

**ĐẠI HỌC QUỐC GIA TP. HCM**  
**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**



**BÀI TẬP VỀ NHÀ BUỔI 2**  
**QUY TRÌNH PHẦN MỀM**

**Giáo viên hướng dẫn: ThS. Võ Tấn Khoa**

**Nhóm thực hiện:**

- 1. Nguyễn Nhật Trường - 20522087**
- 2. Lại Chí Thiện - 20520309**
- 3. Lê Thị Phương Vy - 20520355**
- 4. Lê Trương Ngọc Hải - 20520481**
- 5. Trần Văn Lực - 20521587**
- 6. An Trương Tường Vi - 20520856**
- 7. Trần Thị Anh Thư - 20520792**
- 8. Trương Thị Mai Trinh - 20520825**
- 9. Đỗ Thị Ngọc Bích - 19521263**

**TP. HỒ CHÍ MINH, 2022**

**PHÂN TASK CHO TỪNG THÀNH VIÊN**

<b>STT</b>	<b>MSSV</b>	<b>HỌ TÊN</b>	<b>TASK</b>
1	20522087	Nguyễn Nhật Trường	Phân task, quản lí tiến độ và hỗ trợ nhóm, làm câu 5
2	20520481	Lê Trương Ngọc Hải	Làm câu 8
3	20520355	Lê Thị Phương Vy	Làm câu 6
4	20520309	Lại Chí Thiện	Làm câu 7,10
5	20521587	Trần Văn Lực	Làm câu 1
6	20520856	An Trương Tường Vi	Làm câu 2
7	20520792	Trần Thị Anh Thư	Làm câu 3
8	20520825	Trương Thị Mai Trinh	Làm câu 4
9	19521263	Đỗ Thị Ngọc Bích	Làm câu 9

## BÀI TẬP VỀ NHÀ

1. **(Lực) - Câu 1:** Đề xuất mô hình quy trình phần mềm chung thích hợp nhất có thể được sử dụng làm cơ sở để quản lý sự phát triển của các hệ thống sau. Giải thích câu trả lời của bạn theo loại hệ thống đang được phát triển:

- **Hệ thống kiểm soát phanh chống bó cứng trên ô tô.**

Mô hình quy trình phần mềm thích hợp nhất cho hệ thống kiểm soát phanh chống bó cứng trên ô tô là mô hình thác nước (water fall) vì:

Hệ thống kiểm soát phanh chống bó cứng là một hệ thống quan trọng về an toàn mà mô hình thác nước có nhiều phân tích và tài liệu trước khi bắt đầu triển khai nên phù hợp với hệ thống này.

Để hệ thống kiểm soát phanh chống bó cứng hoạt động tốt thì các yêu cầu phân tích thiết kế phải được lên kế hoạch không có bất kì sai sót nào khi thực hiện hệ thống nên cần phương pháp tiếp cận theo hướng lập kế hoạch với các yêu cầu được phân tích, đặc tả và thiết kế phải đầy đủ, cẩn thận nên mô hình thác nước (water fall) là mô hình quy trình phần mềm thích hợp nhất cho hệ thống này.

- **Hệ thống thực tế ảo hỗ trợ bảo trì phần mềm.**

Mô hình quy trình phần mềm thích hợp nhất cho hệ thống kiểm soát phanh chống bó cứng trên ô tô là mô hình phát triển gia tăng (Incremental development) vì:

Với hệ thống thực tế ảo hỗ trợ bảo trì phần mềm có các yêu cầu hệ thống sẽ có nhiều thay đổi khác nhau, bằng cách chia nhỏ dự án thành nhiều phần nhỏ hơn để phát triển và trình bày với khách hàng để nhận phản hồi, điều này cho phép xây dựng lại hoặc bỏ một số tính năng nhất định.

