
Chương 4

Các nguyên tắc hướng dẫn thực hành

1. Nguyên tắc hướng dẫn quy trình phần mềm.

1. *Linh hoạt*: cho dù mô hình được chọn để xây dựng phần mềm có linh hoạt hay không thì cách tiếp cận xây dựng phần mềm cần phải có được sự linh hoạt.
2. *Tập trung vào chất lượng tại mỗi bước xây dựng phần mềm.*
3. *Sẵn sàng thay đổi*: quy trình có thể thay đổi. Vì vậy cần điều chỉnh quy trình khi cần thiết để đảm bảo giải quyết các vấn đề phát sinh.
4. *Xây dựng một đội ngũ hiệu quả*: quy trình công nghệ phần mềm và thực hành là rất quan trọng, nhưng yếu tố then chốt vẫn là con người. Vì vậy hãy xây dựng một đội ngũ có năng lực, luôn tin tưởng và tôn trọng lẫn nhau.

1. Nguyên tắc hướng dẫn quy trình phần mềm.

- 5. Thiết lập cơ chế thông tin liên lạc và phối hợp: Dự án thất bại vì các thông tin quan trọng mất mát hoặc các bên liên quan không phối hợp để tạo ra sản phẩm cuối cùng.
- 6. *Quản lý sự thay đổi*: Các thức quản lý có thể chính thống hoặc không, nhưng cơ bản các thay đổi phải được quản lý bằng các biện pháp kiểm tra, đánh giá, phê duyệt.
- 7. *Đánh giá rủi ro*: rất nhiều rủi ro có thể xảy ra trong quá trình phát triển phần mềm. Vì vậy sẽ cần thiết phải xây dựng các kế hoạch dự phòng.

1. Nguyên tắc hướng dẫn quy trình phần mềm.

8. Tạo ra các sản phẩm có giá trị với các công việc khác: chỉ tạo ra những sản phẩm có vai trò trong các quá trình, hành động hoặc công việc khác.

2. Nguyên tắc hướng dẫn thực hành.

1. *Chia để trị*: trong quá trình phân tích và thiết kế luôn quan tâm đến việc phân chia vấn đề thành những vấn đề con dễ giải quyết hơn.
2. *Hiểu tác dụng của các khái niệm trừu tượng*: trừu tượng là sự đơn giản hóa các chi tiết phức tạp trong hệ thống, dùng các thành phần đó để trao đổi.
3. *Tính nhất quán*: một bối cảnh nhất quán làm cho phần mềm dễ sử dụng hơn.
4. *Tập trung vào việc trao đổi thông tin*: Dành sự chú ý đặc biệt đến việc phân tích, thiết kế, xây dựng và kiểm thử các giao diện.

2. Nguyên tắc hướng dẫn thực hành.

5. *Xây dựng phần mềm từ các module*: chia vấn đề cần giải quyết thành các vấn đề nhỏ hơn, mỗi vấn đề nhỏ hơn này sẽ được giải quyết bởi một module.
6. *Tìm kiếm các mẫu*: Brad Appleton cho rằng: "Mục tiêu của mô hình trong cộng đồng phần mềm là giúp các nhà phát triển phần mềm giải quyết những vấn đề tái gặp phải trong quá trình phát triển phần mềm.
7. Khi có thể, đưa ra các vấn đề gặp phải và giải quyết vấn đề đó theo một số cách khác nhau.
8. Hãy nhớ rằng, phần mềm sẽ được duy trì bởi một người nào đó.

3. Nguyên tắc trao đổi

1. *Lắng nghe*: cố gắng tập trung vào người nói thay vì suy nghĩ bạn sẽ phải trả lời thế nào.
2. *Chuẩn bị kỹ cho việc trao đổi*: dành thời gian để hiểu kỹ về vấn đề gặp phải trước khi trao đổi nó với ai đó.
3. *Cần có người dẫn dắt hoạt động*: mỗi cuộc trao đổi cần có người leader dẫn dắt cuộc thảo luận theo một hướng hữu hiệu, giải quyết các vấn đề xảy ra và đảm bảo cho các nguyên tắc khác được đảm bảo.
4. *Cách tốt nhất là trực tiếp đối thoại*: sẽ là tốt nhất nếu các bên liên quan đều có người đại diện để trình bày, phân tích vấn đề gặp phải.

3. Nguyên tắc trao đổi

5. *Ghi chú và viết lại các tài liệu* : một người sẽ tham gia vào cuộc trao đổi với nhiệm vụ ghi lại các điểm và quyết định quan trọng.
6. *Cố gắng hợp tác*: Hợp tác để các kiến thức chung của nhóm được kết hợp với nhau.
7. *Xác định mục đích của cuộc thảo luận một cách rõ ràng*: nếu có nhiều người cùng tham gia vào một cuộc thảo luận thì gần như chắc chắn rằng cuộc thảo luận sẽ đi từ topic này đến topic khác, do đó, trước khi thảo luận cần xác định rõ chủ đề và mục tiêu cần đạt được là gì.
8. Nếu một điều nào đó không rõ ràng, hãy nói điều đó ra.

