

# THÁCH THỨC TRONG HIỆN TẠI VÀ TƯƠNG LAI CHO CÁC DỊCH VỤ VÀ ỨNG DỤNG CỦA NGÀNH KỸ THUẬT PHẦN MỀM

● NGÔ THỊ LAN ANH

## TÓM TẮT:

Thế kỷ XXI được coi là kỷ nguyên phát triển của công nghệ thông tin, với vô số những sự đột phá và cả sự chuyển hướng với tốc độ phi thường trên quy mô toàn cầu của nền kinh tế. Năng lực xử lý và lưu trữ thông tin của từng cá nhân, từng tổ chức,... ngày một gia tăng nhờ sử dụng máy tính và mạng máy tính - những công cụ phần mềm, ứng dụng hỗ trợ đắc lực cho con người trong các hoạt động lao động trí óc. Tuy nhiên, trong quá trình triển khai và phát triển các dịch vụ, ứng dụng đã và đang gặp phải một số các vấn đề gây ảnh hưởng đến tiến độ và chất lượng của phần mềm. Trong bài viết, tác giả đã nêu nên một số thách thức trong hiện tại và tương lai gần đối với ngành Kỹ thuật phần mềm.

**Từ khóa:** ngành Kỹ thuật phần mềm, chất lượng phần mềm, ứng dụng kỹ thuật.

## 1. Đặt vấn đề

Trong những năm gần đây, ngành Công nghiệp phần mềm Việt Nam đã có những bước nhảy vọt đáng kể. Năm 2020, doanh thu công nghiệp phần mềm và nội dung số đạt trên 6 tỷ USD, cao gấp 2 lần so với doanh thu năm 2015 (3 tỷ USD)[1]. Nhưng mục tiêu lớn hơn và cao hơn là khẳng định vị thế công nghệ phần mềm Việt Nam trên bản đồ IT thế giới. Phần mềm không đòi hỏi máy móc phức tạp để phát triển, nó có thể được tạo ra trên máy tính cá nhân mà ngày nay hầu hết mọi người trong xã hội đều truy cập được. Điều này tạo sự lầm tưởng rằng bất kỳ ai chỉ cần học một số ngôn ngữ lập trình đơn giản đều phát triển được phần mềm. Đồng thời, tính vô hình của phần mềm khiến cho người sử dụng có xu hướng đầu tư và chú ý vào

các thiết bị phần cứng hơn là vào bản thân phần mềm. Đôi khi, một số modul lại được tạo ra bởi chính chuyên gia trong lĩnh vực đó, thay vì kỹ sư phần mềm, đặc biệt đối với các ứng dụng trong khoa học[2].

Nhìn lại, chúng ta có thể thấy rằng, việc thiếu chuyên môn hoặc thiếu đi các khâu trong quy trình phát triển phần mềm đã gây ra các lỗi phần mềm. Điều này sẽ gây đến những hậu quả nghiêm trọng hoặc nếu phần mềm có được sửa chữa thì cũng phải mất một phần chi phí phát sinh[3]. Những năm gần đây, sự phát triển của công nghệ đã dẫn đến việc tự động hóa ngày càng nhiều trong các giai đoạn như kiểm tra, triển khai, quản lý các phần mềm mới, đồng thời cho phép các nhà nghiên cứu và thực hành xác định các

phương pháp tiếp cận mới để tạo và vận hành phần mềm và dịch vụ. Mặt khác, các phần mềm và thiết bị tương tác đôi khi thay đổi vì nhiều lý do, như lỗi thay đổi giao diện và triển khai, thay đổi yêu cầu... Để xây dựng và quản lý những phần mềm một cách hiệu quả, chúng ta cần củng cố các phương pháp quản lý, nâng cao chất lượng của phần mềm và ứng dụng.