Đây không phải là một hệ thống quan trọng nên không cần một khuôn khổ hay kiến trúc ổn định và các khả năng đáp ứng của các nhóm khác nhau làm việc trên các bộ phận của hệ thống không cần phải được xác định rõ ràng đối với kiến trúc đó nên hệ thống này có thể sử dụng mô hình phát triển gia tăng để triển khai từng bước.

- **Hệ thống kế toán của trường đại học thay thế hệ thống hiện có.**

Mô hình quy trình phần mềm thích hợp nhất cho hệ thống kế toán của trường đại học thay thế hệ thống hiện có là mô hình tích hợp (Integration) và cấu hình (configuration) vì:

Hệ thống đã hiện có nên các yêu cầu của hệ thống có thể dự đoán trước để lên kế hoạch trước giúp triển khai hệ thống nhanh hơn.

Bằng cách sử dụng mô hình tích hợp và cấu hình chúng ta có thể lấy các thành phần có thể tái sử dụng từ hệ thống cũ và tích hợp chúng vào hệ thống mới nhưng với các tính năng tốt hơn hoặc được cải thiện

- **Hệ thống lập kế hoạch du lịch tương tác giúp người dùng lập kế hoạch hành trình với tác động môi trường thấp nhất.**

Mô hình quy trình phần mềm thích hợp nhất cho hệ thống lập kế hoạch du lịch tương tác là mô hình phát triển gia tăng (Incremental development) vì:

Hệ thống này phụ thuộc vào tương tác và tác động của người dùng nên các yêu cầu hệ thống sẽ thay đổi khi có được trải nghiệm người dùng.

Hệ thống cần có nhiều thử nghiệm được thực hiện trong suốt quá trình, với mô hình phát triển gia tăng có thể giao và triển khai sớm cho khách hàng ngay cả khi các chức năng chưa được bao gồm nhằm thử nghiệm trong quá trình phát triển và nhận phản hồi của khách hàng về hệ thống từ đó đảm bảo cho sản phẩm tốt nhất trong quá trình làm.

2. **(Vi) - Câu 2: Việc phát triển phần mềm gia tăng (incremental development) có thể được sử dụng rất hiệu quả cho những khách hàng không có ý tưởng rõ ràng về các hệ thống cần thiết cho hoạt động của họ. Hãy bàn luận và trả lời nhận định này.**

Phát triển phần mềm gia tăng là quy trình phần mềm nhấn mạnh vào ưu điểm của việc thực hiện từng bước nhỏ để đạt được mục tiêu. Phần mềm được phát triển theo mô hình gia tăng, tức là các tính năng được thêm mới hoặc sửa đổi sau mỗi lần thử nghiệm cho đến khi sản phẩm hoàn chỉnh được khách hàng hài lòng.

Ở mỗi bước gia tăng, các giai đoạn chính của quy trình phần mềm như đặc tả, phát triển và xác thực được thực hiện nhiều lần. Cách tiếp cận này cho phép các nhà phát triển dễ dàng

cung cấp bản demo của phần mềm cho khách hàng để có thể đánh giá mức độ hài lòng của khách hàng về phần mềm. Đồng thời nắm rõ và kịp thời đáp ứng các yêu cầu mới của khách hàng trong quá trình phát triển phần mềm.

Sau mỗi giai đoạn phát triển gia tăng, khách hàng hoàn toàn có thể sử dụng và nhận được giá trị từ phần mềm. Qua đó, họ có thể xem xét và đưa ra đề xuất cho những thay đổi tiếp theo để phần mềm được hoàn chỉnh hơn.

Như vậy, phát triển phần mềm gia tăng có thể đối phó với những thay đổi không thể tránh khỏi do khách hàng không có ý tưởng rõ ràng về các hệ thống cần thiết khi bắt đầu dự án.

3. **(Thu)** - Câu 3: Xem xét mô hình quy trình tích hợp và cấu hình được thể hiện trong hình 2.3 (hoặc slide 16). Giải thích tại sao hoạt động kỹ thuật yêu cầu (requirements engineering activity) được lặp lại là điều cần thiết.

