TRƯỜNG ĐẠI HỌC THUỶ LỢI

KHOA CÔNG NGHỆ THÔNG TIN
BỘ MÔN CÔNG NGHỆ PHẦN MỀM

Bài giảng

CÔNG NGHỆ PHẦN MỀM

Hà Nội - 02/2020

TRƯỜNG ĐẠI HỌC THUỶ LỢI

Khoa Công Nghệ Thông Tin
Bộ Môn Công Nghệ Phần Mềm

CÔNG NGHỆ PHẦN MỀM

Giảng viên: TS. Lê Nguyễn Tuấn Thành
Email: thanhlnt@tlu.edu.vn
ĐT: 0898158945

TRƯỜNG ĐẠI HỌC THUỶ LỢI

Khoa Công Nghệ Thông Tin
Bộ Môn Công Nghệ Phần Mềm

CÔNG NGHỆ PHẦN MỀM

Giảng viên: TS. Lê Nguyễn Tuấn Thành
Email: thanhlnt@tlu.edu.vn
ĐT: 0898158945

NỘI DUNG

- Tên môn học: *Công nghệ Phần mềm*
- Tên tiếng Anh: *Software Engineering*
- Mã môn học: *CSE481*
- Số tín chỉ: *3 (LT: 2, TH/BT/TL: 1)*
- Số tiết: *30 Lý thuyết & 15 - Bài tập, Thảo luận*

MỤC ĐÍCH MÔN HỌC

- Cung cấp những khái niệm cơ bản về:
  - Phần mềm (Software)
  - Công nghệ phần mềm (Software Engineering)
  - Dự án phần mềm (Software Project)
  - Quản trị dự án phần mềm (Software Project Management)
  - Tiến trình phần mềm (Software Process)
- Giúp sinh viên biết và hiểu:
  - Quy trình xây dựng phần mềm cùng với một số phương pháp xây dựng phần mềm
  - Các công việc thực hiện và kết quả đạt được trong các giai đoạn: *khảo sát, phân tích, thiết kế, cài đặt, kiểm thử, bảo trì*
  - Các kiến trúc và mô hình triển khai phần mềm
  - Các vấn đề đạo đức và chuyên môn

NỘI DUNG MÔN HỌC

- Bài 1: Phần mềm và Công nghệ phần mềm
- Bài 2: Xác định yêu cầu
- Bài 3: Phân tích và thiết kế hệ thống thông tin
- Bài 4: Quản lý chất lượng phần mềm
- Bài 5: Mô hình CMMI

YÊU CẦU VỚI SINH VIÊN

- Dự lớp đầy đủ
- Tham gia thảo luận (trên lớp hoặc qua Piazza), thực hành
  - Website: *https://sites.google.com/site/cse481spring2017/*
  - Trang web thảo luận: *Piazza*
- **Cách đánh giá:**
  - Điểm quá trình (thực hành, chuyên cần): 50%
  - Điểm thi hết môn: 50%

BÀI TẬP DỰ ÁN (PROJECT)

- Mô hình phát triển: SCRUM
- Số lượng thành viên: ~5 người
- Chủ đề:
  - Nhóm tự đề xuất, hoặc
  - Do giảng viên chỉ định
- Nền tảng: không giới hạn (*web, desktop, mobile)*
- Công nghệ: không giới hạn
- Số tính năng không cần nhiều
- Phải sử dụng các công cụ quản lý dự án: *redmine*, *github*, *bitbucket*, *trello*, *slack, jira, confluence, gitlab, jenkins (CI/CD)*…

TÀI LIỆU THAM KHẢO

- ***Software Engineering***, Ian Sommerville, 10th Edition, 2016
- *Software engineering: A practitioner's approach*, Part 1 & Part 2, Roger S. Pressman, McGraw-Hill Higher Education, 2010. (#000021579)
- *Kỹ nghệ Phần mềm*, Lê Văn Phùng, Nhà xuất bản Thông tin và Truyền thông, 2014.
- *Nhập Môn Công Nghệ Phần Mềm*, Phạm Thị Quỳnh

CÔNG NGHỆ PHẦN MỀM

BÀI1
TỔNG QUAN VỀ PHẦN MỀM VÀ CÔNG NGHỆ PHẦN MỀM

Giảng viên: TS. Lê Nguyễn Tuấn Thành
Email: thanhlnt@tlu.edu.vn

Bộ Môn Công Nghệ Phần Mềm – Khoa CNTT
Trường Đại Học Thủy Lợi

Nội dung

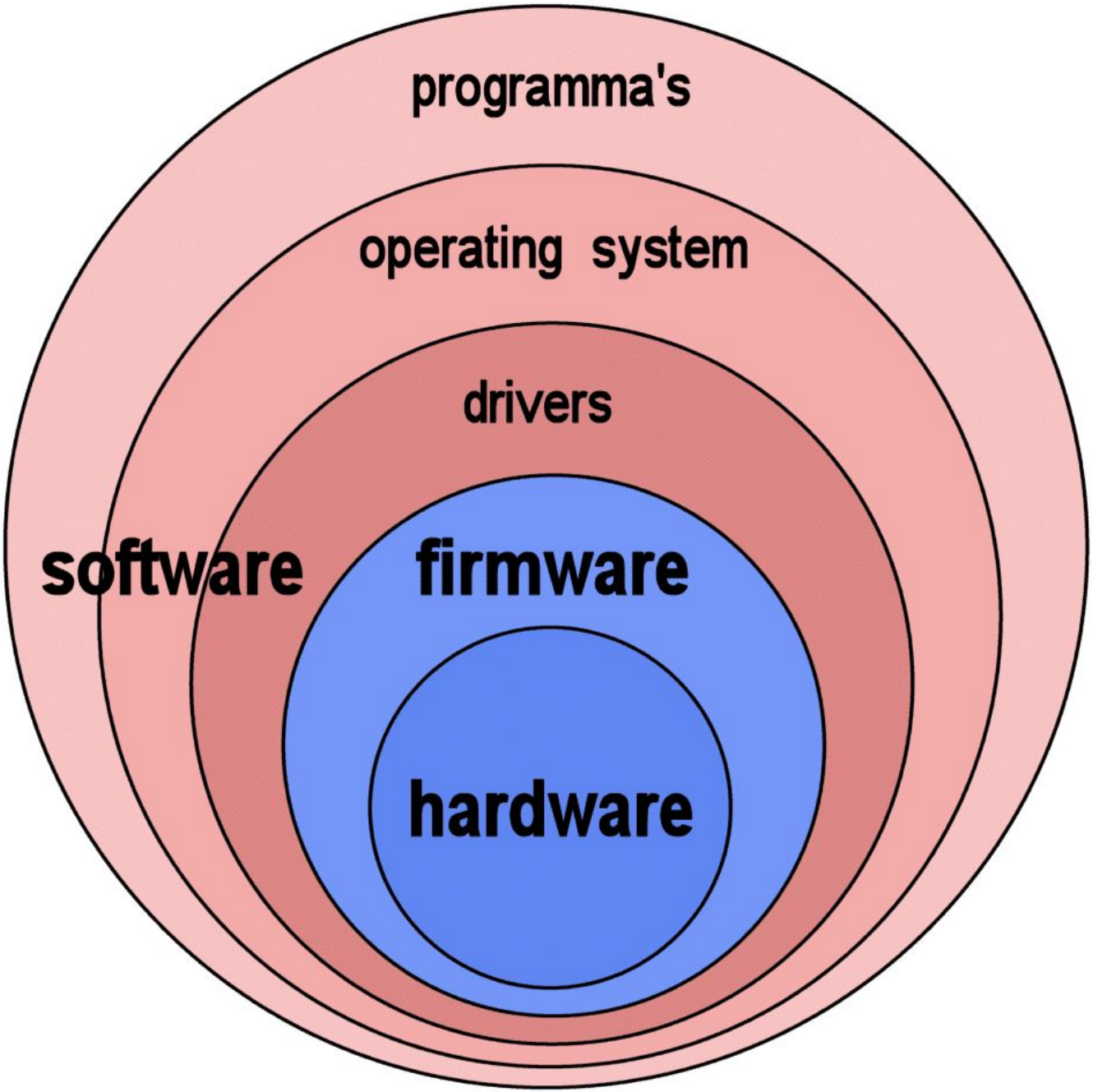
1. Phần mềm
2. Công nghệ phần mềm
3. Quản lý dự án phần mềm
4. Quy trình phần mềm

3

1. PHẦN MỀM LÀ GÌ?

What is the **Software**?


