SOFTWARE PROCESSES

***The software process ″***

A structured set of activities required to develop a software system.

Many different software processes but all involve:

▪Specification – defining what the system should do; and all constraints the system should have.

▪ Design and implementation – defining the organization of the system and implementing the system;

▪ Validation – checking that it does what the customer wants;

▪ Evolution – changing the system in response to changing customer needs.

***Chu trình phần mềm:***

Một tập hợp cấu trúc của các hoạt động cần thiết để phát triền một hệ thống phần mềm:

Nhiều chu trình phần mềm khác nhau nhưng tất cả đều liên quan đến:

▪ Đặc tả - xác định những gì hệ thống nên làm và tất cả các hạn chế mà hệ thống nên có.

▪ Thiết kế và thực hiện – xác định sự tổ chức của hệ thống và triển khai hệ thống

▪ Xác nhận – kiểm tra xem nó làm được những gì khách hàng mong muốn

▪ Tiến hóa – thay đổi hệ thống để áp ứng nhu cầu thay đổi của khách hàng.

***Software process descriptions***

When we describe and discuss processes, we usually talk about the activities in these processes such as specifying a data model, designing a user interface, etc. and the ordering of these activities. ″

Process descriptions may also include:

▪ Products, which are the outcomes of a process activity;

▪ Roles, which reflect the responsibilities of the people involved in the process;

▪ Pre- and post-conditions, which are statements that are true before and after a process activity has been enacted or a product produced.

Khi chúng ta miêu tả và thảo luận về các quy trình, chúng ta thương nói về các hoạt động trong các chu trình như chỉ định một mô hình dữ liệu, thiết kế giao diện người dùng…và trình tự của các hoạt động này.

Sự mô tả quy trình cũng có thể bao gồm:

▪ Sản phẩm – là kết quả của một chu trình hoạt động

▪ Vai trò phản ánh trách nhiệm của những người liên quan đến quá trình

▪ Điều kiện trước và sau – là những phát biểu đúng trước và sau khi một hoạt động quy trình được thực hiện hoặc một sản phẩm được sản xuất.

***Plan-driven and agile processes***

Plan-driven processes are processes where all of the process activities are planned in advance and progress is measured against this plan.

In agile processes, planning is incremental and it is easier to change the process to reflect changing customer requirements.

In practice, most practical processes include elements of both plan-driven and agile approaches.

There are no right or wrong software processes.

***Quy trình theo kế hoạch và linh hoạt:***

Các quy trình theo kế hoạch là các quy trình trong đó tất cả các hoạt động của chu trình được lên kế hoạch trước và tiến độ được đo lường theo kế hoạch này.

Trong các quy trình linh hoạt, việc lập kế hoạch là mang tính gia tăng và việc thay đổi quy trình để phản ánh các yêu cầu đang thay đổi của khách hàng là dễ dàng hơn.

Trong thực tế, hầu hết các quy trình thực tế đều bao gồm các yếu tố của cả phương pháp tiếp cận theo kế hoạch và phương pháp linh hoạt.

Không có quy trình phần mềm đúng hay sai.

***Software process models***

*The waterfall model*

▪ Plan-driven model. Separate and distinct phases of specification and development.

*Incremental development*

▪ Specification, development and validation are interleaved. May be plan-driven or agile.

*Reuse-oriented software engineering*

▪ The system is assembled[lắp-ráp] from existing components. May be plan-driven or agile.

In practice, most large systems are developed using a process that incorporates elements from all of these models.

***Mô hình quy trình phần mềm:***

*Mô hình thác nước:*

▪ Mô hình theo kế hoạch. Các giai đoạn đặc tả và phát triển riêng biệt.

*Phát triển gia tăng:*

▪ Đặc tả, phát triền và xác nhận được xen kẽ. Có thể theo kế hoạch hoặc 1 cách nhanh nhẹn.

*Kỹ thuật phần mềm hướng tái sử dụng:*

▪ Hệ thống được lắp ráp từ các linh kiện có sẵn. Có thể theo kế hoạch hoặc 1 cách nhanh nhẹn.

Trong thực tế, hầu hết các hệ thống lớn đều được phát triền bằng cách sử dụng quy trình kết hợp các phần tử từ tất cả các mô hình này.

*Mô hình thác nước:*

Hoạt động và bảo trỳ

Tổng hợp và kiểm tra hệ thống

Thiết kế hệ thống và phần mềm

Định nghĩa yêu câù

Thực hiên và kiểm tra bộ phận

***Waterfall model phases***

There are separate identified phases in the waterfall model:

▪ Requirements analysis and definition

▪ System and software design

▪ Implementation and unit testing

▪ Integration and system testing

▪ Operation and maintenance

The main drawback of the waterfall model is the difficulty of accommodating change after the process is underway. In principle, a phase has to be completed before moving onto the next phase

***Các giai đoạn của hệ thống thác nước:***

Có các giai đoạn được xác định riêng biệt trong mô hình thác nước:

▪ Phân tích và xác định yêu cầu

▪ Thiết kế hệ thống và phần mềm

▪ Triển khai và thử nghiệm bộ phân đơn vị

▪ Tích hợp và thử nghiệm hệ thống

▪ Vận hành và bảo trì

Hạn chế chính của mô hình thác nước là là khó khăn trong việc thích ứng với sự thay đổi sau chu trình. Về nguyên tắc, phải hoàn thành một giai đoạn trước khi chuyển vào giai đoạn tiếp theo.

