# [[Học nghề] Thủ công hơn nghệ thuật](https://codegym.vn/blog/2018/08/06/hoc-nghe-thu-cong-hon-nghe-thuat/#respond)



## Thủ công hơn nghệ thuật

*Tôi coi lập trình như một nghề thủ công, một loại nghệ thuật, nhưng không phải là mỹ thuật. Thủ công có nghĩa là làm nên các đồ vật hữu ích, và có thể có một chút tính trang trí. Mỹ thuật thì lại có nghĩa là tạo nên mọi thứ hoàn toàn vì vẻ đẹp của nó.*

-Richard Stallman, *Art and Programming-*

## **Bối cảnh**

Bạn đang được trả tiền để giúp giải quyết vấn đề cho một khách hàng.

## **Vấn đề**

Kể cả khi đã sẵn có một giải pháp được công nhận, vấn đề mà khách hàng của bạn gặp phải chính là cơ hội cho bạn để làm một điều gì đó thực sự tuyệt vời, một thứ đẹp đẽ và sẽ gây ấn tượng với đồng nghiệp của bạn.

## **Giải pháp**

Nghề thủ công được xây dựng dựa trên những mối quan hệ sâu sắc. Tập trung vào việc cung cấp giá trị cho khách hàng của bạn hơn là vì lợi ích cá nhân.

Là một thợ thủ công, bạn chủ yếu xây dựng một điều gì đó phục vụ cho nhu cầu của người khác, chứ không đắm mình trong sự biểu lộ nghệ thuật. Sau tất cả, chẳng có cái gì là một thợ thủ công đói khát cả. Như người bạn của chúng ta, Laurent Bossavit đã nói: “Đối với một thợ thủ công, để bị chết đói là một thất bại; anh ta phải kiếm sống bằng nghề của mình”. Bạn cần phải làm ra thứ tốt nhất của mình theo cách đặt lợi ích của khách hàng lên trên khao khát thể hiện kỹ năng hay là làm đẹp hồ sơ của bạn, trong khi vẫn tuân thủ các tiêu chuẩn tối thiểu về năng lực mà cộng đồng phát triển phần mềm đưa ra. Đi trên con đường dài có nghĩa là bạn phải cân bằng các yêu cầu đầy mâu thuẫn này. Nếu bạn chết đói bởi vì bạn quá nghệ sĩ, và tạo tác của bạn quá đẹp để được chuyển giao tới thế giới thực, thì bạn đừng làm thủ công nữa. Nếu mong muốn được làm những thứ đẹp đẽ buộc bạn bạn phải rời việc phát triển phần mềm chuyên nghiệp và rời việc xây nên những thứ thực sự hữu ích, thì bạn cũng đừng làm thủ công nữa.

Những thứ chúng ta tạo ra *có thể* không đẹp như mong muốn, nhưng *nhất thiết phải* hữu ích. Một phần của quá trình trưởng thành bao gồm mẫu này là phát triển khả năng hy sinh cái đẹp cho tiện ích, nếu và khi điều đó cần thiết.

Đắm đuối vào việc tạo nên các hiện vật đẹp nhưng vô dụng không phải là thái độ của một người thợ thủ công. Họ coi trọng một trò chơi viết bằng năm mươi dòng mã làm cho ai đó cười hơn là một trò chơi hàng triệu dòng mà đã mở rộng biên giới của khoa học máy tính nhưng lại không thể chơi được.

Một khía cạnh khác của thủ công ngược với nghệ thuật, đó là khách hàng cần bạn sản xuất ra sản phẩm với chất lượng đạt yêu cầu ngay cả khi bạn không cảm thấy thích nó. Một thợ thủ công sẽ không chờ đợi cho đến khi cảm hứng ùa đến trước khi người này cung cấp đồ tạo tác để thỏa mãn khách hàng của mình. Điều này mang cả nghĩa tích cực và tiêu cực. Một mặt, thợ thủ công bị loại ra khỏi sân chơi nghệ thuật bình dị, nơi mà người khác trả cho họ để xây dựng những thứ mà họ muốn. Mặt khác, người thợ thủ công và khách hàng của mình lại hài lòng về việc tạo và sử dụng phần mềm cung cấp giá trị ngay lập tức.

### ***Ken nói về nghề thủ công***

*Làm việc với những vấn đề thực tế, cho con người ở đời thực là điều giúp trau dồi tay nghề, chứ không chỉ là làm ra nó để tự thỏa mãn mình.*

-Ken Auer, email-

Mẫu này không đơn giản chỉ là làm những gì thiết thực. Nó còn bao gồm ý tưởng rằng một hiện vật thủ công hữu ích luôn luôn thể hiện ít nhất một mức độ tối thiểu nào đó về chất lượng. Khi sử dụng mô hình này, bạn sẽ phải cân bằng giữa mong muốn của khách hàng để giải quyết ngay lập tức vấn đề của họ, với các tiêu chuẩn nội bộ của một thợ thủ công. Việc có thể giữ được các tiêu chuẩn này ngay cả khi bạn bị áp lực, đòi hỏi bạn phải hiểu rằng tiện ích và vẻ đẹp không phải là đối lập nhau, mà là phụ thuộc lẫn nhau. Phần mềm càng hữu ích, thì phần mềm đó càng quan trọng và đó là một phần mềm có chất lượng càng cao. Nhưng chất lượng đòi hỏi thời gian. Bạn sẽ phải làm việc với một mức chất lượng phù hợp bằng cách liên tục cân bằng giữa vẻ đẹp và tính hữu dụng. Đôi khi bạn sẽ tạo ra một thỏa hiệp sai lầm, và sửa chữa sai lầm đó bằng cách viết lại hệ thống từ đầu có thể không phải là lợi ích tốt nhất cho khách hàng. Trong những tình huống đó, bạn sẽ cần phải phát triển khả năng tái cấu trúc và sửa chữa. Như Sennet đã viết: “Đó là bằng cách sửa chữa những thứ mà chúng ta thường hiểu cách thức làm việc của chúng.” Trong trường hợp này, thời gian dành cho việc khắc phục khi bạn chuyển hướng quá xa thiên về nghệ thuật hoặc sự tiện lợi, sẽ dạy cho bạn những bài học về phát triển phần mềm mà không thể học được bằng bất cứ cách nào khác.

## **Hành động**

Trong 24 giờ tiếp theo, hãy tìm một cơ hội để tạo nên một điều gì đó hữu ích hơn là đẹp đẽ. Nó có thể là một sự lựa chọn đơn giản, hoặc nó có thể liên quan đến một sự thỏa hiệp tinh vi hơn. Điều quan trọng là khiến cho bản thân bạn nhận thức được những vấn đề được thảo luận ở đây khi bạn chọn làm gì đó.

Một cách khác để nâng cao nhận thức của bạn là suy ngẫm về các tình huống trong năm qua khi bạn đã chọn nghệ thuật thay vì tiện ích. Việc đó đã xảy ra như thế nào? Viết ra những gì bạn nghĩ sẽ xảy ra nếu như bạn đã lựa chọn khác đi.