Flappy Bird


programma's
operating system
drivers
software
firmware
hardware

Định nghĩa

- ***Phần mềm là các <u>chương trình máy tính</u> và những <u>tài liệu liên quan</u>*** *(tài liệu đặc tả yêu cầu, tài liệu phân tích thiết kế, tài liệu lập trình, tài liệu kiểm thử, tài liệu hướng dẫn sử dụng, …)*
- Sản phẩm phần mềm được chia thành 2 loại:
  1. **Đại trà**
  2. **Chuyên biệt**

Thuộc tính của phần mềm tốt

- Khả năng bảo trì
- Đáng tin cậy
- Hiệu quả
- Chấp nhận được

Lịch sử Phát triển Phần mềm (1)

- **1950 – 1960**
  - Phần cứng thay đổi liên tục, phần lớn phần mềm được chuyên dụng cho ứng dụng đặc biệt
  - Phần mềm được coi là nghệ thuật
  - Phát triển phần mềm chưa được quản lý
  - Môi trường phần mềm có tính cá nhân dẫn đến thiết kế – tiến trình không tường minh, thường không có tài liệu
- 1960 – 1970
- 1970 – 1990
- Sau 1990

Lịch sử Phát triển Phần mềm (2)

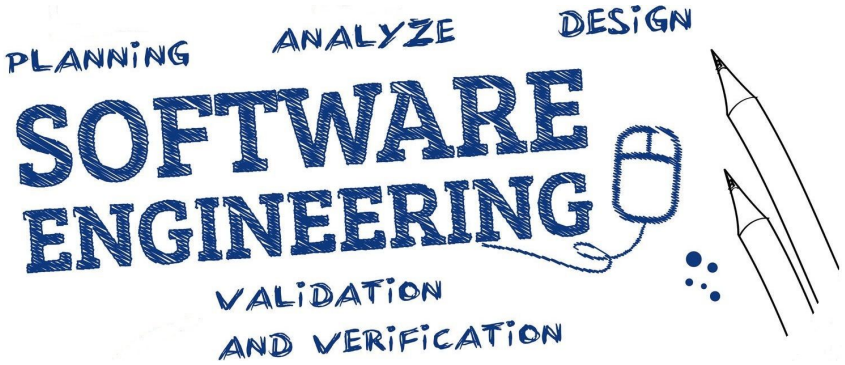
- 1950 – 1960
- **1960 – 1970**
  - Hệ thống đa người dùng dẫn đến khái niệm mới về tương tác người máy
  - Hệ thống thời gian thực
  - Tiến bộ lưu trữ trực tuyến dẫn đến thế hệ đầu tiên của hệ quản trị CSDL
  - Số lượng các hệ thống dựa trên máy tính phát triển dẫn đến thư viện phần mềm mở rộng
- 1970 – 1990
- Sau 1990

Lịch sử Phát triển Phần mềm (3)

- 1950 – 1960
- 1960 – 1970
- **1970 – 1990**
  - Hệ thống phân tán
  - Mạng diện rộng và cục bộ, số giải thông cao, tăng nhu cầu thâm nhập dữ liệu dẫn đến yêu cầu lớn phát triển phần mềm
  - Sử dụng phổ cập các bộ vi xử lý (ô tô, robot, lò vi sóng, …) máy tính cá nhân và các máy trạm
- Sau 1990

Lịch sử Phát triển Phần mềm (4)

- 1950 – 1960
- 1960 – 1970
- 1970 – 1990
- **Sau 1990**
  - Kỹ nghệ hướng đối tượng
  - Hệ chuyên gia và phần mềm trí tuệ nhân tạo
  - Phần mềm mạng nơ ron nhân tạo
  - Kỹ thuật phần mềm hướng thành phần (COM, DCOM của Microsoft, CORBA của OMG, JavaBeans, Enterprise JavaBeans của Sun)
  - Kỹ nghệ hướng dịch vụ (SOA)



12

2. CÔNG NGHỆ PHẦN MỀM LÀ GÌ?

What is the **Software Engineering**?

Công nghệ Xây dựng

Cần phải làm những bước nào?

- Quyết định cần xây những thành phần gì
- Quyết định công trình sẽ trông như thế nào
- Cần xây bao nhiêu tầng, mỗi tầng bao nhiêu phòng, mỗi phòng rộng bao nhiêu, …
- Quyết định vật liệu xây dựng
- Lên kế hoạch dự án, lập lịch, làm việc nhóm
- Tiến hành xây dựng
- Vận hành và bảo trì

Định nghĩa

- ***Công nghệ phần mềm là một chuyên ngành kỹ thuật liên quan đến tất cả khía cạnh của việc sản xuất phần mềm***
- CNPM liên quan tới các *lý thuyết*, *phương pháp* và *công cụ* cho việc phát triển phần mềm chuyên nghiệp

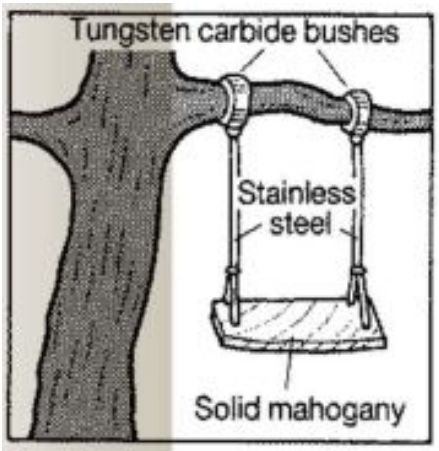

People
High quality products
Processes
Technology

CNPM không chỉ là kỹ năng công nghệ ...

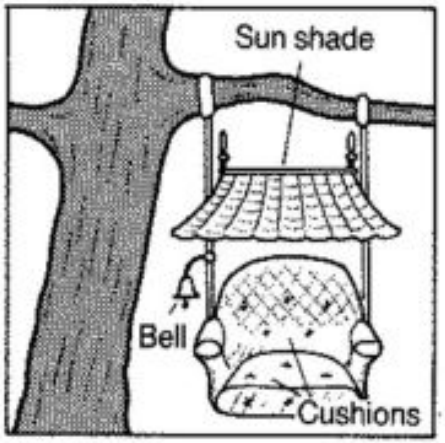
- Các trường đại học có xu hướng tập trung vào *công nghệ*, bỏ qua yếu tố *con người* và *quy trình* thực hiện
- Trong thực tế, lập trình thường chiếm *ít thời gian làm nhất* trong toàn bộ quá trình thực hiện dự án!
  - Thông thường, cần 20% thời gian để tìm hiểu yêu cầu, 25% thời gian cho kiểm thử, 40% cho thiết kế và **chỉ 15% cho coding !**

Đặc điểm của CNPM

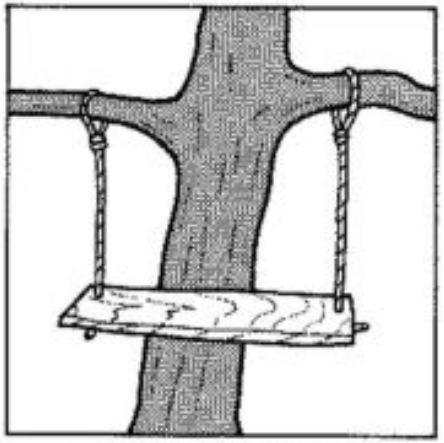
- Tài nguyên và thời gian bị giới hạn
- Đáp ứng các yêu cầu của khách hàng
- Quản lý rủi ro
- Làm việc nhóm



