



CHUYỂN ĐỔI SỐ VỚI HAI XU HƯỚNG CÔNG NGHỆ MỚI: LOW-CODE (ÍT MÃ) VÀ NO-CODE (KHÔNG MÃ)

■ Vũ Chí Cường
Trường Đại học Vinh

1. Đặt vấn đề

Trong một thập kỷ gần đây, khoa học - công nghệ đã có những bước phát triển vượt bậc. Từ việc kết nối thông tin, tạo ra các nền tảng kinh doanh trực tuyến cho đến việc áp dụng trí tuệ nhân tạo và học máy trong các lĩnh vực chuyên sâu, mọi thứ đều đã và đang diễn ra với tốc độ chóng mặt. Trong bối cảnh này, chuyển đổi số trở thành từ khóa không chỉ được nói đến nhiều trong các tổ chức, doanh nghiệp mà còn là mục tiêu hướng tới của nhiều quốc gia trên toàn thế giới.

Chuyển đổi số không chỉ đơn giản là việc áp dụng công nghệ vào các hoạt động hàng ngày mà nó còn liên quan đến việc thay đổi tư duy, quy trình làm việc và thậm chí là văn hóa của tổ chức, doanh

nh nghiệp. Như trong định nghĩa về chuyển đổi số của Bộ Thông tin và Truyền thông: “Chuyển đổi số là quá trình thay đổi tổng thể và toàn diện cách sống, cách làm việc và phương thức sản xuất”. Trong quá trình này, một trong những thách thức lớn nhất mà các tổ chức, doanh nghiệp phải đối mặt là việc nghiên cứu phát triển và triển khai các phần mềm ứng dụng một cách nhanh chóng, hiệu quả, đồng thời phải đảm bảo tính linh hoạt và khả năng mở rộng trong tương lai.

Trước đây, việc này đòi hỏi phải có một đội ngũ lập trình viên và một lượng lớn thời gian. Nhưng mọi thứ đã thay đổi

nhanh chóng với sự xuất hiện của hai phương pháp: Low-Code (ít mã) và No-Code (không mã). Cả hai đều hứa hẹn biến quá trình phát triển phần mềm trở nên nhanh chóng hơn, ít tốn kém hơn và dễ dàng hơn cho mọi người, bất kể họ có phải là nhà phát triển chuyên nghiệp hay không. Low-Code và No-Code ra đời xuất phát từ nhu cầu thực tế, đó là: Làm thế nào để giảm bớt sự phụ thuộc vào đội ngũ lập trình viên, trong khi vẫn đảm bảo các ứng dụng phát triển ra không chỉ đáp ứng nhu cầu ngắn hạn mà còn linh hoạt đủ để thích nghi với thay đổi trong tương lai? Điều này càng trở nên quan trọng khi chúng ta nhận ra rằng tốc độ của chuyển đổi số đang diễn ra một cách nhanh chóng nếu tổ chức, doanh nghiệp không thích nghi, họ có thể sẽ bị tụt lại phía sau.

Cùng với sự xuất hiện của Low-Code và No-Code, một số câu hỏi được đặt ra: Liệu chúng có thực sự giải quyết được vấn đề? Hay chúng chỉ là công cụ tạm thời, không thể thích nghi với sự phức tạp của thế giới thực sự? Liệu việc sử dụng chúng có đồng nghĩa với việc đánh đổi chất lượng và an ninh, an toàn?...

Bài viết sau đây sẽ phần nào giải đáp những câu hỏi nêu trên.

2. Căn bản về Low-Code và No-Code

2.1. Khái niệm

Low-Code là một phương pháp phát triển ứng dụng nhanh (RAD - Rapid Application Development) nhằm giảm bớt sự phụ thuộc vào việc viết mã lập trình thủ công thông qua việc sử dụng các công cụ và giao diện trực quan hơn để xây dựng ứng dụng. Mục tiêu của Low-Code là giúp người phát triển phần mềm và người sử dụng không cần có kiến thức lập trình sâu cũng có thể tạo ra các ứng dụng nhanh chóng và hiệu quả.

Một số đặc điểm quan trọng của Low-Code bao gồm:

- Giao diện trực quan: Các công cụ Low-Code cung cấp giao diện kéo và thả để người dùng có thể

dễ dàng kéo và thả các thành phần, khối xây dựng, hoặc logic kinh doanh vào một ứng dụng mà không cần viết mã lập trình.