## **2. Thực trạng và những thách thức hiện tại trong quá trình triển khai của các dự án phần mềm**

Xu hướng phát triển phần mềm hiện tại là thúc đẩy khách hàng làm trung tâm, mục tiêu là cung cấp cho khách hàng trải nghiệm “liền mạch”, “phù hợp”, “đa kênh”. Trong những năm trở lại đây, ngành Công nghiệp phần mềm phát triển mạnh mẽ trên các lĩnh vực: điện toán đám mây; phân tích và xử lý dữ liệu lớn, phát triển ứng dụng thông minh, trải nghiệm đa giác quan, đa kênh. Cùng theo xu hướng thiết kế và phát triển, các dự án phần mềm đã đang phát triển phương pháp tiếp cận kỹ thuật phần mềm mới, chuyển đổi từ các ứng dụng toàn bộ sang các ứng dụng được phát triển dựa trên việc áp dụng các mô hình microservices và nhu cầu về các phương pháp tiếp cận mới tạo điều kiện thuận lợi cho việc cấu thành dịch vụ/ứng dụng. Việc phát triển theo định hướng trên đã gặp phải những khó khăn trong kỹ thuật triển khai các dự án phần mềm như sau[4]:

### **2.1. Thiết kế phần mềm**

Việc chuyển đổi từ cách tiếp cận phát triển ứng dụng sang cách tiếp cận thành phần ứng dụng được coi là rất quan trọng. Theo hướng này, việc hỗ trợ tái sử dụng phần mềm thông qua phát triển phần mềm dựa trên microservices kết hợp với các khuôn khổ cấu thành phần mềm phải được thực hiện. Trên thực tế, sự chuyển đổi từ các ứng dụng toàn bộ nặng về mã nguồn sang các dịch vụ nhỏ hơn, khép kín đã được thực hiện. Các ứng dụng có kiến trúc microservice bao gồm một tập hợp các dịch vụ có thể triển khai độc lập. Các dịch vụ như vậy có thể được kết hợp để tạo ra đồ thị chuỗi dịch vụ và hỗ trợ các chức năng nâng cao. Các dịch vụ vi mô được triển khai phải được xác thực dựa trên một mô hình đại diện thống nhất liên quan đến các đặc điểm và giao diện kết nối của chúng. Do đó, các dịch vụ như vậy phải được phát triển phần

mềm theo thiết kế, bằng cách áp dụng các mẫu thiết kế có thể tái sử dụng, tách các mối quan tâm và mô hình hóa cấp cao, hỗ trợ việc khảo sát và hiểu các hệ thống phần mềm phức tạp và tái cấu trúc của chúng để phù hợp.

Bên trong các thách thức về thiết kế phần mềm cũng bao gồm các thách thức đối với sự phát triển của các hệ thống tự thích ứng, khả năng đáp ứng, khả năng chịu lỗi, tự phục hồi, các thành phần phần mềm được phát triển phải đáp ứng theo từng thiết kế đối với những thay đổi trong môi trường hoạt động. Trong quá trình thiết kế phần mềm, cần phải tính đến sự thay đổi của các ứng dụng có tính phân tán cao trong môi trường không đồng nhất.

### **2.2. Triển khai phần mềm và điều phối phần mềm trung gian**

Trong các thách thức về triển khai và điều phối phần mềm trung gian là triển khai các ứng dụng trên cơ sở hạ tầng có thể lập trình theo cách tối ưu về hiệu suất phần mềm cũng như các chính sách của nhà cung cấp dịch vụ cấp cao. Do đó, phần mềm được cung cấp phải được mô hình hóa theo cách cung cấp sự linh hoạt cho người dùng (hoặc các công cụ triển khai tự động) để triển khai nó một cách tối ưu. Điều này hầu hết áp dụng cho các ứng dụng có khái niệm phân tán/đồng thời/song song. Hơn nữa, theo hướng này, vị trí dữ liệu, sự biến động dữ liệu và các vấn đề nghiêm trọng về thời gian thúc đẩy định nghĩa về một tập hợp các yêu cầu phải được đáp ứng. Điều quan trọng là có thể quản lý khối lượng công việc sẵn sàng chạy được thiết kế cho các môi trường vật lý, ảo và đám mây dựa trên việc sử dụng các mẫu duy nhất cho tất cả các môi trường đám mây riêng, công cộng hoặc hỗn hợp. Từ những thách thức nói trên trong danh mục này, những thách thức phổ biến nhất là nhu cầu cung cấp các dịch vụ và công cụ dựa trên đám mây để tạo mẫu phần mềm nhanh chóng và nhu cầu quản lý mức độ phức tạp của phần mềm lớn cùng các hệ thống sử dụng nhiều dữ liệu.