Mô hình tích hợp và cấu hình sử dụng các thành phần hiện có để phát triển một hệ thống mới. Các thành phần được sử dụng lại có thể được định cấu hình để điều chỉnh hành vi và chức năng của chúng theo yêu cầu của người dùng.

Việc lặp lại hoạt động kỹ thuật yêu cầu là điều cần thiết, vì:

- Các yêu cầu cần được điều chỉnh để phù hợp với các yêu cầu của hệ thống mới.
- Các thành phần quan trọng đối với hệ thống mới được xác định.
- Các thành phần có thể tái sử dụng được xác định trong quá trình phân tích yêu cầu và các sửa đổi được ghi nhận.
- Nếu không thể sửa đổi, các phương pháp thay thế được xem xét.

4. **(Trinh)** - Câu 4: Gợi ý lý do tại sao điều quan trọng trong việc phải phân biệt giữa yêu cầu phát triển người dùng (developing user requirements) và yêu cầu phát triển hệ thống (developing system requirements) trong kỹ thuật yêu cầu.

Đầu tiên, quy trình kỹ thuật yêu cầu gồm:

- Phân tích và khơi gợi (elicitation) yêu cầu
- Đặc tả yêu cầu

- Xác thực yêu cầu

Trong đặc tả yêu cầu có chứa yêu cầu của người dùng và yêu cầu hệ thống. Sự khác biệt giữa việc phát triển các yêu cầu của người dùng và các yêu cầu hệ thống trong quy trình kỹ thuật yêu cầu như sau:

- Các yêu cầu của người dùng nhằm mô tả các chức năng và tính năng của hệ thống từ góc độ người dùng và điều cần thiết là người dùng phải hiểu các yêu cầu này. Chúng phải được diễn đạt bằng ngôn ngữ tự nhiên và có thể không được diễn đạt quá chi tiết, để cho phép một số linh hoạt khi triển khai. Những người tham gia vào quá trình phải có khả năng hiểu môi trường và miền ứng dụng của người dùng.

- Các yêu cầu phát triển hệ thống chi tiết hơn nhiều so với các yêu cầu của người dùng và nhằm mục đích trở thành một đặc tả chính xác của hệ thống có thể là một phần của hợp đồng hệ thống (system contract<sup>1</sup>). Chúng cũng có thể được sử dụng trong các tình huống mà việc phát triển được thuê ngoài và nhóm phát triển cần một đặc tả đầy đủ về những gì cần được phát triển. Các yêu cầu hệ thống được phát triển sau khi các yêu cầu của người dùng đã được thiết lập.

Vì thế, lý do cần phân biệt giữa yêu cầu phát triển người dùng và yêu cầu phát triển hệ thống trong kỹ thuật yêu cầu là:

Thứ nhất, người dùng không quan tâm hệ thống hoạt động như thế nào, miễn là nó hoạt động. Người dùng không quan tâm đến những thứ đằng sau hậu trường, miễn là những gì họ tiêu thụ đúng như quảng cáo hoặc được ghi lại... Mọi người chỉ muốn những thứ đó hoạt động. Làm thế nào bạn làm cho nó hoạt động là tùy thuộc vào bạn.

Thứ hai, nếu bạn không phân biệt, bạn có thể bị rò rỉ thông tin quan trọng về cách hoạt động của một hệ thống. Bạn sẽ không xuất bản các phần riêng tư của khóa bảo mật bất đối xứng? Thực hiện các bước bắt buộc như xác định xem thứ gì đó hướng về phía người dùng hay phía hệ thống hỗ trợ duy trì sự khác biệt giữa những thứ nên ở phía sau(bao gồm cả những

---

<sup>1</sup> System contract: <https://www.lawinsider.com/dictionary/system-contract>

thứ riêng tư/ an toàn) và những thứ mà đại diện dịch vụ khách hàng như trạng thái, sự kiện, hỗ trợ...