- Tự động hóa: Low-Code tự động tạo mã lập trình dựa trên các hành động và quyết định của người dùng, giúp giảm thời gian và công sức cần thiết để phát triển ứng dụng.

- Mở rộng được: Mặc dù Low-Code cung cấp sự tự động hóa, người dùng vẫn có thể tùy chỉnh và mở rộng ứng dụng thông qua việc viết mã lập trình tùy chỉnh khi cần.

- Tập trung vào kết quả kinh doanh: Low-Code cho phép người phát triển tập trung vào giải quyết các vấn đề kinh doanh thay vì chi tiết kỹ thuật phức tạp. Điều này giúp tăng tốc quá trình phát triển ứng dụng và cải thiện sự linh hoạt trong phản hồi của dự án.

No-Code cũng là một phương pháp RAD và thường được coi là một phần của phương pháp Low-Code. Trong khi Low-Code có một số tình huống được thực hiện bởi người dùng dưới dạng kịch bản hoặc mã lập trình thủ công, thì No-Code lại hoàn toàn không có mã lập trình thủ công, phụ thuộc 100% vào các công cụ giao diện trực quan.

2.2. Sự khác biệt giữa Low-Code và No-Code

Có rất nhiều sự chồng chéo giữa hai phương pháp này (càng trở nên phức tạp hơn bởi sự phân định thiếu rõ ràng của các nhà cung cấp nền tảng Low-Code và No-Code). Dưới đây sẽ là những điểm khác biệt quan trọng của hai phương pháp:

Đối tượng người dùng

Low-Code hướng đến đối tượng người dùng có kỹ năng chuyên nghiệp trong lĩnh vực công nghệ thông tin, giúp họ tránh việc phải sao chép các đoạn mã lập trình cơ bản để tập trung vào các khía cạnh phức tạp hơn của việc phát triển phần mềm. Bằng cách tự động hóa các nhiệm vụ cơ bản, tiêu chuẩn trong lập trình bởi các công cụ có giao diện trực quan, Low-Code cho phép những đối tượng người dùng này tập trung vào sáng tạo và đổi mới, tạo ra các tính năng phong phú và đáp ứng nhanh chóng các yêu cầu thay đổi. Điều này cũng giúp mở rộng đội ngũ lập trình viên tài năng bằng cách giảm đòi hỏi về kỹ thuật và tăng cường đào tạo lại kỹ năng cho đội ngũ.

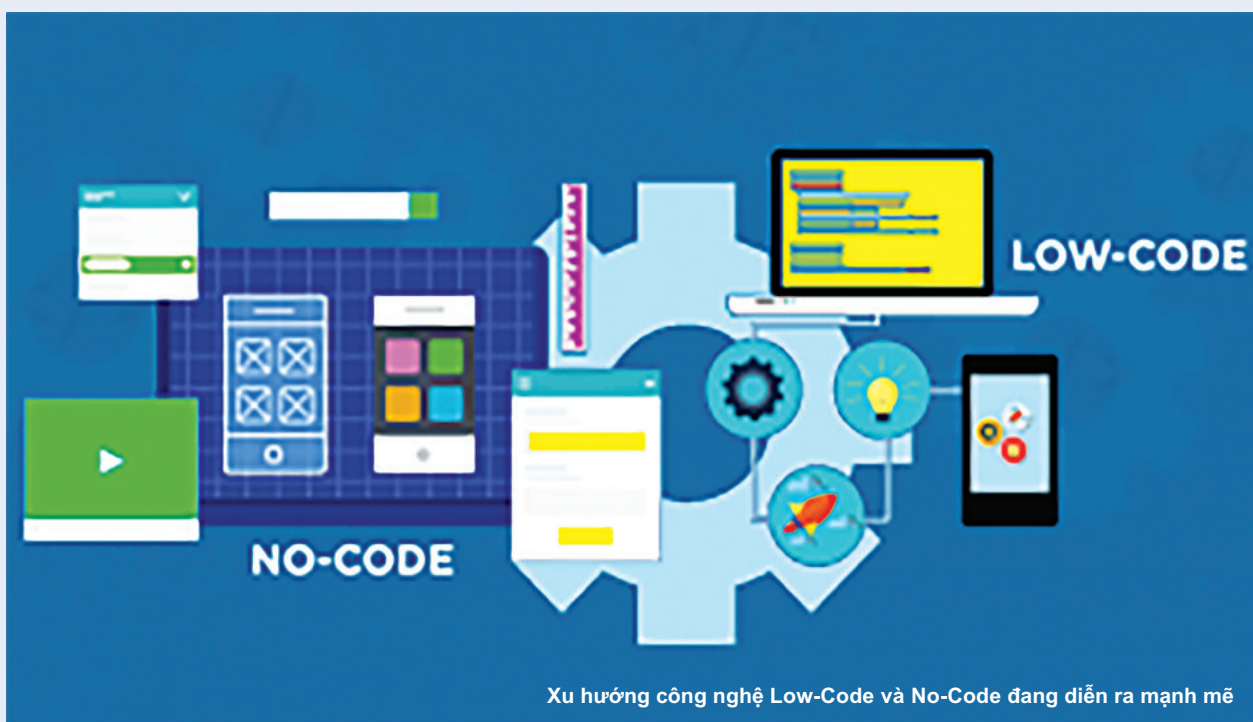
Trong khi đó, No-Code nhắm đến người dùng có kiến thức chuyên môn rộng và có thể cũng có chút kỹ thuật, nhưng thiếu khả năng viết mã lập trình thủ công. No-Code cung cấp một giao diện dễ sử dụng và không yêu cầu kiến thức chuyên sâu về lập trình. Điều này làm cho nó phù hợp cho người dùng kinh doanh, chủ doanh nghiệp nhỏ và các nhóm đối tượng không phải là công nghệ thông tin, như nhân sự, tài chính và luật. No-Code cho phép họ tự mình tạo ra

