

# § Nguyên tắc YAGNI: Anh Đếch cần gì nhiều ngoài em



Trong ngành phần mềm không hiếm những nguyên tắc ngắn gọn như vậy (như nguyên tắc DRY, KISS chẳng hạn). Nhưng cũng bởi gãy gọn là thế nên người nghe tưởng đã hiểu nhưng lại không chính xác.

YAGNI có nguồn gốc từ thời phương pháp lập trình XP (Extreme Programming) vào những năm 90, trước khi Scrum nổi lên. XP là 1 phương pháp chú trọng tốc độ phát triển trong đó YAGNI được xem nguyên tắc tâm pháp quan trọng. **"Đừng giả định và xây dựng trước các Tính năng phần mềm trước khi chúng ta thực sự cần nó"** là 1 phát biểu tương đối đầy đủ hơn "You Aren't Gonna Need It". Đồng nghĩa với việc tối giản xây dựng các tính năng phần mềm không thực sự cần thiết.

Hãy chú ý, tôi đang focus vào "Tính năng phần mềm".

Thỉnh thoảng, tôi cũng vi phạm YAGNI, với lập luận rằng: **"Ngay thời điểm này, nếu làm luôn tính năng này thì chi phí sẽ ít hơn sau này đến 5 lần"** - nghe như 1 tay bán bảo hiểm ế ẩm đang mời chào cho chính mình nhỉ :)) - thế là tôi giả định trước các Tính năng (tôi cho là sớm muộn gì cũng sẽ cần) và ... viết luôn.

May mắn là có khi tôi cũng đúng, nhưng vẫn có vài tính năng chờ mãi cũng chưa được ai kêu làm. Tuy chưa bị ai than phiền, nhưng nếu "Tính năng giả định" quá nhiều, thì lượng code sẽ phình lên đáng kể, gây trở ngại cho những Tính năng Cần được đáp ứng ngay.

"Anh Đếch cần gì nhiều ... ngoài em" - Câu này chỉ dùng để tỏ tình thôi, còn thực tế ai cũng biết là không phải mà. YAGNI cũng vậy.

Đừng hiểu về YAGNI theo cách cực đoan theo kiểu Đen Vâu nhé.

Một lần tôi được đồng nghiệp chia sẻ về nguyên tắc của cậu ấy, cậu ta "tôn thờ YAGNI", tối giản tất cả vấn đề về source code. Tính năng vừa đủ, kiến trúc tối giản hết mức có thể: Không cần flexible, không dự trù business sẽ phát triển, yêu cầu đến đâu thì kiến trúc đến đó. Hiểu theo cách này thì, YAGNI có đồng minh là KISS và xem SOLID như kẻ thù không đội trời chung rồi à.

Xin chú ý lần nữa, YAGNI đang nói về những "Tính năng phần mềm" mà người dùng cuối nhận được, ngoài ra YAGNI không yêu cầu giảm thiểu những nỗ lực làm phần mềm linh động để sửa đổi hơn. (**Yagni only applies to capabilities built into the software to support a presumptive feature, it does not apply to effort to make the software easier to modify**).

Rõ ràng, quan điểm của người bạn tôi là kiểu hiểu lầm tai hại, phổ biến về YAGNI khi chỉ hiểu đúng theo nghĩa "You Aren't Gonna Need It".

## KẾT

Yagni is not a justification for neglecting the health of your code base. Yagni requires (and enables) malleable code.

YAGNI không phải lời biện minh cho việc bỏ qua sự rõ ràng, linh hoạt của source code. Hoàn toàn ngược lại, YAGNI ủng hộ phương pháp lập trình XP - chú trọng tốc độ, yêu cầu source phải dễ thay đổi, thêm chức năng mới 1 cách dễ dàng với chi phí thấp./