### **2.3. Chất lượng của phần mềm**

Đối với các thách thức về chất lượng phần mềm, tập trung vào việc cải thiện chất lượng và khả năng chấp nhận của các dịch vụ dựa trên đám mây thông qua việc kích thích và phân tích các yêu cầu mới, theo hướng cộng tác, các phương

pháp xác thực và xác minh phần mềm, các công cụ và khuôn khổ bao gồm tất cả các tác nhân của chuỗi giá trị, từ các kỹ sư phần mềm và người quản lý sản phẩm cho người dùng cuối. Nỗ lực được phân bổ theo hướng đặc tả các cơ chế nhằm cải thiện độ tin cậy, tính minh bạch và khả năng tương tác của các dịch vụ dựa trên đám mây thông qua việc giới thiệu các phương pháp cho phép (tự/liên kết) xác nhận kết quả đầu ra dựa trên các thủ tục và tiêu chuẩn đã được quy định.

Cũng cần có các phương pháp tiếp cận có thể hướng dẫn sự phát triển chất lượng phần mềm. Các cách tiếp cận như vậy nhằm mục đích thúc đẩy các giai đoạn thiết kế ban đầu của quá trình phát triển ứng dụng và cung cấp các công cụ kỹ thuật chất lượng phù hợp, cũng như tuân theo các thực tiễn tốt nhất về phát triển phần mềm mã nguồn mở, quản trị, cấp phép, kiểm tra, tiêu chuẩn,... Chất lượng phần mềm cũng có thể được cải thiện đáng kể thông qua việc khai thác lượng lớn phản hồi của người dùng. Người dùng cuối ngày càng cung cấp nhiều phản hồi thông qua các đánh giá, xếp hạng và nhận xét trên các diễn đàn trực tuyến, cửa hàng ứng dụng và mạng xã hội.

Mặc dù điều này thể hiện cơ hội để người dùng cuối tham gia vào các quy trình phát triển phần mềm nhằm cải thiện chất lượng của các ứng dụng và dịch vụ phần mềm được cung cấp, nhưng nó đòi hỏi các phương pháp và công cụ để thu thập phản hồi của người dùng cuối cần mở rộng quy

mô và cho phép phân tích phản hồi tự động và dữ liệu ngữ cảnh để hỗ trợ các nhà phân tích yêu cầu, kiến trúc sư hệ thống, nhà phát triển và quản lý dự án trong các nhiệm vụ ra quyết định. Theo hướng này, một thách thức khác là tích hợp dữ liệu hoạt động ẩn danh và tôn trọng quyền riêng tư của khách hàng.

