Review 1

**Hãy trình bày chi tiết quy trình phát triển hệ thống thông tin (Software Life Development Cycle). Theo các bạn, trong quy trình phát triển hệ thống giai đoạn nào là quan trọng nhất? Tại sao?



(THE SYSTEM DEVELOPMENT METHOD)

Các phương pháp

- 1. Structured analysis
- 2. Object-Oriented analysis
- 3. Agile / Adaptive methods



1. PHƯƠNG PHÁP PHÂN TÍCH CẦU TRÚC (STRUCTURED ANALYSIS METHODOLOGIES) 1980



1.1 CÁC KHÁI NIỆM TRONG PHƯƠNG PHÁP PHÂN TÍCH CẦU TRÚC

Structured Analysis Leads to Structured Design and Structured Programming

Modern structured analysis

Events
Data flow diagrams
Entity-relationship diagram

Structured design

Structure charts (one for each event) that define program modules based on the data flow diagrams

Structured programming

Program each module using structured programming constructs

Figure 2-17

Structured Analysis

- Define what system needs to do (processing requirements)
- Define data system needs to store and use (data requirements)
- Define inputs and outputs
- Define how functions work together to accomplish tasks
- Data flow diagrams (DFD) and entity relationship diagrams (ERD) show results of structured analysis

Structured Design

- Technique developed to provide design guidelines
 - What set of programs should be
 - What program should accomplish
 - How programs should be organized into a hierarchy
- Modules are shown with structure chart
- Main principle of program modules
 - Loosely coupled module is independent of other modules
 - Highly cohesive module has one clear task

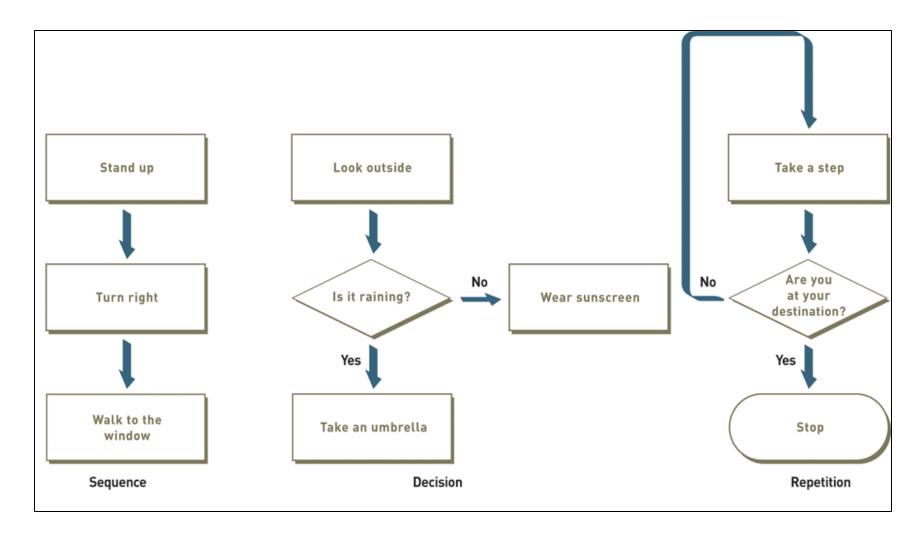
Top-down design

One programming approach that has proven to be most productive is called top-down decomposition. Top-down decomposition is the process of breaking the overall procedure or task into component parts (modules) and then subdivide each component module until the lowest level of detail has been reached. It is called top-down design or top-down decomposition since we start "at the top" with a general problem and design specific solutions to its sub problems.

Bottom – up design

- Already existing facilities/designs are used/taken into consideration as a model for a new (better) design.
- Using Bottom-up design strategy, we take an already existing computer program as a model for our new program. We try to utilize the existing facilities or design in a way, which gives out program a better performance.