3. Nguyên tắc trao đổi

9. Việc gì mà bạn đồng ý, không đồng ý hay có điều gì bạn không thể chỉ rõ ra ngay được, hãy cứ bỏ qua.
10. Đàm phán không phải là một trò chơi hay sự cạnh tranh. Đàm phán thành công khi cả hai bên cùng là người chiến thắng.

4. Nguyên tắc lập kế hoạch.

1. *Hiểu được mục tiêu của dự án:* mục tiêu của dự án giúp đội ngũ làm việc biết được đích đến cuối cùng cần phải đạt được.
2. Đưa khách hàng vào trong kế hoạch: bởi khách hàng là người định rõ thứ tự ưu tiên, deadline của dự án.
3. *Chấp nhận rằng việc lên kế hoạch có thể lặp lại:* khi bắt tay vào xây dựng dự án, có rất nhiều thứ sẽ thay đổi không theo kế hoạch, do đó, việc lập lại kế hoạch là đương nhiên sẽ xảy ra.

4. Nguyên tắc lập kế hoạch.

4. *Ước lượng* : mục đích của ước lượng là cung cấp một cái nhìn tổng quan về chi phí, thời gian, công sức cần bỏ ra để hoàn thành dự án dựa trên hiểu biết hiện tại của nhóm nghiên cứu.
5. *Cân nhắc tới rủi ro khi bạn lập kế hoạch*: khi bạn đã xác định được những rủi ro có khả năng xảy ra cao thì việc có một kế hoạch để phòng những rủi ro đó là hết sức cần thiết.
6. *Mức độ chi tiết khi lập kế hoạch*: đề cập đến mức độ chi tiết là một phần của kế hoạch phát triển.

4. Nguyên tắc lập kế hoạch.

7. *Xác định bằng cách nào để đảm bảo chất lượng:* kế hoạch cần xác định nhóm phát triển sẽ đảm bảo chất lượng phần mềm bằng cách nào.
8. *Hãy thực tế:* mọi người không thể làm việc 100% mỗi ngày.
9. *Xác định việc thích ứng với thay đổi như thế nào trong kế hoạch:* bởi ngay cả với kế hoạch tốt nhất cũng bó tay trước những thay đổi bất ngờ.
10. *Theo dõi kế hoạch* thường xuyên và điều chỉnh theo yêu cầu: kế hoạch có thể bị chậm tiến độ vào một ngày nào đó.

5. Nguyên tắc mô hình hóa

- Trong công nghệ phần mềm, 2 lớp mô hình cần được tạo đó là:
 - Mô hình yêu cầu (mô hình phân tích): thể hiện yêu cầu của khách hàng thông qua việc miêu tả chức năng của phần mềm.
 - Mô hình thiết kế: thể hiện các đặc tính của phần mềm giúp cho những người xây dựng phần mềm một cách hiệu quả: kiến trúc phần mềm, giao diện người dùng..

5.1 Nguyên tắc xây dựng mô hình yêu cầu:

1. Miền thông tin cần phải được hiểu rõ.
2. Các chức năng của phần mềm phải được xác định rõ ràng.
3. Hành vi của phần mềm phải được thể hiện.
4. Mô hình miêu tả chức năng, thông tin, hành vi của phần mềm cần được phân chia theo cách thức phát hiện ra, theo lớp hoặc phân cấp.
5. Nhiệm vụ phân tích nên di chuyển từ các thông tin cần thiết tới chi tiết thực hiện.

5.2 Nguyên tắc xây dựng mô hình thiết kế.

1. Mô hình thiết kế phải dựa trên mô hình yêu cầu (mô hình phân tích).
2. Luôn xem xét các cấu trúc của hệ thống được xây dựng.
3. Thiết kế dữ liệu có vai trò quan trọng như thiết kế các chức năng xử lý.
4. Giao diện người dùng cần thiết kế theo yêu cầu của người sử dụng cuối, tuy nhiên trong nhiều trường hợp cần nhấn mạnh tính dễ sử dụng.
5. Ở mức độ các thành phần component, việc thiết kế nên được tiến hành độc lập.

5.2 Nguyên tắc xây dựng mô hình thiết kế.

6. Nên thiết kế các component lỏng lẻo với nhau và với môi trường bên ngoài.
7. Mô hình nên thiết kế một cách dễ hiểu.
8. Việc thiết kế nên được lặp lại nhiều lần, sau mỗi lần lặp mô hình thiết kế cần đơn giản hơn mô hình trước nó.