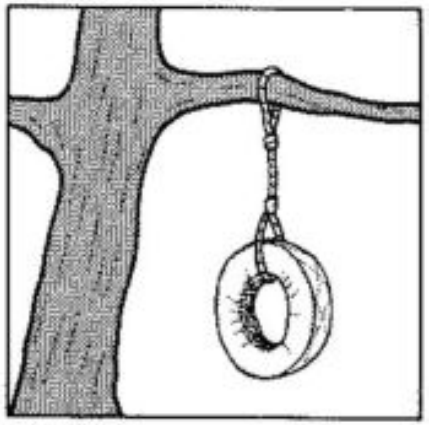
What Product Marketing specified



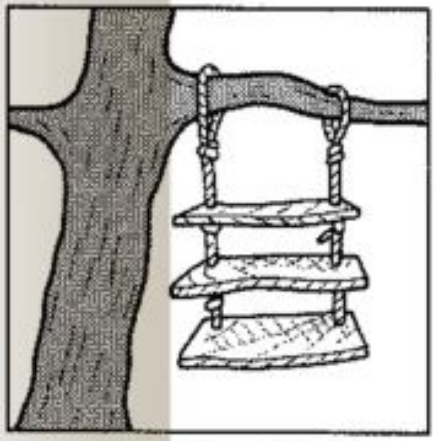
What the salesman promised



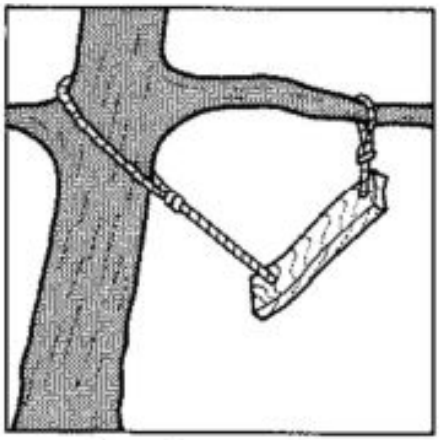
Design group's initial design



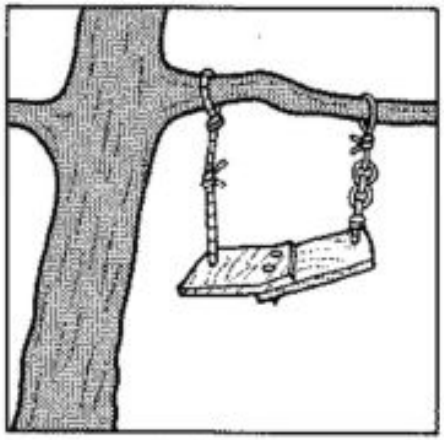
What the customer actually wanted



Corp. Product Architecture's modified design

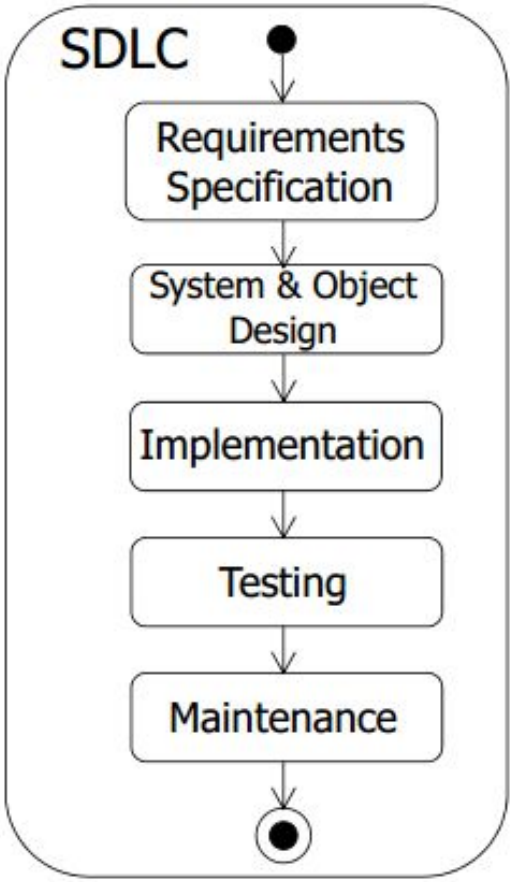


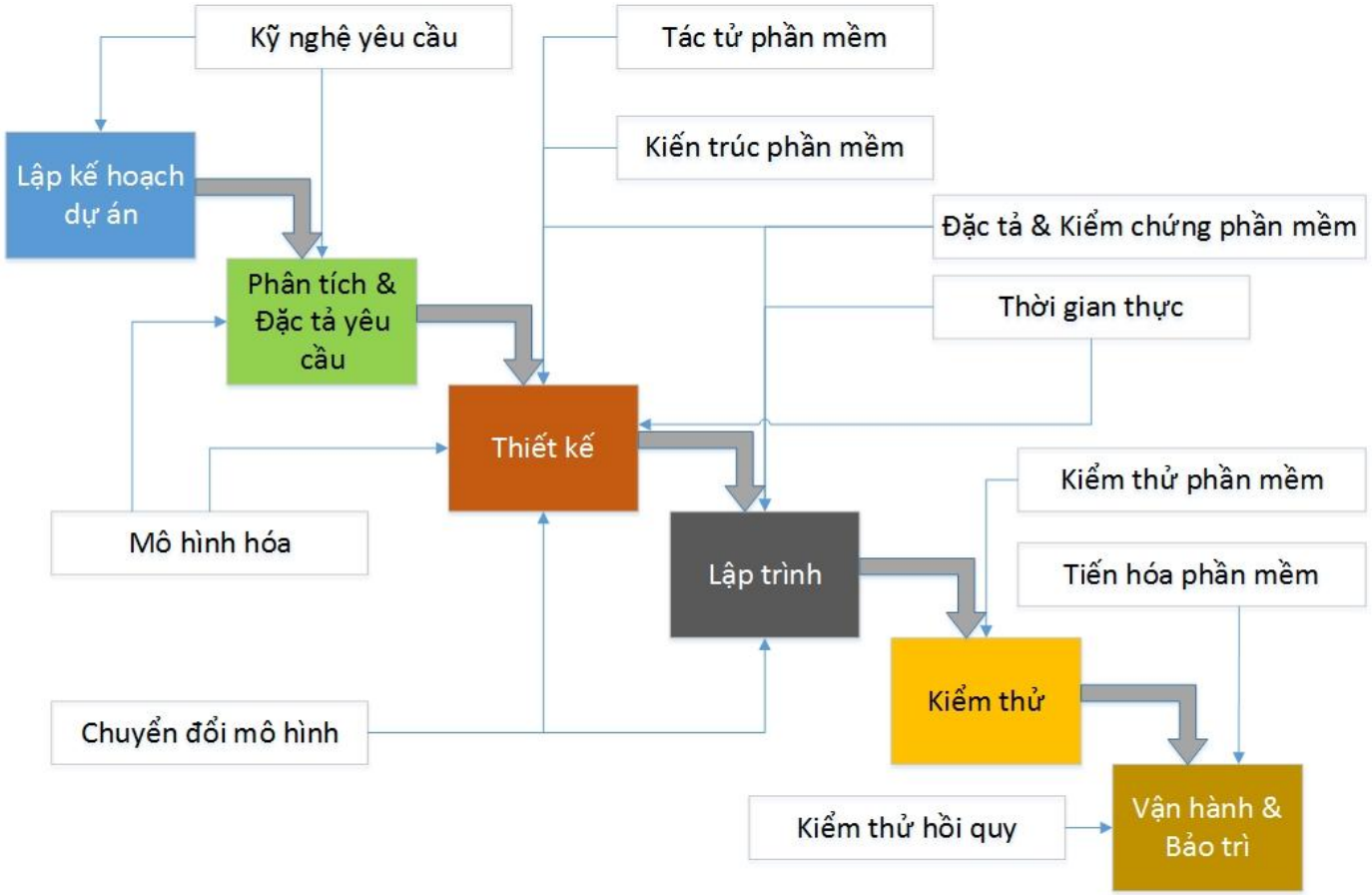
Pre-release version



General release version

Vòng đời phát triển phần mềm





Trách nhiệm Chuyên môn & Đạo đức

- CNPM không đơn thuần chỉ áp dụng những kỹ năng, kỹ thuật mà còn liên quan đến những trách nhiệm (chuyên môn & đạo đức)
- Hành vi đạo đức không chỉ đơn giản là tuân theo pháp luật
- 4 trách nhiệm chuyên môn & 8 nguyên tắc đạo đức

Trách nhiệm chuyên môn

1. Tính bảo mật
2. Năng lực
3. Quyền sở hữu trí tuệ
4. Lạm dụng máy tính

Bộ quy tắc đạo đức ACM/IEEE

Các hiệp hội chuyên nghiệp ở Hoa Kỳ đã hợp tác để đưa ra một bộ quy tắc đạo đức

1. Cộng đồng
2. Khách hàng và người sử dụng lao động
3. Sản phẩm
4. Đánh giá
5. Quản lý
6. Chuyên nghiệp
7. Đồng nghiệp
8. Bản thân