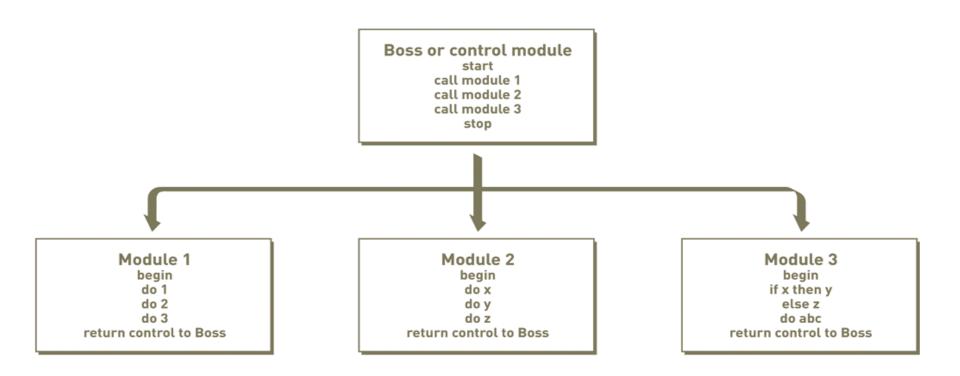
Three Structured Programming Constructs



Top-Down Programming

- Divides complex programs into hierarchy of modules
- The module at top controls execution by "calling" lower level modules
- Modular programming
 - Similar to top-down programming
- One program calls other programs to work together as single system

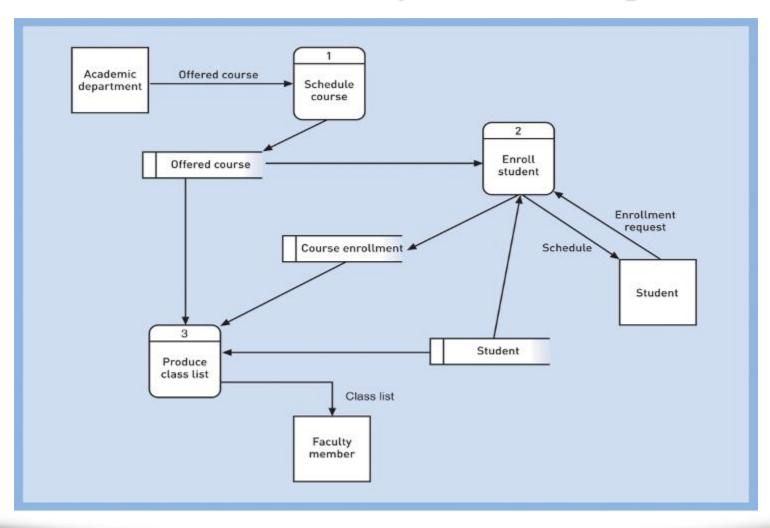
Top-Down or Modular Programming



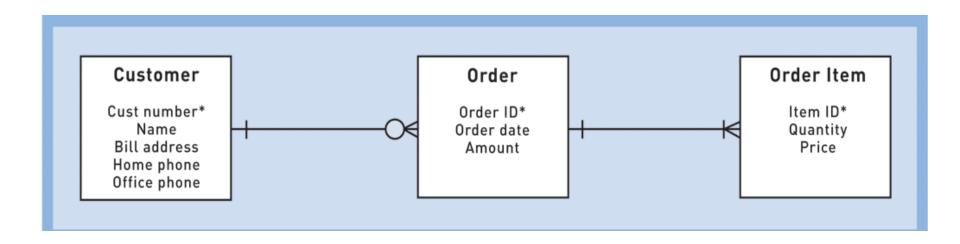


1.2 MODELING TOOLS

Data Flow Diagram (DFD) Created Using Structured Analysis Technique



Entity-Relationship Diagram (ERD) Created Using Structured Analysis Technique





Ưu điểm

- Xác định các yêu cầu hệ thống thời gian dài trước khi bắt đầu lập trình.
- Tối thiểu hóa sự thay đổi yêu cầu khi dự án bắt đầu.