các ứng dụng và giải pháp phức tạp mà không cần phải dựa vào các nhà phát triển chuyên nghiệp, giúp tăng cường sự linh hoạt và đáp ứng nhanh chóng với các nhu cầu kinh doanh và quản lý.

Các bài toán ứng dụng

No-Code thích hợp cho các phần mềm ứng dụng front-end có thể được thiết kế nhanh chóng bằng giao diện trực quan dạng kéo và thả. Các phần mềm ứng dụng có giao diện người dùng (UI) lấy dữ liệu từ các nguồn, sau đó thực hiện lập báo cáo, phân tích, nhập và xuất dữ liệu là các ứng cử viên tiềm năng.

Ngoài ra, No-Code cũng lý tưởng cho việc thay thế các dự án phần mềm quản trị đơn điệu như lập các báo cáo dựa trên Excel được thực hiện bởi các đội nhóm kinh doanh. Những dự án như vậy thường không được ưu tiên bởi đội ngũ lập trình viên chuyên nghiệp, cán bộ phòng IT nhưng có thể triển khai được bởi các đội nhóm kinh doanh. Nó cũng



phù hợp cho các ứng dụng nội bộ không mang gánh nặng của các tính năng phức tạp và cho các ứng dụng doanh nghiệp quy mô nhỏ với ngân sách phát triển ít.

Low-Code, với thư viện các thành phần đầy đủ, có thể được mở rộng cho các ứng dụng phần mềm có các yêu cầu phức tạp, có quy mô doanh nghiệp. Ngoài ra, để tích hợp với các ứng dụng khác khi sử dụng các dịch vụ API, kết nối với nhiều nguồn dữ liệu và xây dựng các hệ thống với các yêu cầu bảo mật cần sự hỗ trợ của cán bộ phòng IT thì Low-Code là một lựa chọn tốt hơn so với No-Code.

Tốc độ triển khai

Low-Code yêu cầu nhiều kiến thức cũng như thời gian để nghiên cứu, phát triển và triển khai các phần mềm ứng dụng vì nó cung cấp nhiều tùy chỉnh để người dùng cấu hình sản phẩm. Tuy nhiên, nó vẫn nhanh hơn nhiều so với phát triển truyền thống.

Ngược lại, vì có khả năng tự cấu hình cao và tất cả các thành phần giao diện trực quan kéo và thả nên No-Code tốn ít thời gian để xây dựng so với Low-Code. Thời gian kiểm thử cũng giảm bởi vì có rất ít rủi ro về các lỗi tiềm ẩn có thể xảy ra do việc mã lập trình thủ công. Ở đây, tất cả đều liên quan đến việc đảm bảo các cấu hình và luồng dữ liệu được thiết lập đúng cách.

