoặc thay đổi những nỗ lực chung.
* Kiểm tra tự động hóa – Xây dựng kiểm tra tự động thích hợp đòi hỏi đầu tư lớn. Công việc này được xem là mục tiêu dài hạn.
* Tự động hóa – Chìa khóa cho việc giảm thiểu rủi ro. Bạn càng tự động hóa, càng chính xác hơn, bạn có thể tái tạo được các thay đổi được giới thiệu trong phát triển phần mềm. Thử nghiệm trong môi trường kiểm thử, lặp lại một cách an toàn khi đi vào sản phẩm. Một quy trình phát triển mạnh với các nguy cơ tối thiểu và hiệu quả tối đa có thể đạt được khi bạn tự động hóa các quy trình và giải quyết thách thức nó hằng ngày.

**Tổng kết**

Cơ sở dữ liệu là một thách thức khi trở thành vấn đề thực tiễn nhất. Đây chính là cách xilo được tạo ra. Một bộ phận phát triển có thể thực hiện theo các quy trình khác nhau khi giao dịch với phát triển mã so với phát triển cơ sở dữ liệu hoặc làm theo hoạt động tốt nhất chỉ cho một số lỗ lực thay đổi nó.

Cách tốt nhất để loại bỏ rủi ro chính là dừng nhìn vào những khác biệt về công nghệ và tìm cách – có thể là một tiến trình hoặc một công cụ - đê làm theo và thực thi tốt nhất và tự đông hóa.