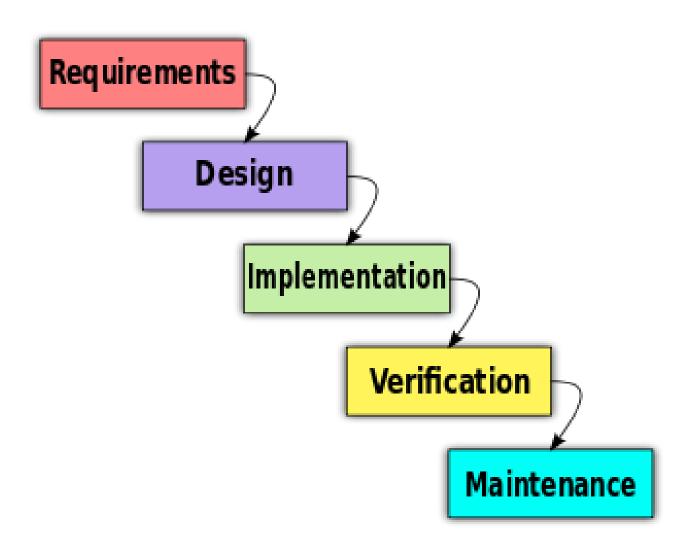
Nhược điểm

- Thiết kế phải hoàn thành trên giấy trước khi bắt đầu lập trình.
- Mất nhiều thời gian giữa việc hoàn thành các đề nghị hệ thống trong giai đoạn phân tích và bàn giao hệ thống.
- Có thể cũng phải điều chỉnh lại vì môi trường kinh doanh đã thay đổi.



1.3 WATER FALL DEVELOPMENT

Water fall model



Water fall model

BRAND CAMP

by Tom Fishburne

THE NEW PRODUCT WATERFALL



HOW DO WE CHART OUR ENTIRE COURSE IF WE DON'T KNOW WHAT'S AHEAD?

HAPPENS, JUST KEEP PADDLING! I WISH WE'D DESIGNED FOR THIS SCENARIO UPFRONT



TEST

PATCH IT AS BEST WE CAN. NO TIME TO CHANGE COURSE NOW



TOM FISHBURNE. COM

PLAN

BUILD



1.3 PARALLEL DEVELOPMENT

Parallel development

• Methodology attempts to address the problem of long delays between the analysis phase and the delivery of the system.

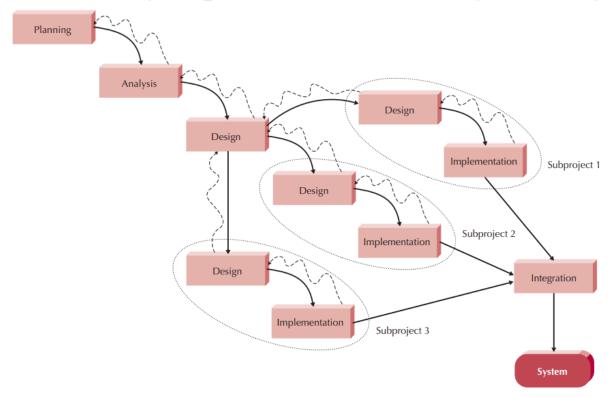


FIGURE 1-3 A Parallel Development-based Methodology

Question?

- Một ngân hàng muốn xây dựng hệ thống mobile banking dựa trên cơ sở dữ liệu hiện. "Bottom up design" và "top down design" cách thiết kế nào sẽ được sử dụng để triển khai?
- **Hãy trình bày khái niệm thiết kế theo hướng từ trên xuống (top-down design) và thiết kế theo hướng từ dưới lên (bottom-up design). Theo các bạn, trong môi trường làm việc outsourcing thì thường sử dụng hình thức thiết kế nào? Tại sao?

Question?