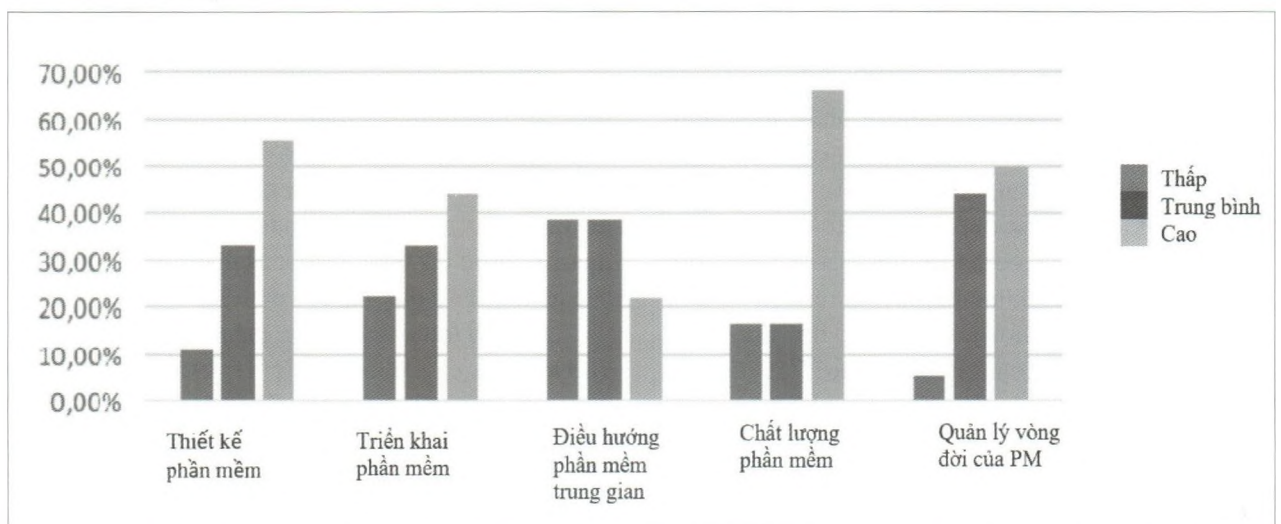
**2.4. Quản lý vòng đời**

Cuối cùng, liên quan đến các thách thức quản lý vòng đời của dịch vụ/ứng dụng, trọng tâm là việc áp dụng các kỹ thuật phát triển dựa trên mô hình và thực hành DevOps.

Quản lý độ phức tạp và rủi ro phát triển trong cả giai đoạn thiết kế và thời gian chạy được coi là rất quan trọng, để tăng chất lượng của phần mềm, giảm thời gian cần thiết để di chuyển các bản phát hành mới trong môi trường hoạt động và cho phép xây dựng các quy trình hợp tác mới và hiệu quả hơn giữa sự phát triển và đội ngũ vận hành. Việc áp dụng các kỹ thuật phát triển dựa trên mô hình bao gồm các giai đoạn khác nhau của vòng đời phát triển phần mềm và kỹ thuật mô phỏng phần mềm cũng được thúc đẩy để tăng năng suất và chất lượng của phần mềm. (Xem Hình)

Từ những thách thức nói trên trong danh mục này, những thách thức phổ biến nhất là nhu cầu cung cấp các dịch vụ và công cụ dựa trên đám mây để tạo mẫu phần mềm nhanh chóng và nhu cầu quản lý mức độ phức tạp của phần mềm lớn và các hệ thống sử dụng nhiều dữ liệu.

**Hình: Mức độ quan trọng của mỗi loại thách thức**



### 3. Những thách thức trong tương lai của ngành Kỹ thuật phần mềm

Những thách thức mới xuất hiện khi tiến bộ công nghệ và các khái niệm mới xuất hiện, trong khi những thách thức hiện tại cũng có những bước ngoặt mới và các hoạt động nghiên cứu và chuyển giao đổi mới tiếp theo là cần thiết để chủ động và giải quyết các vấn đề một cách hiệu quả và hiệu quả. Một trong số các thách thức trong tương lai của ngành Kỹ thuật phần mềm có thể đề cập tới như sau:

#### 3.1. Quy trình, phương pháp và năng suất

Quy trình phần mềm là một lĩnh vực nghiên cứu đã được khảo sát kỹ lưỡng, nhưng ngày nay có một số tiến bộ mới trong công nghệ và thực tiễn tạo ra những thay đổi đáng kể trong khía cạnh này. Phần mềm được đo lường theo khả năng sử dụng, độ tin cậy, khả năng mở rộng. Các khả năng mới để dễ dàng thu thập phản hồi của người dùng và thông tin giám sát cho phép phát triển phần mềm một cách đầy đủ thông tin trong khi các chu kỳ phát triển ngắn hơn đòi hỏi các phương pháp sản xuất phần mềm mới thực sự hiệu quả cho phép quản lý có kiểm soát các chu kỳ phát triển ngắn như vậy. Bên cạnh đó, sử dụng hiệu quả các tài nguyên và hỗ trợ phân tích mức kiến trúc, tối ưu hóa các quyết định triển khai, cũng như sắp xếp và điều phối tự động của các ứng dụng/dịch vụ và áp dụng các phương pháp tiếp cận cơ sở hạ tầng dưới dạng mã để loại bỏ yêu cầu cấu hình và quản lý các cơ sở hạ tầng để có thể lập trình, với trọng tâm là việc sử dụng chúng từ các thành phần trung gian.