3. QUẢN LÝ DỰ ÁN PHẦN MỀM

Software Project Management

Chủ đề

- Những hoạt động quản lý
- Lập kế hoạch dự án
- Lập lịch trình dự án
- Quản lý rủi ro

Quản lý dự án phần mềm

- Liên quan đến những hoạt động để đảm bảo phần mềm được phân phối:
  - *Đúng hạn*
  - *Trong phạm vi ngân sách*
  - *Theo đúng lịch trình*
  - *Phù hợp với những yêu cầu chức năng, chất lượng* của công ty phát triển cũng như công ty đặt hàng phần mềm
- Quản lý dự án là **cực kỳ cần thiết !**

Những hoạt động quản lý chung

1. Viết đề xuất dự án (project proposal)
2. Lựa chọn và đánh giá nhân sự
3. Lập kế hoạch dự án (gồm lịch trình)
4. Lập chi phí dự án
5. Giám sát dự án và duyệt lại
6. Viết báo cáo và trình bày

HĐQL1: Viết đề xuất dự án

- Vấn đề cần giải quyết
- Mục tiêu
- Tiếp cận kỹ thuật
  - Vắn tắt về các chức năng chính
  - Thiết kế đề xuất
- Quản lý dự án
  - Lịch trình (những cột mốc chính)
  - Ngân sách (lương, phần cứng, phần mềm, …)
  - Thành viên nhóm
- Phụ lục

HĐQL2: Lập nhân dự dự án

- Khó khăn: không phải lúc nào cũng tìm được người lý tưởng cho dự án, do
  - Ngân sách dự án không cho phép sử dụng những nhân viên được trả lương cao
  - Nhân sự với khả năng thích hợp có thể không sẵn có
  - Công ty muốn phát triển thêm kỹ năng cho những nhân viên hiện có
  - …

HĐQL3: Lập kế hoạch dự án

- Thường là hoạt động quản lý tốn nhiều thời gian nhất, liên tục từ lúc hình thành khái niệm ban đầu đến khi phân phối hệ thống
  - Các kế hoạch phải thường xuyên được cập nhật khi có thông tin mới
- Kế hoạch dự án đề cập đến:
  - **Các nguồn lực có sẵn cho dự án**
  - **Phân chia công việc**
  - **Lịch trình cho công việc**

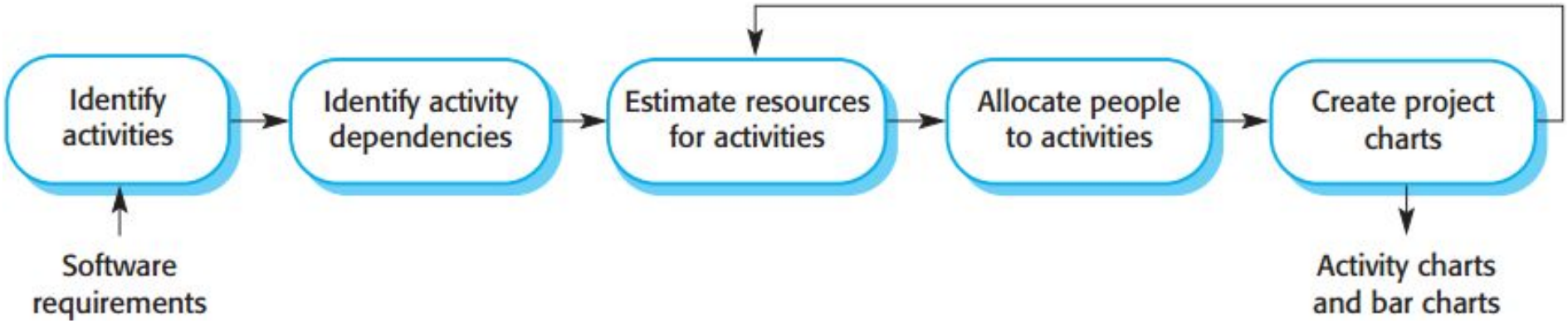
HĐQL3: Cấu trúc kế hoạch dự án

A. Giới thiệu
B. Tổ chức dự án
C. Phân tích rủi ro
D. Những yêu cầu tài nguyên phần cứng và phần mềm
E. Phân chia công việc
F. Lịch trình dự án
G. Những cơ chế báo cáo và giám sát

HĐQL3-F: Lập lịch trình dự án

- Chia dự án thành các tác vụ và **ước lượng** thời gian, tài nguyên để hoàn thành mỗi tác vụ
- **Lưu ý:**
  - Tối giản hóa những phụ thuộc giữa các tác vụ
  - Tổ chức những tác vụ sao cho tối ưu hóa việc sử dụng nhân lực

HDQL3-F: Quy trình lập lịch trình dự án



Bảng thời lượng tác vụ và những phụ thuộc

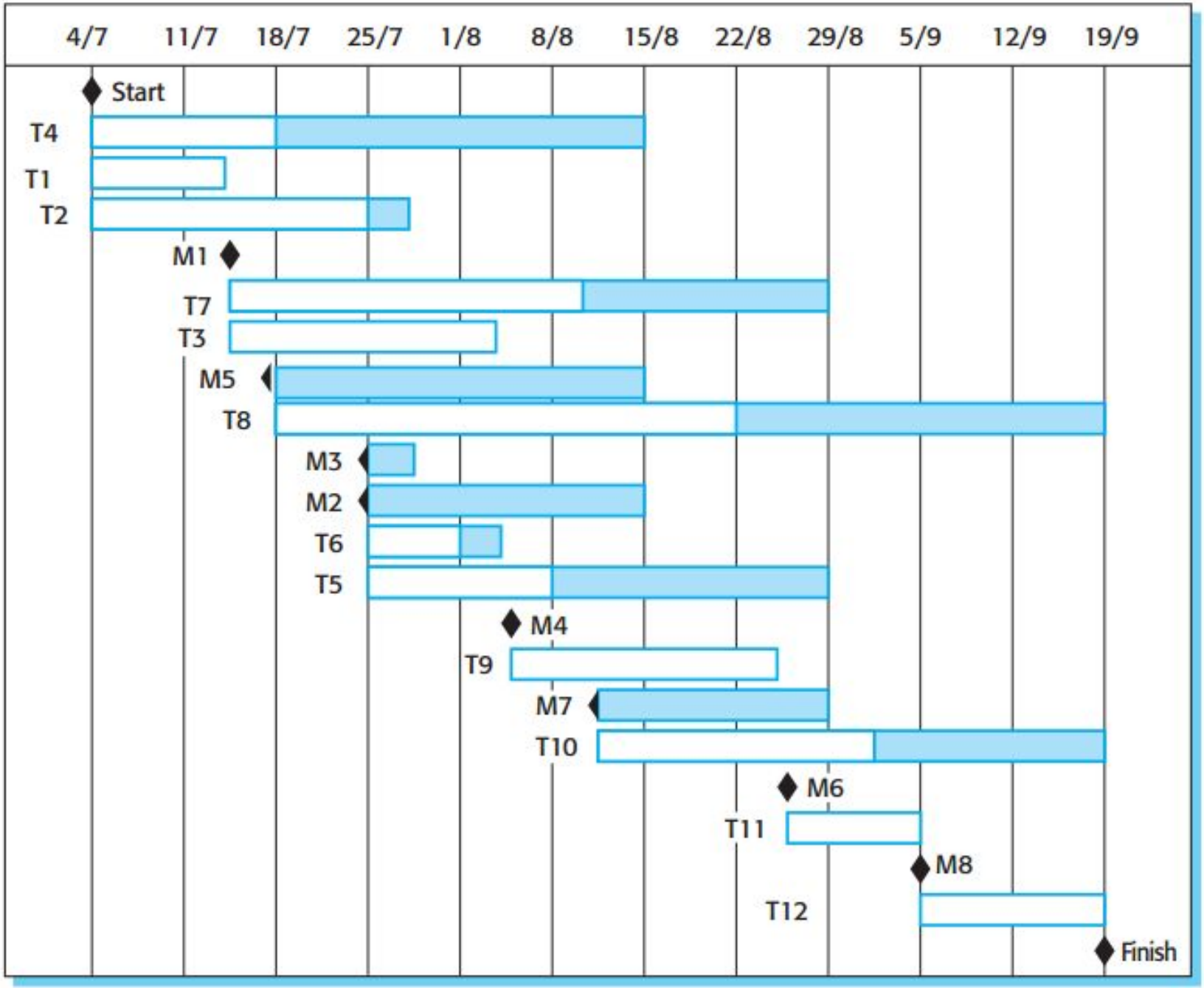
Task	Duration (days)	Dependencies
T1	8	
T2	15	
T3	15	T1 (M1)
T4	10	
T5	10	T2, T4 (M2)
T6	5	T1, T2 (M3)
T7	20	T1 (M1)
T8	25	T4 (M5)
T9	15	T3, T6 (M4)
T10	15	T5, T7 (M7)
T11	7	T9 (M6)
T12	10	T11 (M8)