- Một công ty A đã xây dựng xong một hệ thống quản lý việc đăng ký và sắp xếp thời khóa biểu cho sinh viên giang viên. Công ty A nhận được hợp đồng triển khai hệ thống ở trường C, trường B. Nếu được lựa chọn phương pháp để phát triển hệ thống tại trường C, B bạn sẽ chọn các nào "Water Fall" hay "Parallel"? Vì sao?
- Hiện tại khi xây dựng một project, bạn đang tuân thủ theo cách phát triển nào? Vì sao?
- Khi nào bạn sử dụng "Water Fall"?



2. PHƯƠNG PHÁP PHÂN TÍCH HƯỚNG ĐỐI TƯỢNG (OBJECT-ORIENTED)



2 PHƯƠNG PHÁP PHÂN TÍCH HƯỚNG ĐỐI TƯỢNG

2.1 CÁC KHÁI NIỆM TRONG PHƯƠNG PHÁP PHÂN TÍCH HƯỚNG ĐỐI TƯỢNG

Object-Oriented Approach

- Object-oriented analysis (OOA)
 - Defines types of objects users deal with
 - Shows use cases are required to complete tasks
- Object-oriented design (OOD)
 - Defines object types needed to communicate with people and devices in system
 - Shows how objects interact to complete tasks
 - Refines each type of object for implementation with specific language of environment
- Object-oriented programming (OOP)
 - Writing statements in programming language to define what each type of object does



2.2 MODELING TOOLS

UML (Unified Modeling Language)

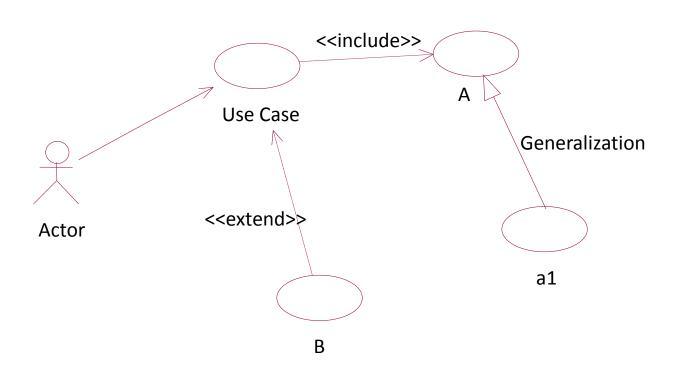
- UML là ngôn ngữ mô hình hướng đối tượng hợp nhất các ngôn ngữ trước đây – có sự đóng góp của nhiều người: IBM, Microsoft + Booch, Rumbaugh và Jacobson
 - Phát triển bởi Rational Software Corporation (hiện nay là một phần của IBM Software Group)
 - Xác nhận bởi OMG Object Management Group (www.OMG.org)
 năm 1997
 - Một tập các ký hiệu chuẩn có ngữ nghĩa được định nghĩa rõ ràng.
 - Ngôn ngữ truyền thông thống nhất giữa các nhóm phát triển
 - Tiếp cận hướng đối tượng
 - Quy trình phát triển UP Unified Process là phương pháp luận của
 UML

UML- Models and Views

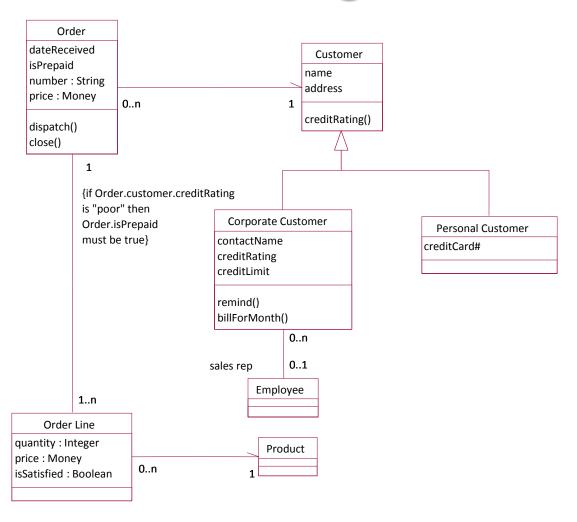
- Structural Modelling: cấu trúc tĩnh của dữ liệu
 - Static View (Class Diagram)
- Behavioural Modelling
 - Functional View (Use-case, Activity Diagrams)
 - Dynamic View (Interaction Diagrams sequence and collaboration diagram)
 - Temporal View (State Chart Diagram): mô hình sự thay đổi trạng thái các đối tượng qua thời gian.
- Architectural Modelling
 - Implementation view (Component Diagram)
 - Configuration view (Deployment Diagram)