#### 3.2. Bối cảnh ứng dụng

Hiện nay các sản phẩm phát triển của phần mềm phát triển nhiều trên các mô hình ứng dụng các hệ thống thực ảo (CPS) và mạng lưới các thiết bị vật lý liên kết qua internet (IoT). Sự phát triển của các dịch vụ, ứng dụng có liên quan nhiều đến khả năng thích ứng với các thay đổi thời gian thực, do đó cho phép các bối cảnh ứng dụng khác nhau. Những thách thức chính do CPS hỗ trợ IoT đưa ra bao gồm việc phát triển các mô hình, phương pháp và công cụ thiết kế cho các ứng dụng hỗ trợ IoT/CPS vượt ra ngoài nghiên cứu các phương pháp chính thức để tạo ra các khái niệm trừu tượng và hình thức để xây dựng và lập luận về các

hệ thống đa dạng các thành phần. Hơn nữa, các nhu cầu do CPS mang lại các phương pháp mới về khả năng thích ứng, Khả năng mở rộng và Khả năng bảo trì của phần mềm đôi khi không được tính đến khi thiết kế trong môi trường CPS mở quy mô lớn.

#### 3.3. Phát triển mô hình thiết kế cho phương pháp tiếp cận hệ thống

Như đã đề xuất ở trên và lưu ý đến bối cảnh ứng dụng khác nhau, cũng cần phải nghiên cứu sâu hơn về việc phát triển các mẫu thiết kế cho phương pháp tiếp cận hệ thống. Các mẫu mới ở cấp độ kiến trúc mô tả các nghĩa vụ/ràng buộc phải được thực hiện bởi hệ thống mà phần mềm đang chạy. Và để xác nhận và chuẩn hóa chúng là cần thiết và các phương pháp áp dụng chúng vào môi trường bối cảnh luôn thay đổi. Do đó, các vấn đề như khung tham chiếu, kiến trúc thiết kế và khả năng tương tác, ngôn ngữ mô hình hóa, tích hợp công cụ và mô phỏng và phân tích,... phải được xử lý.

#### 3.4. Đảm bảo chất lượng

Sự phát triển nhanh chóng trong những năm qua của các phương pháp phân phối nhanh trong bối cảnh của DevOps, cũng như vì nhu cầu giảm thời gian phát triển phần mềm, tuy nhiên phải thực hiện phương pháp tiếp cận nghiên cứu có thể làm tăng chất lượng của hệ thống, giảm thời gian chờ đợi khôi phục dịch vụ và phát triển các phương pháp luận nhanh chóng để kiểm tra chất lượng thông qua quá trình của thử nghiệm phần mềm.

#### 3.5. Yêu cầu kỹ thuật

Một khía cạnh cốt lõi khác có tác động trực tiếp tới bất kỳ hoạt động của phần mềm nào là kỹ thuật yêu cầu. Các thiết bị, dịch vụ mới và thậm chí các cá nhân trở thành một phần của hệ sinh thái hỗ trợ phần mềm. Tính linh hoạt, sự tiến hóa liên tục và kết nối với nhau mâu thuẫn với đầu ra yêu cầu kỹ thuật hiện tại, vì các phương pháp tiếp cận hiện tại đôi khi chưa tính đến tính năng bổ sung trong quá trình hoạt động và các yêu cầu được chưa xác định. Cần có một cách tiếp cận khác biệt hoàn toàn để nắm bắt các hành vi mới từ hệ thống và người dùng. Các công nghệ và xu hướng mới đang làm sáng tỏ các chủ đề nghiên cứu tiềm năng như phân tích dữ liệu lớn đa kênh để thu thập các yêu cầu từ các trang web quy mô

lớn (như cơ sở hạ tầng thành phố thông minh kết hợp con người, máy móc và nói chung là các đặc điểm và hành vi của hệ thống), các phương pháp mới để tương tác với người dùng hướng tới trực tiếp trích xuất các yêu cầu, các yêu cầu gián tiếp tương ứng với các mô hình trích xuất gián tiếp khai thác nhận thức ngữ cảnh của các cá nhân một cách độc lập về việc sử dụng một phần mềm cụ thể.

