• Lê Văn Cường – 14521116

• Phạm Hoài Nguyên – 14520605

• Phan Quang Duy – 14521161

**BÀI TẬP TUẦN 7:**

***Exercise 8.3 Give reasons why it is important to reuse programs instead of writing them from scratch.***

***Bài làm:***

Lí do tại sao nên sử dụng lại chương trình thay vì viết lại chúng?

Năng suất của các kỹ sư phần mềm có thể được tăng lên rất nhiều vì giảm trong thời gian và nỗ lực mà có thể đã được chi cho đặc điểm kỹ thuật, thiết kế, thực hiện và kiểm tra các thay đổi. Có thể tái sử dụng, phù hợp, giảm thời gian và chi phí cần thiết để duy trì sản phẩm phần mềm. Để sử dụng lại được chứng minh, thời gian thực hiện để tìm kiếm các thư viện tái sử dụng và để hiểu, điều chỉnh và kết hợp thành phần tái sử dụng cần phải được đáng kể ít hơn cần thiết để ghi các thành phần từ đầu.

***Exercise 8.4 What benefits can be derived from reusing software?***

Các thành phần có khả năng dùng lại có xu hướng có các đặc điểm như tổng quát, sự gắn kết và khớp nối cao thấp, nhất quán về phong cách lập trình, mô đun và các tiêu chuẩn. Đây cũng là đặc điểm mà chương trình duy trì nên biểu lộ. Như vậy, một nỗ lực để cải thiện có thể dùng lại các thành phần phần mềm có thể đóng góp đáng kể vào việc tăng cường bảo trì của họ.

***Exercise 8.5 Compare and contrast the different approaches to reuse, giving examples of systems that can be obtained from each of these approaches.***

Bài làm:

Các cách tiếp tận để tái sử dụng:

* ***Composition-Based Reuse***
  + *Black-box reuse:* Black-box reuse là cách tiếp cận mà trong đó, thành phần được sử dụng lại mà không sửa đổi. Ví dụ: Các thành phần tái sử dụng có thể được tìm thấy trong UNIX và các ứng dụng toán học trong Fortran, như được phản ánh trong tỷ lệ cao của các kỹ sư phần mềm tái sử dụng chúng.
* *White-box reuse:*Trái ngược với Black-box reuse, White-box reuse là cách tiếp cận mà trong đó, thành phần được sử dụng sau được khi sửa đổi. Truy cập vào mã nguồn của các module cho phép người dùng thực hiện để sửa đổi các thành phần theo yêu cầu của hệ thống đích.
* ***Generation-Based Reuse***
  + **Application Generator Systems** : là hệ thống phần mềm được sử dụng để tạo ra các ứng dụng khác. Được cung cấp các đặc điểm kỹ thuật của đơn để được tạo ra được thể hiện trong một số ký hiệu, nó được tạo ra tự động.
  + **Transformation-Based Systems**: dựa trên sản phẩm được phát triển bằng cách sử dụng một cách tiếp cận theo đó chi tiết kỹ thuật cấp cao của hệ thống được chuyển đổi thông qua một số giai đoạn trong chương trình hoạt động.
  + **Evaluation of the Generator-Based Systems:** là những thành phần tái sử dụng bởi vì chúng được sử dụng để tạo ra các chương trình khác nhau.

***Exercise 8.6 You have just joined a team of software engineers in which you are the only one who has studied and practised software reuse and reusability. The company you work for has no reuse programme although they are willing to start one. You are asked to implement the reuse programme.***

* + ***What is the first step you would take?***
  + ***Outline the technical, managerial and organisational steps you***
  + ***would go through. What tactics do you need to employ in order for the programme to succeed?***
  + ***What difficulties do you anticipate and how would you overcome them?***

***Bài làm:***

Các bước cần làm khi thực hiện công việc bảo trì, trong một nhóm SE mà chỉ mình mình là đã học và thực hành với bảo trì phần mềm:

* Hiểu chương trình: Đây là bước đầu tiên và căn bản nhất, đơn giản vì bạn không thể bảo trì phần mềm hay thực hiện bất kì tác động nào có lợi đến nó được nếu bạn không hiểu. Hiểu những gì? Hiểu về cấu trúc, chức năng, cấu phần,…Việc hiểu vấn đề thì có thể tìm hiểu các log phần mềm, hoặc hiểu qua tính năng, đọc code, và nhất là tận dụng những người còn lại, vì họ đi trước và tham gia phần mềm, sẽ nhanh chóng hơn cho việc nắm bắt chương trình của bạn. Việc hiểu này nhằm xác định: bạn cần bảo trì cái gì, bảo trì như thế nào, trong thời gian bao lâu…
* Chọn module cần thiết để dùng: chọn tool cần thiết để thực hiện bảo trì cho một module nhất định. Tôi đang phân vân về việc khi bảo trì bạn nên đi từ cái bao quát đến cái tiểu tiết, hay là ngược lại. Tôi nghĩ cái nào cũng hay cả. Nên việc xác định module nào cần làm trước tiên, làm thế nào, dùng tool nào hỗ trợ là dựa trên kinh nghiệm rồi. Lấy ra những module cần thiết cho việc tái sử dụng là quan trọng, vì không phải cái nào cũng có thể lấy được, và tận dụng bao nhiêu % ở 1 module cũng là tùy thực tế mà xác định.
* Viết lại cho phù hợp. Viết thêm các module cần thiết.

***Khó khăn:*** Khó tiếp cận, đọc hiểu, tốn thời gian hơn là xây dựng một mô hình mới.