Use Case Diagram

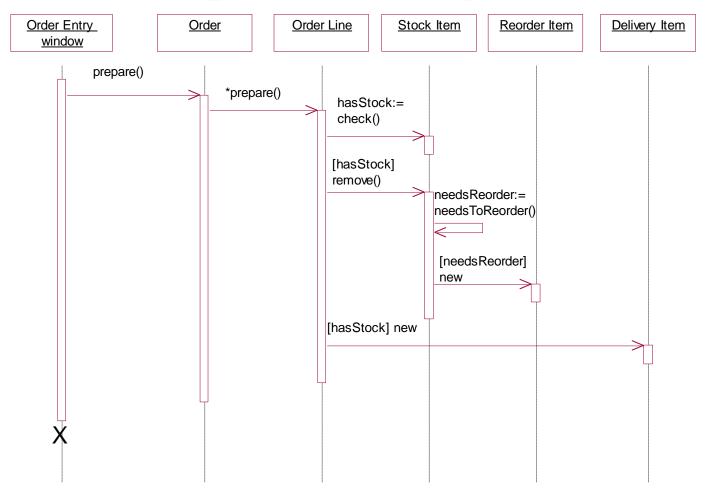
Use Case Diagram



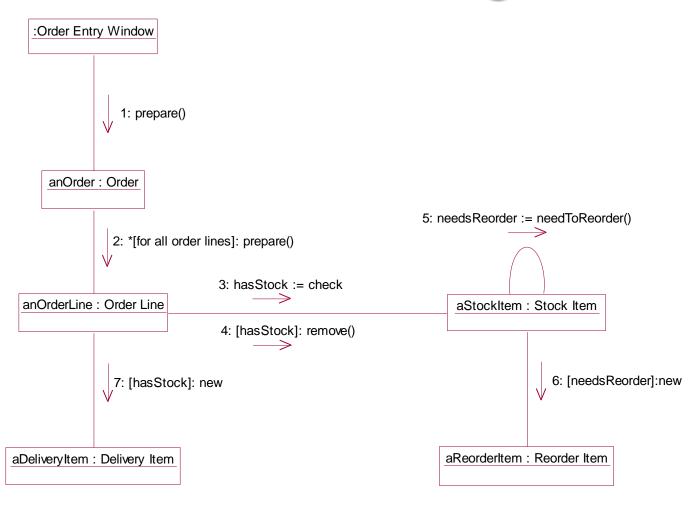
Class Diagram



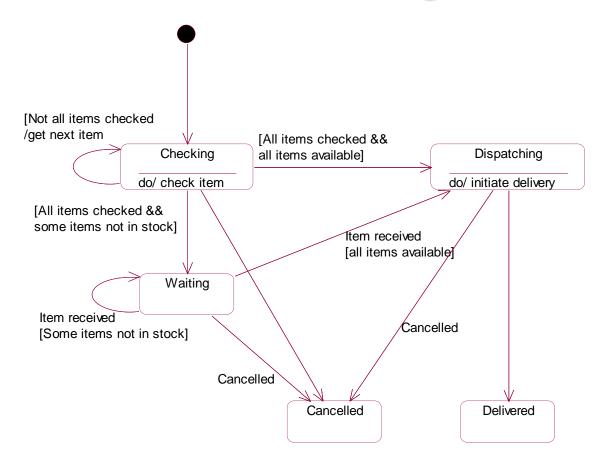
Sequence Diagram



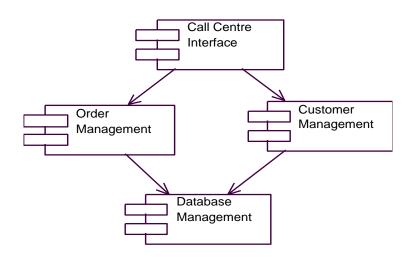
Collaboration Diagram



Statechart Diagram



A UML Component Diagram showing the relationship between components that make up a call centre system



Component diagrams show how the system is organised into cooperating components; components are built out of objects.