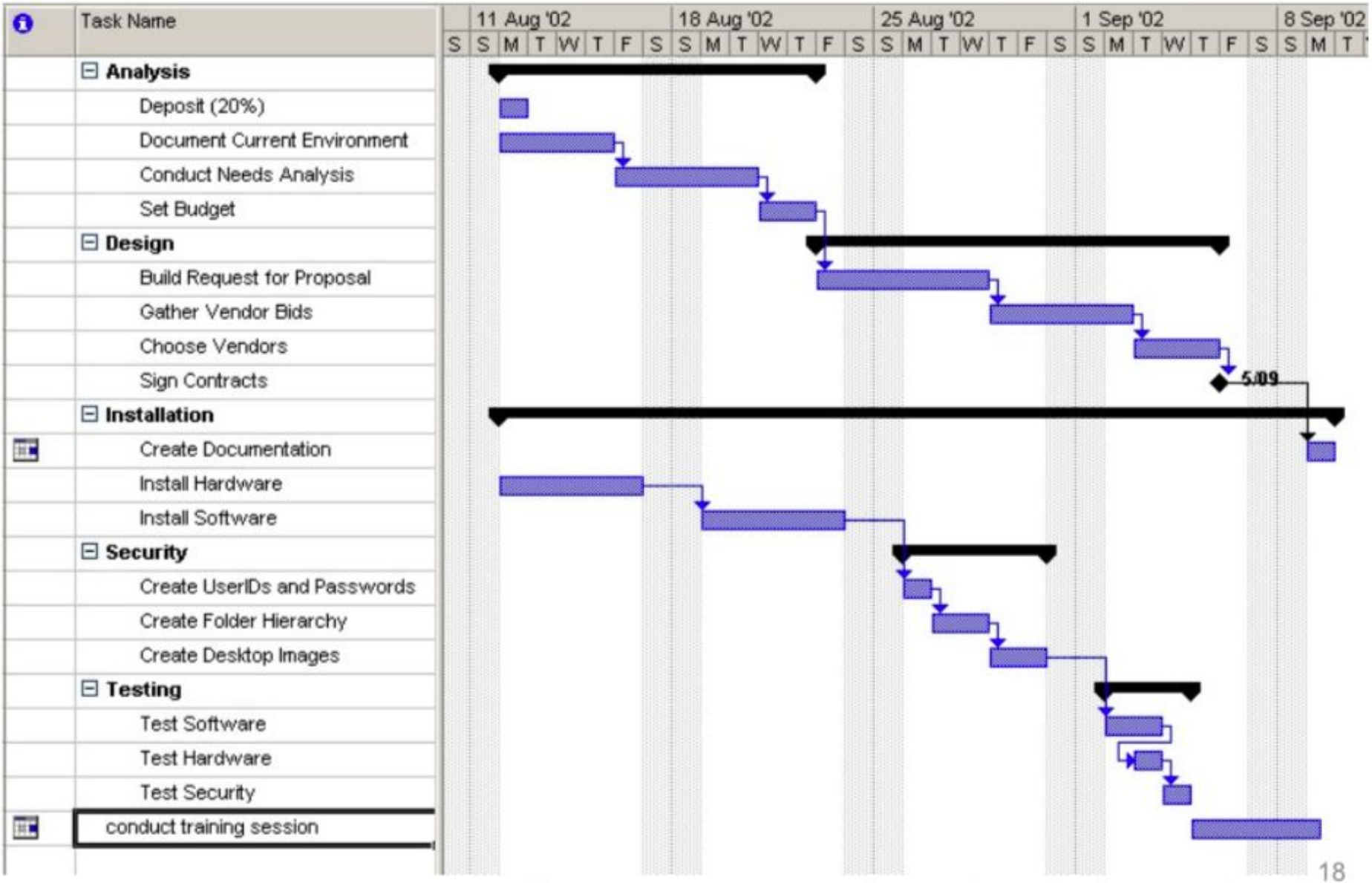
HDQL3-F: Các kỹ thuật lập lịch trình

1. **Gantt**
2. **PERT** (Program Evaluation and Review Technique)
3. **CPM** (Critical Path Method)

Biểu đồ Gantt

- Do Henry Gantt phát minh để thể hiện quá trình thực hiện tiến độ cho các công việc
  - Chỉ ra tên các tác vụ, người thực hiện, thời gian bắt đầu và kết thúc.
- Các công việc được biểu diễn theo trình tự thời gian với trục thời gian được trình bày theo trục hoành
  - Độ dài của đoạn thẳng là độ dài của công việc
  - Vị trí giữa các đoạn thẳng biểu diễn mối quan hệ trước sau giữa các công việc.


4/7
11/7
18/7
25/7
1/8
8/8
15/8
22/8
29/8
5/9
12/9
19/9
Start
T4
T1
T2
M1
T7
T3
M5
T8
M3
M2
T6
T5
M4
T9
M7
T10
M6
T11
M8
T12
Finish



18

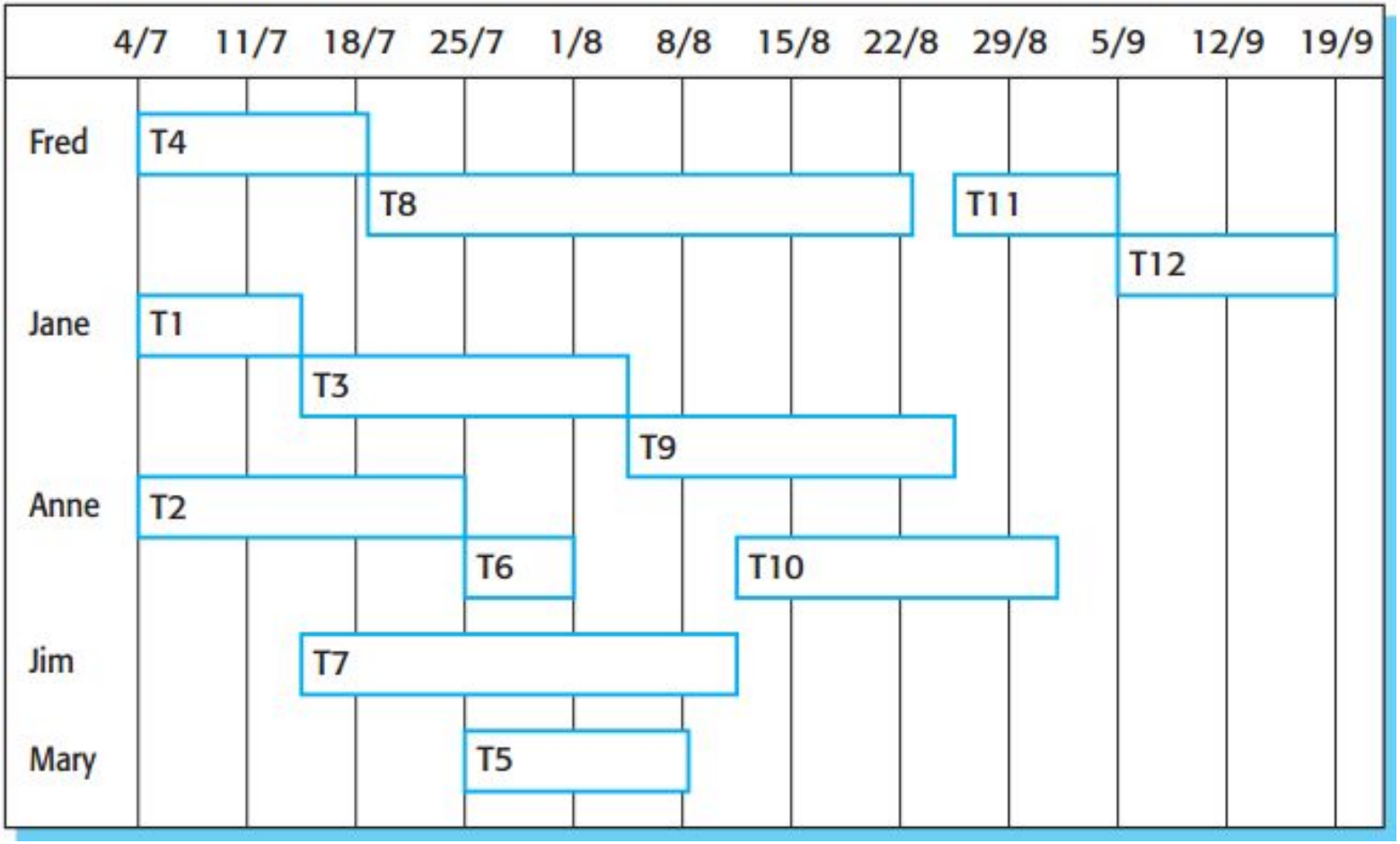
http://toolboxes.flexiblelearning.net.au/demosites/series9/906/map_respak/images/gantteg1.gif

