# [[Học nghề] Tiếp xúc](https://codegym.vn/blog/2018/09/09/hoc-nghe-tiep-xuc/#respond)



Tôi thích được tự do một chút để tìm hiểu hoặc sáng tạo, nhưng tôi cũng thích sự hợp tác. Tôi tìm kiếm và phát triển các dự án mà tôi sẽ học hỏi từ những người mà tôi đang làm việc cùng.

-William Kempe-

## **Bối cảnh**

Tuy bạn có thể đã có cố vấn và người cùng chí hướng, nhưng khi phát triển phần mềm, bạn làm việc một mình.

## **Vấn đề**

Năng suất của bạn đã đạt đến một mức đỉnh cao, tuy nhiên sự học tập của bạn đang trì trệ, và bạn có cảm giác rằng có những kỹ thuật và phương pháp ưu việt trong nghề mà bạn chưa nắm bắt được.

## **Giải pháp**

Tìm cách ngồi với một nhà phát triển phần mềm khác và thực hiện một công việc thực hành với nhau. Có một số điều bạn chỉ có thể học khi bạn ngồi với một nhà phát triển phần mềm khác để đạt được một mục tiêu chung.

Mẫu này có thể liên quan chặt chẽ đến mẫu Người Cùng Chí Hướng. Đây là trường hợp của Dave, người đã tìm ra một đồng minh là Roman và sau đó đã có cuộc tranh cãi khi ăn trưa khi họ học chung về các công nghệ như ngôn ngữ lập trình Ruby và phát triển trình cắm thêm Eclipse. Tuy nhiên, ngay cả khi Roman không phải là một người đồng chí hướng, Dave vẫn có thể có được lợi ích khi cùng làm việc với một lập trình viên tài năng, dù là làm trong các dự án cưng. Sẽ có một số kỹ thuật vi mô mà bạn sẽ chỉ học khi hợp tác chặt chẽ với một đồng nghiệp. Đây thường được xem là quá tầm thường để đề cập đến khi giảng dạy, nhưng tác động của chúng cho biết thêm. Việc hợp tác với Roman rất quan trọng đối với sự tiến bộ của Dave với tư cách là một nhà phát triển, vì ông hiếm khi có cơ hội làm việc với các nhà phát triển tài năng vào thời điểm đó trong sự nghiệp của mình.

Thực tiễn phát triển của Lập Trình Theo Cặp là một ví dụ cụ thể của mô hình này, và người học việc nên tìm kiếm cơ hội để làm việc trên các sử dụng kỹ thuật này.Trong khi lập trình theo cặp có thể là một kỹ thuật tuyệt vời để học tập, đó là một hoạt động phức tạp và không phải lúc nào cũng là một trải nghiệm tích cực. Tuy nhiên, khi được sử dụng có hiệu quả, đó là một trong những cách mạnh mẽ nhất để học hỏi, đặc biệt là từ những người cố vấn. Vậy làm thế nào để bạn biết nếu lập trình cặp đang được sử dụng có hiệu quả? Và những gì một học viên có thể làm gì về nó?

Bạn thường sẽ cảm thấy bị mất hút hoặc bị bỏ xa đằng sau đối tác của bạn khi đang thực hiện mẫu này thông qua lập trình theo cặp. Điều này không có nghĩa là lập trình theo cặp không thành công – nó chỉ đơn giản có nghĩa là bạn cần phải làm chậm mọi thứ bằng cách đặt câu hỏi hoặc chịu đựng cảm giác bị mất và cố gắng tiếp thu những mảnh kiến thức mà bạn hiểu. Nhưng nếu bạn luôn cảm thấy mình bị tụt hậu, tuần này qua tuần, và bạn đang bắt đầu tuyệt vọng, thì đã đến lúc phải thay đổi. Bạn có thể bị mắc kẹt với một cặp đối tác lập trình kém, hoặc bạn đời của bạn có thể cần một số gợi ý để cải thiện trải nghiệm của bạn. Là người học việc, bạn có thể không có nhiều quyền để thay đổi tình huống của bạn, nhưng nếu bạn đang ở trong một dự án có nhiều hơn một vài người, bạn có thể tìm thấy cơ hội để luân phiên ngày hoặc tuần giữa các cặp đối tác lập trình. Vòng xoay này có thể giúp bạn thoát ra khỏi tình huống khó khăn và quay trở lại con đường tiến bộ. Ngoài ra, nếu bạn đang thực hành phát triển theo thử nghiệm, bạn có thể đề xuất Lập trình Ping-Pong như một cách để tăng sự tham gia của bạn.