5.3 Nguyên tắc xây dựng mô hình agile.

1. Mục tiêu cuối cùng của nhóm phát triển là tạo ra phần mềm, không phải tạo ra mô hình.
2. Đường đi tốt không tạo ra nhiều mô hình hơn bạn cần.
3. Cố gắng tạo ra các mô hình đơn giản để thể hiện các vấn đề trong việc xây dựng phần mềm.
4. Xây dựng mô hình sao cho có thể dễ dàng thay đổi chúng.
5. Nêu một mục đích rõ ràng cho mỗi mô hình được tạo ra.
6. Thích ứng với các mô hình tự bạn phát triển cho hệ thống.

5.3 Nguyên tắc xây dựng mô hình agile.

7. Cố gắng xây dựng các mô hình hữu ích, nhưng đi việc việc xây dựng mô hình hoàn hảo.
8. Đừng quá cứng nhắc về cú pháp của mô hình.
9. Nếu cảm thấy mô hình có vấn đề mặc dù nó ổn trên giấy, bạn có lý do để lo ngại.
10. Nhận phản hồi trong thời gian sớm nhất có thể.

6. Nguyên tắc xây dựng.

- Việc xây dựng bao gồm các công việc lập trình, kiểm thử đảm bảo cho phần mềm hoạt động và sẵn sàng cho việc phân phối.
- Nguyên tắc và khái niệm lập trình khá tương đồng với ngôn ngữ lập trình, phong cách lập trình và phương pháp lập trình.
- Nguyên tắc và khái niệm kiểm thử dẫn đến việc thiết kế các bài test với chi phí thấp để tìm ra các lỗi.

7. Nguyên tắc chuẩn bị.

- Trước khi bắt tay vào lập trình, hãy chắc chắn rằng:
 - Hiểu vấn đề mà bạn đang cố gắng giải quyết.
 - Có hiểu biết cơ bản về khái niệm và nguyên tắc thiết kế.
 - Lựa chọn ngôn ngữ lập trình phù hợp với mục đích của phần mềm và môi trường mà nó hoạt động.
 - Lựa chọn môi trường làm việc hỗ trợ được bạn tốt nhất.
 - Tạo ra các bài test nhỏ để bạn có thể kiểm thử sau mỗi lần bạn hoàn thành một bộ phận nào đó.

8. Nguyên tắc lập trình.(1/2)

- Khi bắt đầu lập trình, hãy chắc chắn rằng:
 - Ràng buộc giữa cấu trúc dữ liệu và giải thuật bạn sử dụng.
 - Cân nhắc về việc lập trình theo cặp.
 - Lựa chọn cấu trúc dữ liệu phù hợp với thiết kế.
 - Hiểu kiến trúc của phần mềm và tạo ra các giao diện phù hợp với nó.
 - Giữ các biểu thức điều kiện càng đơn giản càng tốt.
 - Tạo vòng lặp lồng nhau nhưng phải đảm bảo chúng dễ dàng được kiểm thử.

8. Nguyên tắc lập trình.(2/2)

- Khi bắt đầu lập trình, hãy chắc chắn rằng:
 - Đặt tên biến mang ý nghĩa, sau đó mới theo các quy tắc khác.
 - Viết dòng code dễ hiểu, dễ đoán biết được chức năng của khối lệnh.
 - Tạo ra một bố cục trực quan (ví dụ lùi đầu dòng...).

9. Nguyên tắc thông qua.

- Sau khi hoàn thành bước lập trình hãy:
 - Sắp xếp lại code khi thích hợp.
 - Thực hiện các bài test nhỏ để kiểm thử và sửa lỗi vừa được phát hiện.
 - Cấu trúc lại mã.

10. Nguyên tắc kiểm thử.

1. Các bài test nên hướng đến các chức năng mà khách hàng yêu cầu.
2. Lên kế hoạch cho các bài test trước khi bước vào test thực sự.
3. Áp dụng nguyên tắc Pareto để kiểm thử phần mềm.
4. Các bài test thực hiện trong các quá trình nhỏ trước rồi mới đến các quá trình lớn.
5. Không thể kiểm tra toàn diện được.

11. Nguyên tắc triển khai

1. Quản lý kì vọng của khách hàng: thông thường khách hàng kì vọng nhiều hơn những gì đội phát triển có thể làm. Do vậy sự thất vọng sẽ xảy ra ngay lập tức.
2. Bàn giao phần mềm đầy đủ bao gồm việc lắp ráp và kiểm thử.
3. Cung cấp chế độ bảo hành: khi một vấn đề xảy ra phải đảm bảo thỏa mãn được yêu cầu của người dùng cuối.
4. Cung cấp tài liệu sử dụng tới người dùng.
5. Phần mềm có lỗi phải được xử lí trước khi đến tay người dùng.