5. **(Trường)** - Câu 5: Hãy đưa ra một ví dụ để giải thích tại sao các hoạt động thiết kế gồm: thiết kế kiến trúc (architectural), thiết kế cơ sở dữ liệu (database), thiết kế giao diện (interface), thành phần (component) là PHỤ THUỘC lẫn nhau.

### **Các hoạt động thiết kế phụ thuộc lẫn nhau vì:**

Thiết kế của bất kỳ phần mềm nào là hoạt động rất quan trọng. Nó bao gồm thiết kế cơ sở dữ liệu, thiết kế các thành phần khác nhau, thiết kế giao diện người dùng và thiết kế kiến trúc.

Dưới đây là một ví dụ về hệ thống quản lý thư viện và nhận biết được các hoạt động thiết kế này PHỤ THUỘC lẫn nhau như thế nào.

1. **Thiết kế kiến trúc:** Trong giai đoạn thiết kế này, cấu trúc tổng thể của hệ thống được xác định. Nó cũng xác định sự phân bố và mối quan hệ giữa các thành phần nguyên tắc của hệ thống.
  - Kiến trúc thiết kế của hệ thống quản lý thư viện mô tả cấu trúc tổng thể của thư viện. Nó bao gồm những sinh viên có quyền truy cập vào thư viện, bộ phận quản trị thư viện và sách liên quan đến các bộ phận khác nhau, ...
  - Vì vậy, để phát hành một cuốn sách từ thư viện, các thành phần này phải được trao đổi thông tin với nhau.
2. **Thiết kế cơ sở dữ liệu:** Cấu trúc của cơ sở dữ liệu với các thể hiện tương ứng của chúng được xác định trong giai đoạn này. Ở đây, người dùng phải xác định rằng họ sẽ phát triển một cơ sở dữ liệu mới bằng cách sử dụng cơ sở dữ liệu hiện có.
  - Giai đoạn này rất quan trọng trong hệ thống quản lý thư viện vì một lượng lớn sách liên quan đến các phòng ban khác nhau được lưu trữ trong cơ sở dữ liệu.
  - Sinh viên được phép mượn sách bằng thẻ sinh viên của họ và cần phải nộp sách theo quy định của người quản lý thư viện.

- Việc quản lý phải thống nhất khi sách mượn và trả từ cơ sở dữ liệu.
- Nếu dữ liệu được thiết kế tốt thì việc tìm kiếm và duy trì thông tin sách sẽ trở nên dễ dàng hơn.

**3. Thiết kế giao diện:** Giao diện giữa các thành phần khác nhau của hệ thống được xác định ở đây. Có thể một thành phần lại được sử dụng cho bất kỳ quá trình khiêu khai nào trong hệ thống.

- Giai đoạn này xác định các giao diện khả thi khác nhau để sinh viên có quyền truy cập các quyền sách.
- Mỗi sinh viên đều có ID của thư viện hoặc mã số để có thể mượn sách. Với những thông tin này, sinh viên có thể trực tiếp đến thư viện và mượn sách hoặc có thể sử dụng cổng thông tin điện tử của thư viện.
- Bên cạnh đó, sinh viên có thể tìm kiếm các sách cần thiết có sẵn và có thể xem hạn chót để trả sách, ...
- Vì vậy, giao diện là một giai đoạn quan trọng trong khi thiết kế bất kỳ phần mềm nào.

**4. Thiết kế thành phần và việc lựa chọn chúng:** Các thành phần có thể tái sử dụng phù hợp với các yêu cầu của hệ thống mới được xác định ở đây.

- Nếu một sinh viên yêu cầu một cuốn sách không có trong thư viện, quản lý viên nên kiểm tra các cuốn sách phù hợp để đáp ứng yêu cầu của sinh viên.
- Nếu không tìm thấy kết quả phù hợp, ban quản lý phải mua một cuốn sách mới được yêu cầu từ sinh viên.