### **3.6. Quyền riêng tư và bảo mật của thiết kế**

Cần phải quan tâm đặc biệt đến quyền riêng tư và bảo mật trong các hệ thống phân tán phức tạp mà trong nhiều trường hợp phải xử lý khối lượng dữ liệu lớn theo cách phân tán. Các thách thức bao gồm việc xác định các mẫu của hệ thống theo ngữ cảnh liên quan đến các đoạn mã rò rỉ quyền riêng tư, tính toán an toàn cấu trúc dữ liệu, các phương pháp thiết lập mức độ mã hóa tối ưu, đánh giá mã nguồn liên tục tại thời điểm thiết kế cũng như đánh giá lỗ hổng của các ứng dụng đã phát triển, đóng gói an toàn và cơ chế vị trí của các ứng dụng đã phát triển trên cơ sở hạ tầng có thể lập trình, cơ chế điều phối hỗ trợ quản lý chính sách an toàn và hiệu quả của các dịch vụ và ứng dụng, các kỹ thuật đánh giá và xác định rủi ro theo thời gian thực, cùng với việc kích hoạt các hành động giảm thiểu thích hợp. Cần đặc biệt nhấn mạnh đến chủ đề bảo mật và quyền riêng tư bằng cách thiết kế các phương pháp tiếp cận kỹ thuật phần mềm góp phần tạo ra các sản phẩm phần mềm có thể hoạt động trong môi trường đa cơ sở hạ tầng CNTT với các đặc tính bảo mật tăng lên.

### **3.7. Xử lý dữ liệu lớn**

Nghiên cứu những thách thức đối với kỹ thuật phần mềm theo hướng này bao gồm các công cụ mới sử dụng các kỹ thuật máy học và khai thác dữ liệu để tiết lộ các khía cạnh tri thức ẩn và trích xuất thông tin từ các kiến trúc dựa trên cảm biến, khai thác kiến thức mà con người không thể đào ra, nhưng cần được con người chú ý và có tình cảm để cải thiện chất lượng phần mềm. Nghiên cứu sự phát triển/ngừng hoạt động của các khuôn khổ ứng dụng, phân tích xu hướng, sở thích và hành vi của người dùng với các hệ thống để hiểu rõ hơn nhu cầu của người dùng, các công cụ và phương pháp để xác định các cơ hội cải thiện tính

năng và hiệu suất, xác định nguyên nhân gốc rễ của lỗi và hệ thống tạm dừng dựa trên các tệp nhật ký (rất lớn (>> GBs), hoặc cập nhật nhanh đến từ các hệ thống và cơ sở hạ tầng phân tán phức tạp khác nhau, thông tin chi tiết được thu thập trong thời gian chạy về các triệu chứng và thay đổi bối cảnh kích hoạt các hoạt động thích ứng, đồng thời thực hiện các phân tích dự báo và mô tả để chủ động lập kế hoạch và chuẩn bị các hành động thích ứng.

Tóm lại, các thách thức của ngành Kỹ thuật phần mềm bao gồm cả việc xây dựng phương pháp luận và phát triển công cụ hỗ trợ để phát hiện và xử lý các mâu thuẫn và lỗ hổng trong các thông số kỹ thuật yêu cầu, kiến trúc tách biệt. Hơn nữa, các quy trình sản xuất phần mềm cũng bao gồm các thách thức về tổ chức phải được đáp ứng bằng cách tiếp cận liên ngành nhằm tạo ra và quản lý các cộng đồng gồm những người đóng góp mã, người đánh giá, người kiểm tra, người dùng cấp một,... và phương pháp truyền thông và phát triển toàn diện kết hợp các công cụ hiện có dưới một tập hợp các tập hợp phổ biến, được chính thức hóa phương pháp luận.