Biểu đồ Gantt: Các công cụ hỗ trợ

- Microsoft Excel, Microsoft Project, …
- creately.com, …

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	Week of…	14-Jul	21-Jul	28-Jul	4-Aug	11-Aug	18-Aug	25-Aug	1-Sep	8-Sep	15-Sep	22-Sep	29-Sep	6-Oct	13-Oct
2	**Research Design Phase**														
3	Kickoff														
4	Define Scope														
5	Revise														
6															
7	**Data Collection Phase**														
8	Schedule Interviews														
9	Conduct Interviews														
10	Survey														
11	Analysis														
12															
13	**Final Report Creation**														
14	Writing														
15	Copy editing														
16	Layout														
17	Report Launch														
18															

Biểu đồ phân bổ nhân viên

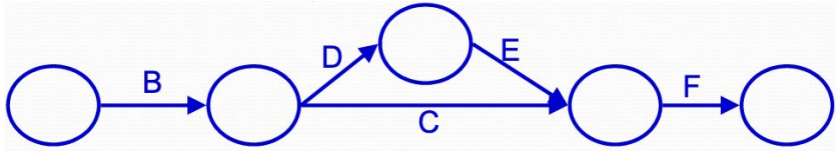


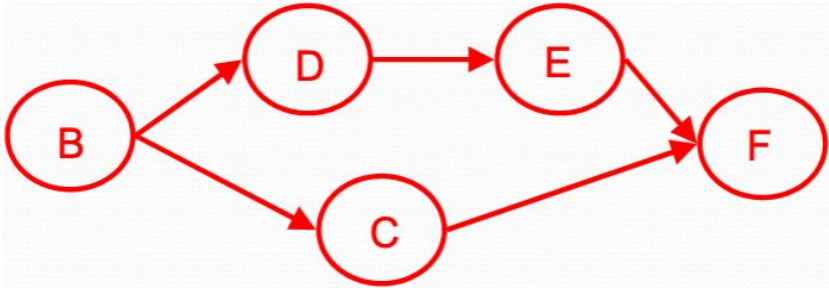
PERT:
Mạng hoạt động (1)

- Chỉ ra sự lệ thuộc giữa các tác vụ khác nhau tạo thành dự án:
  - **Công việc**: các việc cần làm
  - **Sự kiện**: kết quả công việc
- Mối liên hệ giữa các công việc (CV)
  - Có CV trước không có CV sau
  - Có CV sau không có CV trước
  - Có cả CV trước và sau

PERT: Mạng hoạt động (2)

- Có 2 kiểu biểu diễn:
  - **AoA** (Activity on Arrow):
    - Các mũi tên chỉ các công việc
    - Các nút chỉ các sự kiện
  - **AoN** (Activity on Node):
    - Các công việc được biểu diễn trên các nút





Vẽ mạng hoạt động AoN và AoA tương ứng

Công việc	Công việc trước
A	-
B	-
C	A
D	B
E	B
F	C,D

???

Công việc găng & Đường găng

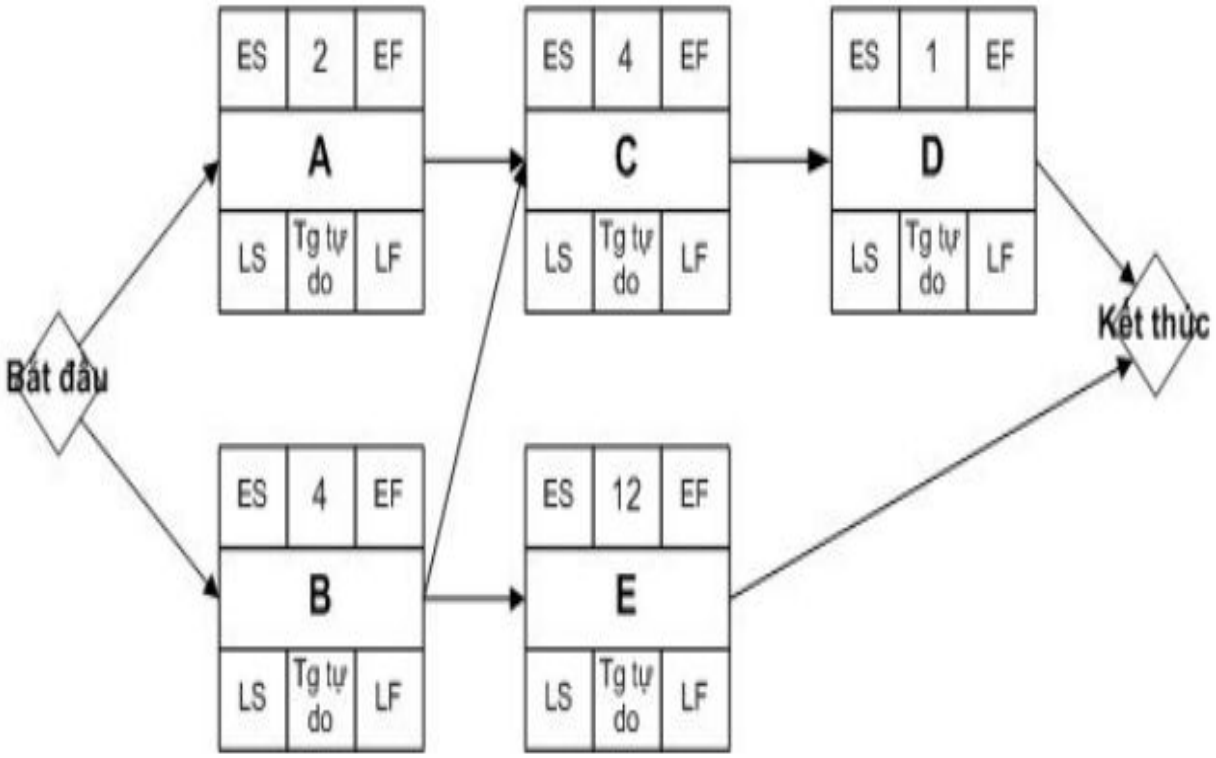
- **Công việc găng** (critical task) là các công việc có trữ lượng thời gian (thời gian tự do) bằng 0
- **Đường găng** (critical path) là *đường dài nhất* đi xuyên mạng hoạt động, từ nút bắt đầu tới nút kết thúc, và đi qua các công việc găng
- **Ý nghĩa:**
  - Độ dài của đường găng trên trục thời gian chính là **thời gian ít nhất** mà dự án có thể hoàn thành theo kế hoạch

Phương pháp Đường găng CPM (1)

- **ES** (Early Start): Thời gian bắt đầu sớm nhất
- **EF** (Early Finish): Thời gian kết thúc sớm nhất
- **LS** (Late Start): Thời gian bắt đầu muộn nhất
- **LF** (Late Finish): Thời gian kết thúc muộn nhất
- **Tg**: Thời gian hoàn thành
- **ES** của các công việc ngay sau khi bắt đầu quy định là 1
- **ES = max (EF của các công việc trước) + 1**
- **EF = ES + Tg – 1**
- **LF** của các công việc ngay trước khi kết thúc = **max (EF của các công việc này)**
- **LF = min (LS của các công việc sau) – 1**
- **LS = LF + 1 - Tg**