Hệ thống mở và đóng

Low-Code là một hệ thống mở cho phép người dùng có thể mở rộng chức năng thông qua mã lệnh lập trình thủ công. Điều này đồng nghĩa với sự linh hoạt và khả năng tái sử dụng cao hơn. Ví dụ, người dùng có thể tạo các plugin tùy chỉnh và kết nối nguồn dữ liệu để phù hợp với các trường hợp sử dụng và tái sử dụng chúng sau này. Tuy nhiên, cần lưu ý rằng khi nâng cấp và vá mới nền tảng Low-Code người dùng cần kiểm tra, bổ sung những mã lệnh đã được lập trình thủ công.

No-Code là một hệ thống đóng hơn và chỉ có thể được mở rộng thông qua các bộ tính năng mẫu. Điều này đồng nghĩa với việc hạn chế các trường hợp sử

dụng và truy cập vào các plugin được tích hợp sẵn, nhưng dễ dàng đảm bảo tính tương thích ngược với các phiên bản sau của nền tảng No-Code.

Các rủi ro về Shadow IT

Shadow IT (Công nghệ thông tin bên ngoài) là một thách thức quan trọng trong lĩnh vực công nghệ thông tin. Điều này đề cập đến việc nhân viên và các phòng ban trong tổ chức sử dụng các ứng dụng, dịch vụ, hoặc thiết bị công nghệ mà không thông qua bộ phận IT chính thống của công ty. Sự tồn tại của Shadow IT mang theo một số rủi ro, và chúng có thể ảnh hưởng đến cả hai nền tảng Low-Code và No-Code, nhưng mức độ rủi ro có thể khác nhau.

Trong trường hợp sử dụng No-Code, nguy cơ liên quan đến Shadow IT có thể cao hơn, do No-Code gần như không yêu cầu sự can thiệp từ bộ phận IT. Người dùng có thể dễ dàng tạo và triển khai ứng dụng mà không cần kiến thức lập trình, điều này có thể tạo ra một môi trường không được quản lý chặt chẽ, dễ dẫn đến các lỗ hổng bảo mật hay vấn đề nợ kỹ thuật.

Ngược lại, các trường hợp sử dụng phương pháp Low-Code vẫn nằm trong tầm kiểm soát của bộ phận IT, bởi vậy có thể giúp đảm bảo quản trị và kiểm soát tốt hơn.

Phạm vi kiến trúc

Low-Code vượt trội hơn No-Code trong việc hỗ trợ khả năng mở rộng và tương thích đa nền tảng. Việc thêm các plugin tùy chỉnh và mã lệnh thủ công mở ra khả năng triển khai phần mềm ứng dụng trên nhiều nền tảng khác nhau.

No-Code có độ mở rộng ít hơn và tiềm năng giới hạn trong việc kết nối với các hệ thống cũ hoặc tích hợp với các nền tảng khác. Do đó, nó chỉ giải quyết một tập hợp có giới hạn các trường hợp sử dụng và có khả năng mở rộng giảm đi.

3. Lợi ích của Low-Code và No-Code đối với chuyển đổi số

Sự xuất hiện của các phương pháp Low-Code và No-Code đã đưa ra một hướng tiếp cận đột phá trong việc phát triển phần mềm ứng dụng, mang lại nhiều lợi ích thiết thực cho các tổ chức, doanh nghiệp trong quá trình chuyển đổi số.

3.1. Tăng tốc độ phát triển ứng dụng

- Quy trình phát triển nhanh chóng: Các công cụ nền tảng sử dụng Low-Code và No-Code cho phép người dùng tạo ra các phần mềm ứng dụng mà không cần phải viết từng dòng mã lệnh. Điều này cắt giảm đáng kể thời gian phát triển ứng dụng, từ vài tháng hoặc vài tuần xuống chỉ trong vài ngày hoặc vài giờ.

- Thời gian đưa sản phẩm ra thị trường (Time-to-market): Với tốc độ phát triển nhanh, các doanh

nh nghiệp có thể nhanh chóng thích ứng với thị trường, đáp ứng nhanh chóng nhu cầu của khách hàng và đạt được lợi thế cạnh tranh.

3.2. Giảm chi phí phát triển

- Giảm chi phí nhân lực: Việc giảm bớt nhu cầu về đội ngũ lập trình viên, các chuyên gia lập trình cao cấp giúp tiết kiệm đáng kể chi phí nhân sự, đồng thời mở cửa cho việc phân phối nguồn lực tài chính vào các lĩnh vực khác của doanh nghiệp.