Vì vậy, để quản lý thư viện hiệu quả và đáp ứng các yêu cầu của sinh viên, quản lý thư viện cần tuân theo tất cả các hoạt động thiết kế đã nêu ở trên.

Do đó, có thể nói rằng tất cả những điều này đều PHỤ THUỘC lẫn nhau.



6. (Vy) - Câu 6: Giải thích tại sao kiểm thử phần mềm luôn là một hoạt động gia tăng, theo giai đoạn. Những người lập trình có phải là những người giỏi nhất để kiểm tra các chương trình mà họ đã phát triển hay không?

Trong việc sản xuất một phần mềm việc kiểm thử luôn là hành động gia tăng, theo giai đoạn vì nếu chỉ kiểm thử phần mềm khi đã hoàn thành sẽ dẫn đến rất nhiều lỗi và việc sửa lỗi sẽ mất rất nhiều thời gian. Kiểm thử được chia thành ba giai đoạn là kiểm thử thành phần (component testing), kiểm thử hệ thống (system testing), kiểm thử khách hàng (customer testing):

- Một phần mềm có nhiều chức năng khác nhau và nhiều đối tượng và hàm chức năng sẽ đảm nhiệm việc thực hiện các chức năng này việc kiểm thử thành phần sẽ giúp kiểm tra các đối tượng và chức năng này đảm bảo chúng thực hiện tốt nhiệm vụ của mình.
- Khi các thành phần đã được kiểm thử chúng sẽ được kết hợp với nhau và kiểm thử hệ thống sẽ đảm bảo các chức năng vẫn hoạt động tốt và không bị xung đột khi kết hợp.
- Nếu hai giai đoạn trước được thực hiện bởi đội ngũ làm phần mềm thì giai đoạn cuối của quá trình, giai đoạn kiểm thử khách hàng sẽ được thực hiện bởi khách hàng hoặc các khách hàng tiềm năng nhằm giúp phần mềm đảm bảo được trải nghiệm tốt nhất khi sử dụng.

Thực tế nhưng người lập trình không phải là những người giỏi nhất để kiểm tra phần mềm. Lập trình viên là những người viết ra phần mềm nên cũng vì thế mà họ khó phát hiện ra lỗi trong chương trình, hơn thế phần mềm được phát triển để phục vụ khách hàng nên phần mềm phải hợp với thị hiếu và dễ dàng cho nhóm đối tượng này sử dụng.

7. (Thiện) - Câu 7: Hãy tưởng tượng một chính phủ muốn có một chương trình phần mềm giúp theo dõi việc sử dụng tài nguyên khoáng sản rộng lớn của đất nước. Mặc dù các yêu cầu mà chính phủ đưa ra không rõ ràng lắm, nhưng một công ty phần mềm đã được giao nhiệm vụ phát triển một nguyên mẫu. Chính

phủ nhận thấy nguyên mẫu rất ấn tượng và yêu cầu nó được mở rộng để trở thành hệ thống thực tế sẽ được sử dụng. Thảo luận về ưu và nhược điểm của cách tiếp cận này.

**Ưu điểm:**

- Giúp cho chính phủ dễ dàng hình dung được kết quả sau khi hoàn thành
- Cách tiếp cận này có thể làm cho chính phủ hài lòng, ngay cả khi các yêu cầu chưa được rõ ràng
- Dựa vào nguyên mẫu có sẵn, thuận tiện hơn trong quá trình lên kế hoạch thực hiện
- Vì đã có nguyên mẫu cơ sở, các yêu cầu nhiều hơn có thể được thực hiện dễ dàng hơn bằng việc thay đổi vài thành phần của nguyên mẫu có sẵn