2.3 ƯU VÀ NHƯỢC ĐIỂM

Ưu và nhược điểm

Ưu điểm

- Dễ dàng kết hợp với ngôn ngữ lập trình hướng đối tượng.
- Dễ dàng điều chỉnh và có tính tái sử dụng => giảm thiểu quá trình phát triển và giảm giá thành
- Dễ dàng bảo trì và mở rộng hệ thống.

Nhược điểm

- Trong một số hệ thống lớn sự tương tác giữa lớp và đối tượng khá phức tạp.
- Chưa quản lý tốt sự thay đổi về yêu cầu của người dùng trong quá trình triển khai cũng như sự thay đổi về nhân sự.

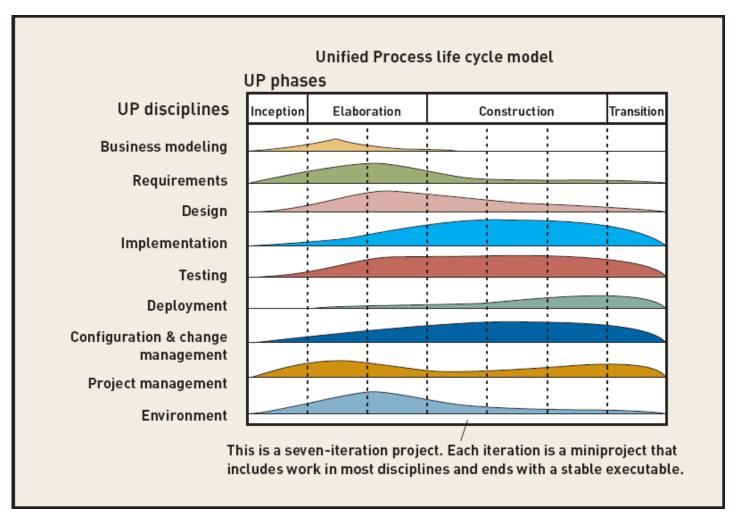


2.4 RUP MODEL

Relational Unified Process (RUP)

- Iterative and incremental
- Use-case-driven
- Architecture-centric
- Uses UML as its modeling notation
- Process framework
 - Comprehensive set of document templates, process workflow templates, and process guidelines
 - Distributed by IBM/Rational on a CD