- Tái sử dụng và tích hợp: Các nền tảng sử dụng Low-Code và No-Code thường có các mô-đun có thể tái sử dụng và dễ dàng tích hợp với các hệ thống sẵn có, giảm thiểu nhu cầu về việc phát triển từ đầu.

3.3. Dễ dàng thay đổi và bảo trì

- Giao diện trực quan: Giao diện trực quan kéo và thả đã đơn giản hóa việc



Nhiều doanh nghiệp quyết định đầu tư vào Low-Code và No-Code

thêm hoặc chỉnh sửa các chức năng, làm cho việc bảo trì và nâng cấp phần mềm ứng dụng trở nên dễ dàng hơn.

- Phản hồi nhanh với thay đổi: Tổ chức, doanh nghiệp có thể nhanh chóng điều chỉnh ứng dụng để phản hồi trước những thay đổi của thị trường hoặc yêu cầu của người dùng mà không cần phải trải qua một quá trình nghiên cứu, phát triển dài.

3.4. Phân quyền phát triển

- Democratization of Development (dân chủ hoá phát triển): Low-Code và No-Code cho phép người dùng không chuyên từ các bộ phận khác như marketing, quản lý sản phẩm và kinh doanh có thể tham gia trực tiếp vào quá trình nghiên cứu, phát triển và triển khai các phần mềm ứng dụng. Điều này phá vỡ rào cản giữa bộ phận IT chuyên nghiệp và các bộ phận khác.

- Tăng cường sự sáng tạo: Khi những người không phải là đội ngũ lập trình viên có khả năng tạo ra các ứng dụng thì sự sáng tạo có thể được thúc đẩy từ nhiều phía, mở ra cánh cửa cho các ý tưởng đổi mới mà trước đây không thể khám phá.

3.5. Nâng cao trải nghiệm người dùng

- UI/UX thân thiện: Các nền tảng sử dụng phương pháp Low-Code và No-Code thường cung cấp các mẫu thiết kế giao diện người dùng (UI) và trải nghiệm người dùng (UX) chuyên nghiệp, giúp tạo ra các ứng dụng không chỉ chức năng tốt mà còn có vẻ ngoài hấp dẫn và dễ sử dụng.

- Đáp ứng nhanh với phản hồi: Việc thu thập và triển khai phản hồi của người dùng vào ứng dụng trở nên nhanh chóng và dễ dàng hơn, giúp doanh nghiệp liên tục cải thiện sản phẩm của mình.

3.6. Đơn giản hóa quản trị và tự động hóa

- Quản lý dễ dàng: Nền tảng sử dụng

phương pháp Low-Code và No-Code cung cấp dashboard quản lý trực quan giúp theo dõi hiệu suất, lưu lượng sử dụng và các chỉ số khác một cách dễ dàng.

- Tự động hóa quy trình: Việc kết hợp Low-Code và No-Code với công cụ tự động hóa quy trình làm việc giúp tự động hóa các tác vụ lặp đi lặp lại, giải phóng sức lao động cho các công việc có giá trị gia tăng cao hơn.

4. Thách thức và triển vọng

4.1. Thách thức

- Khả năng mở rộng: Mặc dù Low-Code và No-Code rất phù hợp cho việc phát triển các ứng dụng nhanh chóng, nhưng khi tổ chức, doanh nghiệp mở rộng và quy mô của dự án tăng lên, việc điều chỉnh và tùy biến các ứng dụng trên nền tảng này có thể trở nên rất phức tạp. Điều này là đặc biệt quan trọng khi các phần mềm ứng dụng cần phải đáp ứng các yêu cầu phức tạp và mở rộng quy mô.

- Vấn đề bảo mật và an ninh thông tin: Như đã trao đổi ở phần trên, việc sử dụng các nền tảng Low-Code và No-Code đặt ra mối lo ngại về bảo mật dữ liệu. Hiện nay, vấn đề an ninh mạng ngày càng trở nên quan trọng, việc đảm bảo tính bảo mật của thông tin quan trọng và tuân thủ các quy định về quyền riêng tư là một thách thức lớn đối với tổ chức, doanh nghiệp. Bởi vậy, các phần mềm ứng dụng được tạo ra bằng Low-Code và No-Code cần được cấu hình và kiểm tra đúng cách bởi đội ngũ chuyên nghiệp, nhằm làm giảm nguy cơ rò rỉ dữ liệu hoặc tấn công mạng.