**Nhược điểm:**

- Phụ thuộc vào nguyên mẫu đã được thiết kế sẵn, làm cản trở tính sáng tạo, linh hoạt trong quá trình phát triển phần mềm
- Tốn chi phí cho việc phát triển nguyên mẫu

8. **(Hải)** - Câu 8: Ban đã phát triển một nguyên mẫu của một hệ thống phần mềm và người quản lý của bạn rất ấn tượng về nó. Cô ta đề xuất rằng nó nên được đưa vào sử dụng như một hệ thống sản phẩm, với các tính năng mới được bổ sung theo yêu cầu. Điều này tránh đi chi phí phát triển hệ thống và làm cho hệ thống hữu dụng ngay lập tức. Viết một báo cáo ngắn cho người quản lý của bạn giải thích lý do tại sao các hệ thống nguyên mẫu thường không được sử dụng làm hệ thống sản phẩm.

Đối với người thiết kế những thiết bị nguyên mẫu hay những thiết bị thương mại giới hạn sử dụng, mức độ an toàn của nó mà ta mong đợi luôn luôn là mơ hồ và không rõ ràng, thậm chí ta còn không biết được rằng đã có những tiêu chuẩn nào được áp dụng. Theo ta thường thấy, một kỹ sư thiết kế sẽ không được đào tạo quá bài bản về an toàn của hệ thống, cũng không có nhận thức rõ được là mức độ an toàn nào là tương đối và cần thiết cho hệ thống của

chính họ. Trong khi đó, với thân là một kỹ sư thiết kế thì để đạt được những mức độ mong đợi là rất khó, hơn nữa có hàng tá những tiêu chuẩn và nguyên lý rất khó hiểu và khổng lồ khiến họ bị “ngộp”, thông thường thì những thứ này sẽ được các cơ quan an toàn tư nhân và thương mại cung cấp. Trong báo cáo đó sẽ có những xem xét về mặt kỹ thuật, tiêu chuẩn và quy định về an toàn của thiết kế sản phẩm dành cho những hệ thống nguyên mẫu hoặc những thiết bị thương mại giới hạn sử dụng. Những thông tin đó hầu hết sẽ dựa trên các yếu tố như tần suất sử dụng, mức độ rủi ro tiềm ẩn, trình độ kỹ năng của người dùng và cả môi trường sử dụng.

Hơn nữa, một cách đơn giản hơn thì ta có thể nói những nguyên mẫu không có tất cả các tính năng theo yêu cầu của hệ thống sản xuất và trong một số trường hợp, chúng chỉ hiển thị giao diện của hệ thống đang được xây dựng (đây là định nghĩa của prototype), do đó việc sử dụng nguyên mẫu như một hệ thống sản phẩm là bất hợp lý vì bản chất nó không dành ra để phục vụ mà chỉ để tham khảo xây dựng và hoàn thiện đến lúc nó thật sự thành một hệ thống sản phẩm hoàn chỉnh.

9. **(Bích) - Câu 9: Đề xuất hai ưu điểm và hai nhược điểm của phương pháp tiếp cận đánh giá và cải tiến quy trình trong Framework SEI.**

- Ưu điểm của phương pháp tiếp cận đánh giá và cải tiến quy trình trong Framework SEI.
  1. Cung cấp phương tiện đo lường trạng thái của quy trình và cách tiếp cận có cấu trúc để đưa ra các cải tiến quy trình.
  2. Hữu ích như một cách xây dựng dựa trên kinh nghiệm của những người khác trong việc cải tiến quy trình.
- Nhược điểm của phương pháp tiếp cận đánh giá và cải tiến quy trình trong Framework SEI.
  1. Giống như bất kỳ hệ thống đo lường nào, có xu hướng đưa ra các cải tiến để cải thiện xếp hạng được đo lường hơn là tập trung vào các cải tiến đáp ứng các mục tiêu kinh doanh thực tế.
  2. Tốn kém để vận hành. Không phù hợp với các cách tiếp cận agile.