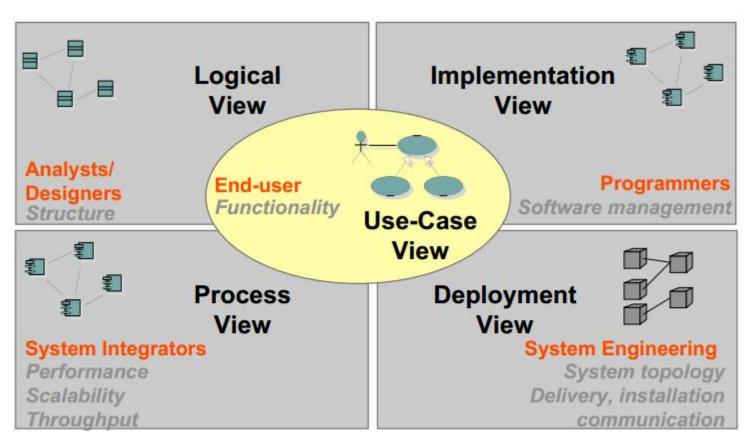
Unified Process Life Cycle

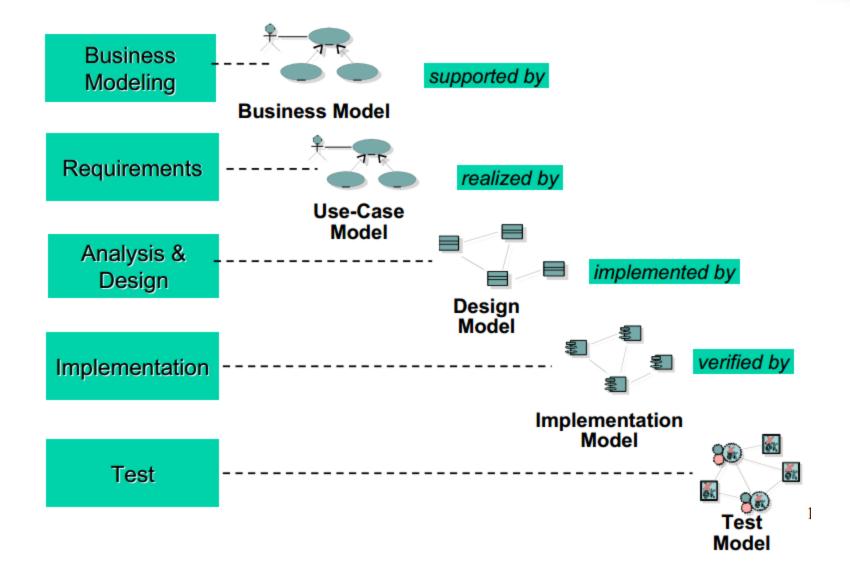


Rational Unified Process Is Use-Case-Driven

- Use cases are concise, simple, and understandable by a wide range of stakeholders
 - End users, developers and acquirers understand functional requirements of the system
- Use cases drive numerous activities in the process:
 - Creation and validation of the design model
 - Definition of test cases and procedures of the test model
 - Planning of iterations
 - Creation of user documentation
 - System deployment
- Use cases help synchronize the content of different models

Representing Architecture: The 4+1 View Model







3. PHƯƠNG PHÁP AGILE /ADAPTIVE



AGILE / ADAPTIVE METHOD

- Stresses intense team-based effort.
- Breaks development process down into cycles or iteration that add functionality.
- Each iteration is designed built and tested in ongoing process.
- Attempts to reduce major risks by incremental steps in short time intervals.

Agile SDLC's

- Speed up or bypass one or more life cycle phases
- Usually less formal and reduced scope
- Used for time-critical applications
- Used in organizations that employ disciplined methods

Some Agile Methods

- Scrum
- Extreme Programming (XP)
- Adaptive Software Development (ASD)
- Feature Driven Development (FDD)
- Crystal Clear
- Dynamic Software Development Method (DSDM)
- Incremental SDLC



Ưu điểm

- Very flexible and efficient in dealing with change.
 Stresses team interaction and reflects a set of community-based values.
- Frequent deliverables constantly validate the project and reduce risk.

Nhược điểm

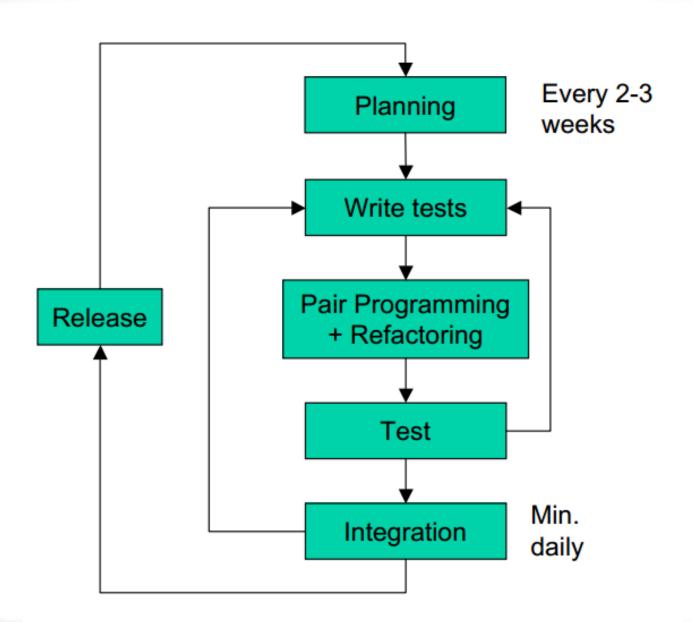
- Team members need a high level of technical and communications skills.
- Lack of structure and documentation can intro duce risk factors.
- Overall project might be subject to scope change as user requirements change.