- Tùy biến hạn chế và khả năng tương thích: Low-Code và No-Code hỗ trợ phát triển ứng dụng một cách nhanh chóng và dễ dàng nhưng điều này thường đồng nghĩa với việc tùy biến bị hạn chế. Các ứng dụng được xây dựng theo xu hướng này thường tuân theo các mô hình

quy định và có giới hạn trong việc tùy chỉnh mã nguồn. Điều này có thể tạo ra khó khăn đối với các dự án đòi hỏi sự tùy chỉnh cao hơn hoặc cần tương thích, tích hợp với hệ thống hiện có.

4.2. Triển vọng

- Phát triển và hỗ trợ liên tục: Mặc dù có thách thức, những nền tảng Low-Code và No-Code đang trải qua sự phát triển và hỗ trợ liên tục. Những nỗ lực này bao gồm việc cải thiện khả năng hỗ trợ mở rộng, tích hợp và bảo mật. Các công ty phát triển nền tảng này luôn lắng nghe phản hồi của người dùng và cập nhật sản phẩm để giải quyết những thách thức và vấn đề xuất phát từ cộng đồng người dùng.

- Sự tham gia của các công ty lớn: Sự tham gia của các công ty công nghệ lớn như Google, Microsoft và Salesforce trong lĩnh vực Low-Code và No-Code là một tín hiệu tích cực. Điều này cho thấy rằng nền tảng này sẽ đóng một vai trò quan trọng trong tương lai của công nghệ. Các tên tuổi lớn này mang lại sự tin tưởng và cam kết về sự phát triển bền vững cho lĩnh vực này. Một báo cáo của Gartner dự báo rằng vào năm 2024, việc áp dụng công nghệ Low-Code và No-Code sẽ trở nên phổ biến đến mức 75% giải pháp phần mềm toàn cầu sẽ được thực hiện với sự trợ giúp của công cụ này.

5. Kết luận

Nghiên cứu, phát triển phần mềm trong thời điểm hiện nay đòi hỏi sự linh hoạt, tốc độ và sự đơn giản

hóa quy trình. Nền tảng Low-Code và No-Code đã chứng minh là những giải pháp sáng tạo, giúp tháo gỡ những rào cản truyền thống trong việc phát triển ứng dụng, qua đó đem lại cho các tổ chức doanh nghiệp khả năng thích ứng nhanh chóng với thị trường đang thay đổi từng ngày. Tuy còn gặp phải thách thức trong việc mở rộng và bảo mật, nhưng sự hỗ trợ mạnh mẽ từ các nhà cung cấp lớn và cộng đồng người dùng đang dần khắc phục những hạn chế này. Điều này, cùng với dự báo về sự tăng trưởng và tích hợp ngày càng sâu rộng của các nền tảng này vào các quy trình kinh doanh, là minh chứng cho tiềm năng to lớn và vai trò không thể thiếu của Low-Code và No-Code trong tương lai của chuyển đổi số. Chúng ta có thể kỳ vọng rằng, với sự tiến bộ không ngừng của công nghệ và sự đổi mới liên tục, Low-Code và No-Code sẽ tiếp tục mở rộng phạm vi ảnh hưởng của mình, trở thành một phần không thể thiếu trong hành trình chuyển đổi số của mỗi tổ chức, doanh nghiệp./.

Tài liệu tham khảo:

1. Quyết định 749/QĐ-TTg ngày 03/06/2020 phê duyệt Chương trình Chuyển đổi số Quốc gia đến năm 2025, định hướng đến năm 2030.
2. Cẩm nang chuyển đổi số, Bộ Thông tin và Truyền thông, Nxb Thông tin và Truyền thông, 2021.
3. Drive digital transformation with low-code platforms, Microsoft PowerApps (<https://powerapps.microsoft.com/en-us/low-code-trends-accelerate-digital-transformation/>).
4. Digital transformation in 2022 and beyond: These are the key trends, Charles McLellan, Senior Editor ZDNet (<https://www.zdnet.com/article/digital-transformation-in-2022-and-beyond-these-are-the-key-trends/>)
5. Digital Transformation On The Fast-lane With Low-code/no-code Platforms (<https://www.hcltech.com/white-papers/digital-transformation-fast-lane-low-codeno-code-platforms>)