10. (Thiện) - Câu 10: Trong lịch sử, sự ra đời của công nghệ đã gây ra những thay đổi sâu sắc trên thị trường lao động (ít nhất là tạm thời) và người lao động. Thảo luận xem việc áp dụng tự động hóa quy trình một cách rộng rãi thì có thể gây ra hậu quả tương tự cho các kỹ sư phần mềm hay không. Nếu bạn cho rằng là không, hãy giải thích? Nếu bạn nghĩ rằng nó làm giảm các cơ hội nghề nghiệp, thì bạn có nghĩ những kỹ sư bị ảnh hưởng đó sẽ có tâm lý bị động hay chủ động chống lại sự ra đời của công nghệ đó?

Trong thực tế lịch sử, mục đích của sự phát triển về khoa học công nghệ không chỉ để cho con người và nhân loại những kiến thức mới và góc nhìn mới mà nó còn mang trong mình mục đích, sứ mạng giúp con người tiết kiệm sức lao động nhưng vẫn mang lại năng suất bằng hoặc thậm chí là cao hơn rất nhiều. Tiết kiệm sức lao động cũng đồng nghĩa với việc những công việc mà trước kia được thực hiện bởi những người lao động có thể sẽ được các công nghệ kỹ thuật tiên tiến thay thế. Nhưng đó là với những tác vụ yêu cầu đòi hỏi sức lao động chân tay như lắp ghép các linh kiện hay những tác vụ nhận thức cơ bản như phát hiện đối tượng, truy vết đối tượng,... Kỹ sư phần mềm là những người hoạt động dựa trên những kỹ năng rất đặc thù và sẽ rất khó khăn để công nghệ có thể tích hợp được những kỹ năng này. Các kỹ năng này có thể bao gồm những kỹ năng về kỹ thuật hoặc những kỹ năng giao tiếp. Giả sử ta có một hệ thống rộng lớn được đồng bộ với nhau để có thể giải quyết vấn đề giao tiếp giữa các đơn vị một cách dễ dàng, tư duy kỹ thuật của một lập trình viên, hay của một bộ não người là một vấn đề khó đối với công nghệ để có thể mô phỏng lại. Tư duy kỹ thuật của một trình viên không chỉ từ những kinh nghiệm trong quá khứ, mà ở đó còn chứa những yếu tố đột biến, sáng tạo không theo kinh nghiệm. Trong khi đó những công nghệ học máy hiện nay chủ yếu dựa vào những dữ liệu đã có, tìm những mẫu số chung hay có thể xem như là kinh nghiệm để đưa ra những quyết định cho những sự kiện chưa gặp. Ở một số mức độ của lập trình, công nghệ học máy hiện tại có thể tự đưa ra những dòng code, thuật toán đơn giản để có thể giải quyết những bài toán thường thấy trong những cuộc thi lập trình, tuy nhiên điều này đang cảnh báo trước một tương lai có thể sẽ khó khăn hơn cho các kỹ sư phần mềm truyền thống. Tóm lại, ở một số lĩnh vực, công nghệ có thể làm giảm cơ hội nghề

nghiệp nhưng nó lại mở ra những cơ hội nghề nghiệp mới, trên thực tế, những kĩ sư phần mềm hiện tại vẫn đang nỗ lực để cải tiến các công nghệ của bản thân mình từng ngày, tốc độ phát triển của khoa học công nghệ đang ở một tốc độ chóng mặt dù cho nó có tiềm ẩn nhiều nguy cơ đối với việc làm, nhưng con người có thể thích ứng rất nhanh với những sự thay đổi, các kĩ sư phần mềm tốt là những người đã trang bị cho mình những kĩ năng cần thiết và quan trọng trong lĩnh vực của mình, sẵn sàng cho những vai trò mới khi việc lập trình ngày càng được tự động hoá và đây cũng là nhân tố chính thúc đẩy cho sự phát triển của công nghệ.

-----**Hết**-----