## **4. Kết luận**

Trong bài viết, tác giả đã cung cấp một cái nhìn tổng quan với những thách thức trong hiện tại và tương lai của ngành Kỹ thuật phần mềm. Các hệ thống phần mềm nằm trong một hệ thống sinh thái phức tạp và siêu kết nối. Xã hội ngày càng phụ thuộc vào phần mềm trong nhiều lĩnh vực, như: giải trí, giáo dục, chính trị, cơ sở hạ tầng công nghiệp và dân dụng, các sáng kiến kinh tế và kinh doanh, cũng như công việc và các hoạt động cá nhân trong nhiều lĩnh vực khác. Tất cả các lĩnh vực này đã trở nên gắn bó chặt chẽ với các hệ thống và ứng dụng phần mềm. Như vậy, phát triển phần mềm là một lĩnh vực đòi hỏi sự hỗ trợ, cải tiến và nghiên cứu các phương pháp mới và công cụ hỗ trợ được áp dụng trong quá trình phát triển phần mềm, cùng với sự phát triển hoàn thiện về kiến thức, kỹ năng của yếu tố con người (người phát triển phần mềm, người sử dụng) sẽ đem đến kỳ vọng giúp cho quá trình phát triển phần mềm dễ dàng, đảm bảo các tiêu chí yêu cầu về chất lượng, bảo mật, tính sử dụng,... của ứng dụng phần mềm ■

**TÀI LIỆU THAM KHẢO:**

1. Đình Nam (2021). Trọng trách tiên phong trong chuyển đổi số. Truy cập tại: [https://sngv.thainguyen.gov.vn/tin-doi-ngoai-viet-nam/asset\\_publisher/r9PVC4I8cvh7/content/trong-trach-tien-phong-trong-chuyen-oi-so/20181](https://sngv.thainguyen.gov.vn/tin-doi-ngoai-viet-nam/asset_publisher/r9PVC4I8cvh7/content/trong-trach-tien-phong-trong-chuyen-oi-so/20181).
2. Zeeya Merali. (2010). Computational science: ... Error. Retrieved from: <https://www.nature.com/news/2010/101013/pdf/467775a.pdf>.
3. Alexandra Witze. (2016). Software error doomed Japanese Hitomi spacecraft. Retrieved from: <https://www.nature.com/articles/nature.2016.19835.pdf>.
4. Giuliano Casale, Cristina Chesta, Peter Deussen, Elisabetta Di Nitto, Panagiotis Gouvas, Sotiris Koussouris, Vlado Stankovski, Andreas Symeonidis, Vlassis Vlassiou, Anastasios Zafeiropoulos, Zhiming Zhao. (2016). Current and Future Challenges of Software Engineering for Services and Applications. Retrieved from: <https://sites.google.com/site/kienthuccongnghphanmem/home/kho-khan-va-khung-hoang-trong-san-xuat-phan-mem>

**Ngày nhận bài: 7/3/2022**

**Ngày phản biện đánh giá và sửa chữa: 3/4/2022**

**Ngày chấp nhận đăng bài: 13/4/2022**

*Thông tin tác giả:*

**NGÔ THỊ LAN ANH**

**Trường Đại học Sư phạm Kỹ thuật Hưng Yên**

**CURRENT AND FUTURE CHALLENGES  
FOR SERVICES AND APPLICATIONS OF THE SOFTWARE  
ENGINEERING INDUSTRY**

● **NGO THI LAN ANH**

Hung Yen University of Technology and Education

**ABSTRACT:**

The 21st century is considered the era of information technology development with countless breakthroughs and rapid changes in the global economy. The capacity to process and store information of each individual, organization, etc. is increasing day by day thanks to the use of computers, computer networks, and applications which effectively support people in various fields. However, in the process of deploying and developing services, applications have faced problems affecting their quality and development process. This paper points out some current and near future challenges for the software engineering industry.

**Keywords:** software engineering, software quality, applications.