***Waterfall model problems***

Inflexible partitioning of the project into distinct stages makes it difficult to respond to changing customer requirements.

▪ Therefore, this model is only appropriate when the requirements are well-understood and changes will be fairly limited during the design process.

▪ Few business systems have stable requirements.

The waterfall model is mostly used for large systems engineering projects where a system is developed at several sites.

▪ In those circumstances, the plan-driven nature of the waterfall model helps coordinate the work.

***Vấn đề của mô hình thác nước***

Phân chia dự án thành các giai đoạn riêng biệt một cách không linh hoạt gây khó khăn cho việc đáp ứng vơi sự thay đổi yêu cầu của khách hàng.

▪ Do vậy, mô hình này chỉ phù hợp khi yêu cầu được hiểu rõ và những thay đổi sẽ khá hạn chết trong quá trình thiết kế.

▪ Một vài hệ thống kinh doanh có yêu cầu ổn định

Mô hình thác nước được sử dụng chủ yếu cho các hệ thống lớn, các dự án kỹ thuật trong đó một hệ thống được phát triển tại 1 số trang web.

▪ Trong hoàn cảnh đó, tính chất quy hoạch của mô hình thác nước giúp điều phối công việc.

***Incremental development – phát triển gia tăng***

PHIÊN BẢN CUỐI CÙNG

PHIÊN BẢN TRUNG CẤP

PHIÊN BẢN ĐẦU TIÊN

SỰ XÁC NHẬN

SỰ PHÁT TRIỂN

THÔNG SỐ KỸ THUẬT

PHÁC THẢO MÔ TẢ



***Incremental development benefits***

The cost of accommodating changing customer requirements is reduced.

▪ The amount of analysis and documentation that has to be redone is much less than is required with the waterfall model.

It is easier to get customer feedback on the development work that has been done.

▪ Customers can comment on demonstrations of the software and see how much has been implemented.

More rapid delivery and deployment of useful software to the customer is possible.

▪ Customers are able to use and gain value from the software earlier than is possible with a waterfall process.

***Lợi ích phát triển gia tăng***

Chi phí lưu trữ việc yêu cầu thay đổi của khách hàng ngắn đi.

▪ Khối lượng phân tích và tài liệu cần được làm việc ít hơn nhiều so với mô hình thác nước.

Dễ dàng nhận được phản hồi của khách hàng hơn về quá trình phát triển công việc đã được thực hiện.

▪ Khác hàng có thể bình luận về phần demo của phần mềm và xem đã thực hiện được bao nhiêu.

Cung cấp và triển khai phần mềm hữu ích nhanh hơn cho khách hàng là có thể.

▪ Khách hàng được sử dụng và nhận được giá trị từ phần mềm có thể sớm hơn so với quy trình thác nước.

***Incremental development problems***

System structure tends to degrade as new increments are added.

▪ Unless time and money is spent on refactoring to improve the software, regular change tends to corrupt its structure. Incorporating further software changes becomes increasingly difficult and costly.

***Phát triển các vấn đề gia tăng***

Cấu trúc hệ thống có xu hướng giảm dần khi các phần tăng mới được thêm vào.

▪ Trừ khi dành thời gian và tiền bạc cho việc tái cấu trúc để cải thiện phần mềm, sự thay đổi thường xuyên có xu hướng làm hỏng cấu trúc của nó. Việc kết hợp thêm thay đổi phần mềm trở nên khó khăn và tốn kém.

***Reuse-oriented software engineering***

Based on systematic reuse where systems are integrated from existing components or COTS (Commercial-off-the-shelf) systems.

Process stages

▪ Component analysis;

▪ Requirements modification;

▪ System design with reuse;

▪ Development and integration.

Reuse is now the standard approach for building many types of business system

***Kỹ thuật phần mềm hướng tái sử dụng***

Dựa trên việc tái sử dụng có hệ thống khi các hệ thống được tích hợp từ các thành phần hiện có hoặc COTS

Các giai đoạn sử lý:

▪ Phân tích thành phần

▪ Sửa đổi yêu cầu

▪ Thiết kế hệ thống có khả năng tái sử dụng

▪ Phát triển và tích hợp

Việc tái sử dụng hiện nay là cách tiếp cận tiêu chuẩn để xây dựng nhiều loại hệ thống kinh doanh.