**BÁO CÁO PHẦN TÌM HIỂU LUỒNG CÔNG VIỆC PHA THỰC THI**

**1: Các công việc cần thực hiện trong luồng công việc**

* Thực hiện quy trình thực hiện.
* Lựa chọn ngôn ngữ lập trình
* nguyên tắc kỹ thuật phần mềm có thể dẫn đến code chất lượng tốt hơn.
* Thực hành lập trình tốt: Sử dụng tên biến có tính nhất quán và có ý nghĩa, chú thích code, sử dụng tham số, bố cục code tăng khả năng đọc
* Chuẩn hoá code:
* Tái sử dụng code:
* Tích hợp: tích hợp từ trên xuống, tích hợp từ dưới lên, tích hợp sandwich, tích hợp các sản phẩm hướng đối tượng
* Nguyên tắc phân chia và code modul
* Một method(modul) chỉ tối đá 30 -50 câu lệnh
* Nếu nhiều hơn thì nên tách thành một modul con
* Thực hiện kiểm thử đơn vị hộp đen, hộp kính(nêu ở phần kỹ thuật)
* Cleanroom: là một cách tiếp cận khác đối với phát triển phần mềm
* Một số vấn đề có khả năng cao xảy ra khi kiểm thử các đối tượng: lớp, đối tượng, chạy từng trường hợp thử nghiệm với các đối tượng, các đối tượng điểm hình, các phương thức kế thừa,…
* Khía cạnh khác cảu kiểm thử đơn vị: phân tích chi phí – lợi nhuận, phân tích rủi ro, các kỹ thuật thống kê
* Thực hiện kiểm thử tích hợp: là việc kiểm thử khi có một modun mã mới được thêm vào nhóm các modul đã được kiểm thử
* Tiến hành kiểm thử sản phẩm :
* Cho phần mềm COTS: alpha, beta
* Đối với phần mềm tuỳ chỉnh: đảm bảo phần mềm chuyển qua kiểm thử chấp nhận
* Kiểm thử chấp nhận: khách hàng xác định liệu phần mềm thoả mãm những đặc tả của nó. 4 thảnh phần chính: tính chính xác, tính mạnh mẽ, hiệu năng và tài liệu
* Đánh giá cao nhu cầu thực hành lập trình tốt và các tiêu chuẩn lập trình.

**2:Các kỹ thuật được áp dụng**

* Các kỹ thuât tính hợp:
* Top – down: Tích hợp từ trên xuống
* Ưu điểm: hỗ trợ cách ly lỗi, giúp phát hiện sớm các lỗi thiết kế lớn
* Nhược điểm: các modul có khả năng tái sử dụng không được kiểm thử một cách thích đáng, các modul thấp hơn(mức hành động) không được kiểm thử thường xuyên
* Bottom – up: Tích hợp từ dưới lên
* Ưu điểm: các modul hoạt động được kiểm thủ kỹ lưỡng, các code có khả năng tái sử dụng có tính đầy đủ(giải quyết được yếu điểm của Top -down)
* Nhược điểm: thiết kế chính sẽ bị phát hiện muộn. các thành phần logic được tạo ra cuối cùng, do đó nếu có nếu có lỗi thiết kế lớn, sẽ được xử lý ở cuối => tốn kém chi phí và thời gian
* Sandwich :các modul logic được tích hợp từ trên xuống, các modul hoạt động tích hợp từ dưới lên, tích hợp theo cả 2 chiều trên xuống và dưới lên: có được điểm mạnh của cả 2 kỹ thuật trên
* Kỹ thuật kiểm thử hộp đen:
* Kiểm thử tương đương và phân tích các giá trị biên
* Kiểm thử chức năng
* Kỹ thuật kiểm thử hộp kính: kiểm thử cấu trúc
* Phủ dòng lệnh
* Phủ nhánh
* Phủ đường dẫn
* Chuỗi mã tuyến tính
* Phủ đường dẫn sử dụng tất cả định nghĩa

**3: Các kết quả (artifact) cần đạt**

* Lựa chọn được ngôn ngữ lập trình
* Nắm được các chuẩn lập trình cơ bản
* Các kỹ thuật tích hợp: trên xuống, dưới lên ,sandwich
* Nắm đươc kỹ thuật kiểm thử hộp đen, hộp trắng
* Kiểm thử tích hợp

**4: Cài đặt/thực thi (chi tiết) cho một chức năng (UC) trong thiết kế:**