3.3 XP METHOD

Extreme Programming (XP)

- Formulated in 1999 by Kent Beck, Ward Cunningham and Ron Jeffries
- Agile software development methodology (others: Scrum, DSDM)
- Developed in reaction to high ceremony methodologies



The Five Values of XP

- Communication
- Simplicity
- Feedback
- Courage
- Respect

XP Team and Roles

- Client
- Programmer
- Manager
- Tracker
- Coach

Client

- Answer questions
- Help in writing the tests (as requirements)
- Drive the iteration by selecting the priority tasks of a release
- Can change the requirements
- Valid the delivery

Programmer

- Testing, coding, refactoring
 - instead of designing, coding, testing
- Pair programming
- Test-driven development
- Continuous integration

Manager

- Present the project to the outside world (advancements, delivery)
- Get the resources (human, material, etc.)
- Pilot the team (organize the iteration and release meetings)
- Handle any problems (with the client, the project team members, etc.)

Tracker

- Follow the delivery schedule (user story)
- Follow the iteration schedule (tasks)
- Review the validation of the tests

Coach

- Monitor the whole process (meetings overflowing, etc. ...)
- Maintain the XP rules
- Change the process if necessary
- Mentor (often excellent programmer)

Lightweight vs. Heavyweight Processes

Heavyweight

e.g., V-Process

Customizable

Framework

e.g., Rational

Unified

Process (RUP)

Agile (Lightweight)

e.g., eXtreme

Programming (XP)

Document driven

Elaborate workflow definitions

Many different roles

Many checkpoints

High management overhead

Highly bureaucratic

Focus on working code rather than documentation

Focus on direct communication

(between developers and

between developers and the customer)

Low management overhead



SELECTING THE APPROPRIATE DEVELOPMENT METHODOLOGY

Selecting the Appropriate Development Methodology

	Structured Methodologies			RAD Methodologies		Agile Methodologies
Ability to Develop Systems	Waterfall	Parallel	Phased	Prototyping	Throwaway Prototyping	ХР
with Unclear User Requirements	Poor	Poor	Good	Excellent	Excellent	Excellent
with Unfamiliar Technology	Poor	Poor	Good	Poor	Excellent	Poor
that are Complex	Good	Good	Good	Poor	Excellent	Poor
that are Reliable	Good	Good	Good	Poor	Excellent	Good
with a Short Time Schedule	Poor	Good	Excellent	Excellent	Good	Excellent
with Schedule Visibility	Poor	Poor	Excellent	Excellent	Good	Good

FIGURE 1-8 Criteria for Selecting a Methodology

References

- Software Engineering 8 Edition, SommerVille
- Systems Analysis and Design 9 Edition, Shelly CashMan Series
- Systems Analysis and Design 8 Edition, KenDall
- System Analysis and Design with UML2, Alan Dennis and cooperator

Exercise

- **Hãy trình bày chi tiết về quy trình Relational Unified Process (RUP).
- **Hãy trình bày chi tiết về mô hình "4+1" view.
- ***Hãy trình bày, so sánh những ưu và nhược điểm khi triển khai hệ thống theo hướng hướng đối tượng (đặc trưng là quy trình RUP) và khi triển khai hệ thống bằng phương pháp Agile (Scrum/Extreme Programming -XP).