Theo cuốn *The Craftsman* của Richard Sennett, hội thảo nghề thủ công lý tưởng là nơi để “hấp thu vào tri thức ngầm, không nói và không bằng chữ” của “hàng nghìn động tác nhỏ bé hàng ngày mà cộng lại thành một tập luyện” (tr.77). Vì những môi trường lý tưởng như vậy hiện nay rất hiếm, chúng ta phải sử dụng mẫu Tiếp Xúc để thay thế ngày nay. Các ứng dụng của mô hình này không giới hạn chỉ để ghép nối chương trình. Mục đích là để tìm cách để lộ ra thói quen làm việc hàng ngày của những người có tay nghề khác, và quan sát cách thức mà họ dần dần tinh chỉnh những thói quen thành kỹ năng thậm chí nhiều hơn. Những thói quen này vượt xa sự mã hóa và vào tất cả các khía cạnh của phát triển phần mềm.

Ví dụ: bạn có thể hợp tác với một người nào đó trên một bài báo học hoặc một bài thuyết trình hoặc tại một cuộc chạy nước rút của dự án mã nguồn mở. Hoặc, như đã xảy ra với Ade, bạn có thể tự nguyện giúp một ai đó muốn áp dụng lý thuyết đồ thị để thiết kế quản lý phụ thuộc thành phần của một hệ thống quản lý nội dung được viết bằng kịch bản lệnh Unix! Chia sẻ bảng với người muốn sử dụng một công cụ rất thấp để giải quyết vấn đề bạn tự động giải quyết bằng ngôn ngữ bậc cao (hoặc ngược lại) buộc bạn phải tạm thời suy nghĩ giống người khác để giao tiếp hiệu quả. Ngay cả khi bạn từ chối quan điểm đó, bạn đã có một cách nhìn mới về các vấn đề. Quan điểm đó có thể là giải pháp đúng đắn cho một số vấn đề trong tương lai, ngay cả trong ngắn hạn nếu bạn thấy nó đang chao đảo.

Cho dù kinh nghiệm tiếp xúc của bạn là tích cực hay tiêu cực, bạn vẫn nên Ghi Lại Những Gì Đã Học để có thể suy ngẫm về kinh nghiệm của bạn sau này. Một ngày nào đó, bạn có thể sẽ ở cùng vị trí với đối tác lập trình theo cặp của mình, và những trải nghiệm trong quá khứ sẽ cho bạn cái nhìn sâu sắc về suy nghĩ của người trẻ đang ngồi cạnh bạn.

## **Hành động**

Tìm một ai đó mà bạn biết đã bày tỏ quan tâm đến việc bắt đầu hoặc đóng góp cho một dự án mã nguồn mở. Sắp xếp để dành một buổi tối mỗi tuần làm việc cùng nhau trong dự án. Xem bạn có thể làm việc với nhau trong bao lâu. Dòng chảy của một cuộc sống bận rộn chắc chắn sẽ làm ảnh hưởng tiêu cực đến động lực đằng sau sự hợp tác của bạn; khi điều đó xảy ra, bạn phải thích ứng và tìm cách để giữ cho dự án đi cho đến khi bạn tìm lại được động lực. Tất nhiên, nếu động lực không bao giờ trở lại, bạn sẽ tìm ra một mối quan hệ đối tác mới, nơi bạn có thể học những điều mới.