Phương pháp Đường găng CPM (2)

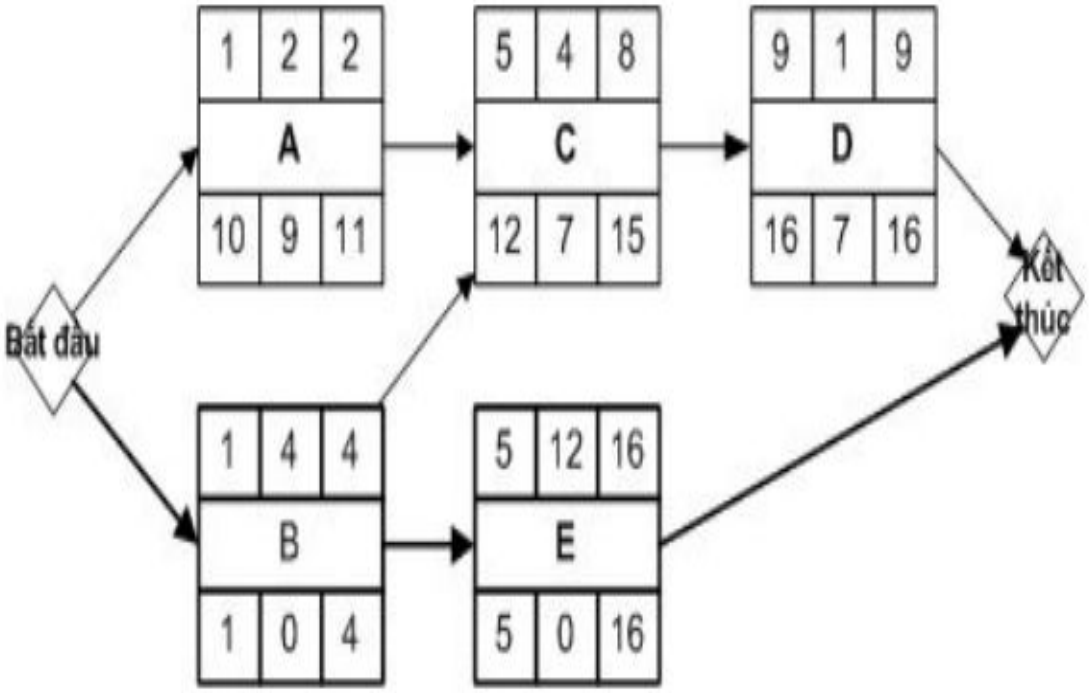
Công việc	Thời gian (ngày)	Công viêc trước
A	2	
B	4	
C	4	A, B
D	1	C
E	12	B



Mạng AoN tương ứng

Phương pháp Đường găng CPM (3)

Công việc	Thời gian (ngày)	Công viêc trước
A	2	
B	4	
C	4	A, B
D	1	C
E	12	B



Mạng AoN tương ứng

Đường găng là đường chứa toàn các công việc có thời gian tự do là 0. Trong sơ đồ trên, đường **B → E** là đường găng.

Bài tập: Tìm đường găng của dự án sau

Bài tập nhóm: Lập lịch trình dự án

1. Hình dung các yêu cầu cho dự án của nhóm
2. Nhận diện tác vụ cần thiết để hoàn thành dự án
3. Nhận diện sự phụ thuộc giữa các tác vụ
4. Ước lượng thời gian hoàn thành cho từng tác vụ
5. Phân phối các thành viên vào từng tác vụ
6. Vẽ các biểu đồ (Gantt, AoN, AoA) minh hoạ lịch trình
7. Tìm đường găng và thời gian tối thiểu để hoàn thành dự án

Quản trị rủi ro

Risk management

Rủi ro là gì ?

- **Rủi ro** là sự kiện xảy ra gây bất lợi đến quá trình phát triển dự án

Một số rủi ro phần mềm (1)

Rủi ro	Miêu tả
Nhân sự biến động	Nhân viên có kinh nghiệm rời khỏi dự án trước khi nó hoàn thành
Sự thay đổi trong cách quản lý	Có một sự thay đổi trong cách quản lý công ty với những ưu tiên khác
Phần cứng không có sẵn	Phần cứng cần thiết cho dự án không được phân phối theo như lịch trình
Yêu cầu thay đổi	Có một số lượng lớn sự thay đổi trong yêu cầu hơn dự kiến
Đặc tả bị trễ	Đặc tả của những giao diện cần thiết không sẵn có trên lịch trình

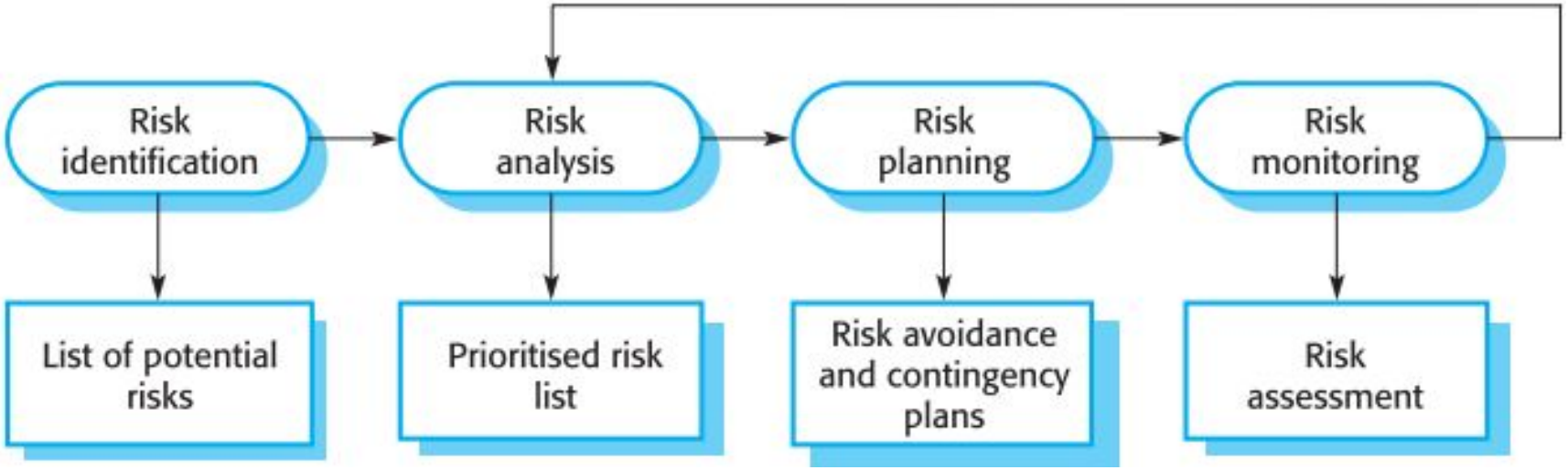
Một số rủi ro phần mềm (2)

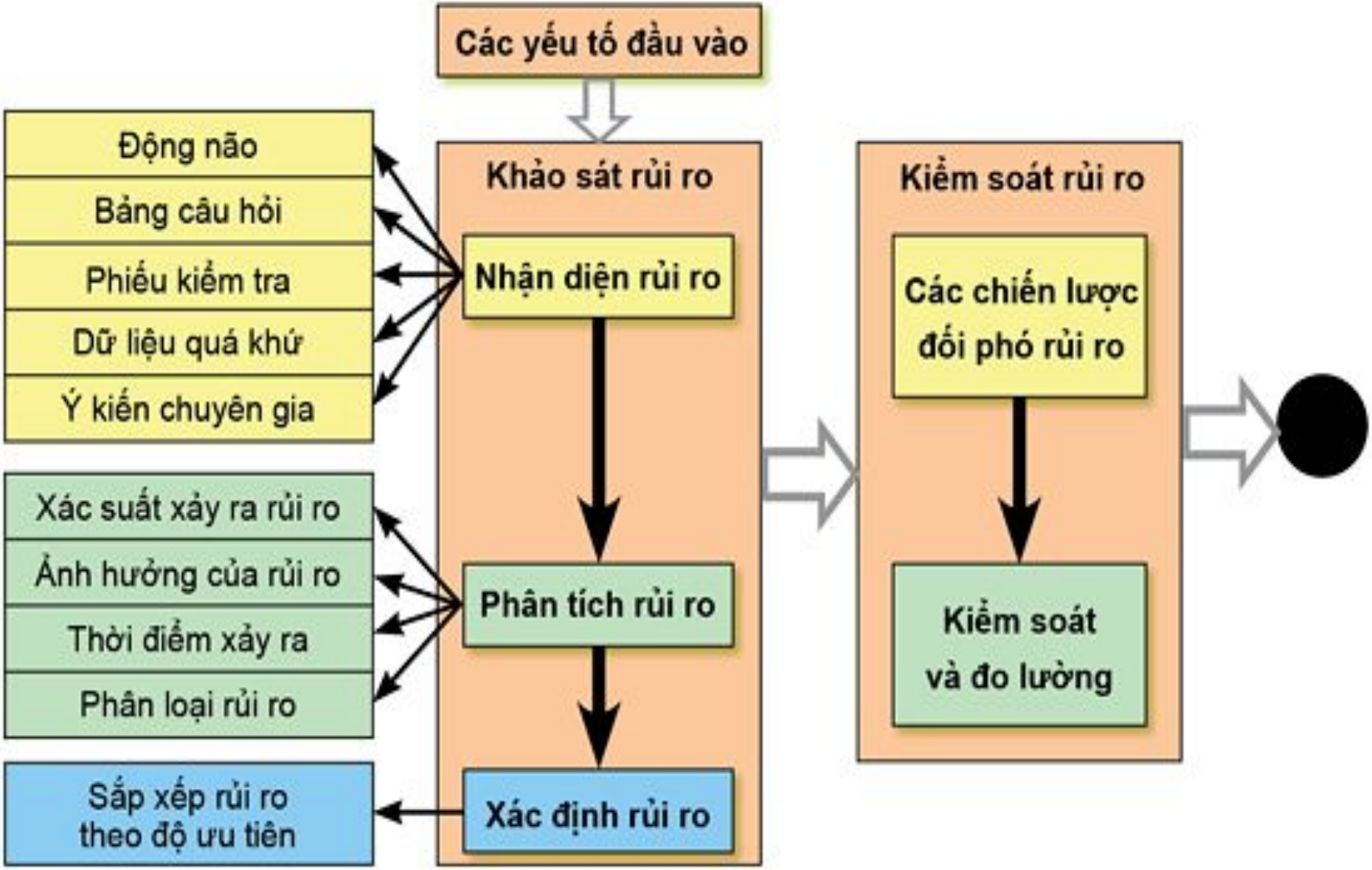
Rủi ro	Miêu tả
Ước tính thấp kích thước	Kích thước của hệ thống đã bị ước tính thấp
Công cụ CASE kém hiệu quả	Những công cụ CASE hỗ trợ dự án không thực thi như dự kiến
Công nghệ thay đổi	Công nghệ nền tảng mà hệ thống được xây dựng trên đó bị thay thế bởi công nghệ mới
Sự cạnh tranh sản phẩm	Một sản phẩm cạnh tranh đã được đưa ra thị trường trước khi sản phẩm hoàn thiện

Quản trị rủi ro

- Liên quan đến ***xác định những rủi ro*** và ***thiết lập những kế hoạch*** để tối giản hóa ảnh hưởng của chúng đối với dự án

Quy trình Quản trị rủi ro




Các yếu tố đầu vào
Động não
Bảng câu hỏi
Phiếu kiểm tra
Dữ liệu quá khứ
Ý kiến chuyên gia
Khảo sát rủi ro
Nhận diện rủi ro
Xác suất xảy ra rủi ro
Ảnh hưởng của rủi ro
Thời điểm xảy ra
Phân loại rủi ro
Phân tích rủi ro
Sắp xếp rủi ro
theo độ ưu tiên
Xác định rủi ro
Kiểm soát rủi ro
Các chiến lược
đối phó rủi ro
Kiểm soát
và đo lường

HĐ1: Xác định rủi ro

- Rủi ro công nghệ
- Rủi ro con người
- Rủi ro tổ chức công ty
- Rủi ro yêu cầu
- Rủi ro ước lượng

Một số loại rủi ro (1)

Loại rủi ro	Những rủi ro có thể
Công nghệ	• Cơ sở dữ liệu được sử dụng trong hệ thống không thể xử lý nhiều giao dịch (transactions) trong một giây như mong đợi • Những thành phần hệ thống nên được sử dụng chứa những khiếm khuyết dẫn đến hạn chế chức năng của chúng
Con người	• Không thể tuyển dụng nhân sự có những kỹ năng được yêu cầu • Nhân sự chủ chốt ốm và không sẵn sàng trong những thời điểm quan trọng • Khóa huấn luyện yêu cầu cho nhân sự không sẵn có
Tổ chức công ty	• Công ty phải tái cấu trúc và quản lý khác chịu trách nhiệm dự án • Những vấn đề tài chính công ty buộc ngân sách dự án giảm

Một số loại rủi ro (2)

Loại rủi ro	Những rủi ro có thể
Công cụ	• Mã được sinh ra bởi những công cụ CASE không hiệu quả • Công cụ CASE không thể được tích hợp
Yêu cầu	• Thay đổi yêu cầu dẫn đến việc thiết kế lại phần lớn được đề xuất • Khách hàng không hiểu được ảnh hưởng của việc yêu cầu thay đổi
Ước lượng	• Thời gian yêu cầu để phát triển phần mềm bị ước lượng quá thấp • Tỷ lệ khuyết khiếm phải sửa bị ước lượng quá thấp • Kích thước của phần mềm bị ước lượng quá thấp

HĐ2: Phân tích rủi ro (1)

- Đánh giá ***xác suất*** và ***mức độ tác động*** của từng rủi ro
- Xác suất có thể là: *(1) rất cao, (2) cao, (3) trung bình, (4) thấp,* hoặc *(5) rất thấp*
- Mức độ tác động có thể là: *(1) thảm khốc, (2) nghiêm trọng, (3) chấp nhận được* hoặc *(4) không đáng kể*

HĐ2: Phân tích rủi ro (2)

Rủi ro (Risk)	Xác suất (Probability)	Tác động (Effects)
Những vấn đề tài chính công ty buộc ngân sách dự án giảm	Thấp (4)	Thảm khốc (1)
Không thể tuyển dụng nhân sự có những kỹ năng được yêu cầu	Cao (2)	Thảm khốc (1)
Nhân sự chủ chốt ốm và không sẵn sàng trong những thời điểm quan trọng	Trung bình (3)	Nghiêm trọng (2)
Những thành phần hệ thống nên được sử dụng chứa những khiếm khuyết dẫn đến hạn chế chức năng của chúng	Trung bình (3)	Nghiêm trọng (2)
Thay đổi yêu cầu dẫn đến việc thiết kế lại phần lớn được đề xuất	Trung bình (3)	Nghiêm trọng (2)

HĐ2: Phân tích rủi ro (3)

Rủi ro (Risk)	Xác suất (Probability)	Tác động (Effects)
Công ty phải tái cấu trúc và quản lý khác chịu trách nhiệm dự án	Cao (2)	Nghiêm trọng (2)
Cơ sở dữ liệu được sử dụng trong hệ thống không thể xử lý nhiều giao dịch trong một giây như mong đợi	Trung bình (3)	Nghiêm trọng (2)
Thời gian yêu cầu để phát triển phần mềm bị ước lượng quá thấp	Cao (2)	Nghiêm trọng (2)
Công cụ CASE không thể được tích hợp	Cao (2)	Chấp nhận được (3)
Khách hàng không hiểu được ảnh hưởng của việc yêu cầu thay đổi	Trung bình (3)	Chấp nhận được (3)

HĐ2: Phân tích rủi ro (4)

Rủi ro (Risk)	Xác suất (Probability)	Tác động (Effects)
Khóa huấn luyện yêu cầu cho nhân sự không sẵn có	Trung bình (3)	Chấp nhận được (3)
Tỷ lệ khuyết khiếm phải sửa bị ước lượng quá thấp	Trung bình (3)	Chấp nhận được (3)
Kích thước của phần mềm bị ước lượng quá thấp	Cao (2)	Chấp nhận được (3)
Mã được sinh ra bởi những công cụ CASE không hiệu quả	Trung bình (3)	Không đáng kể (4)

Mức độ

"Cây" rủi ro

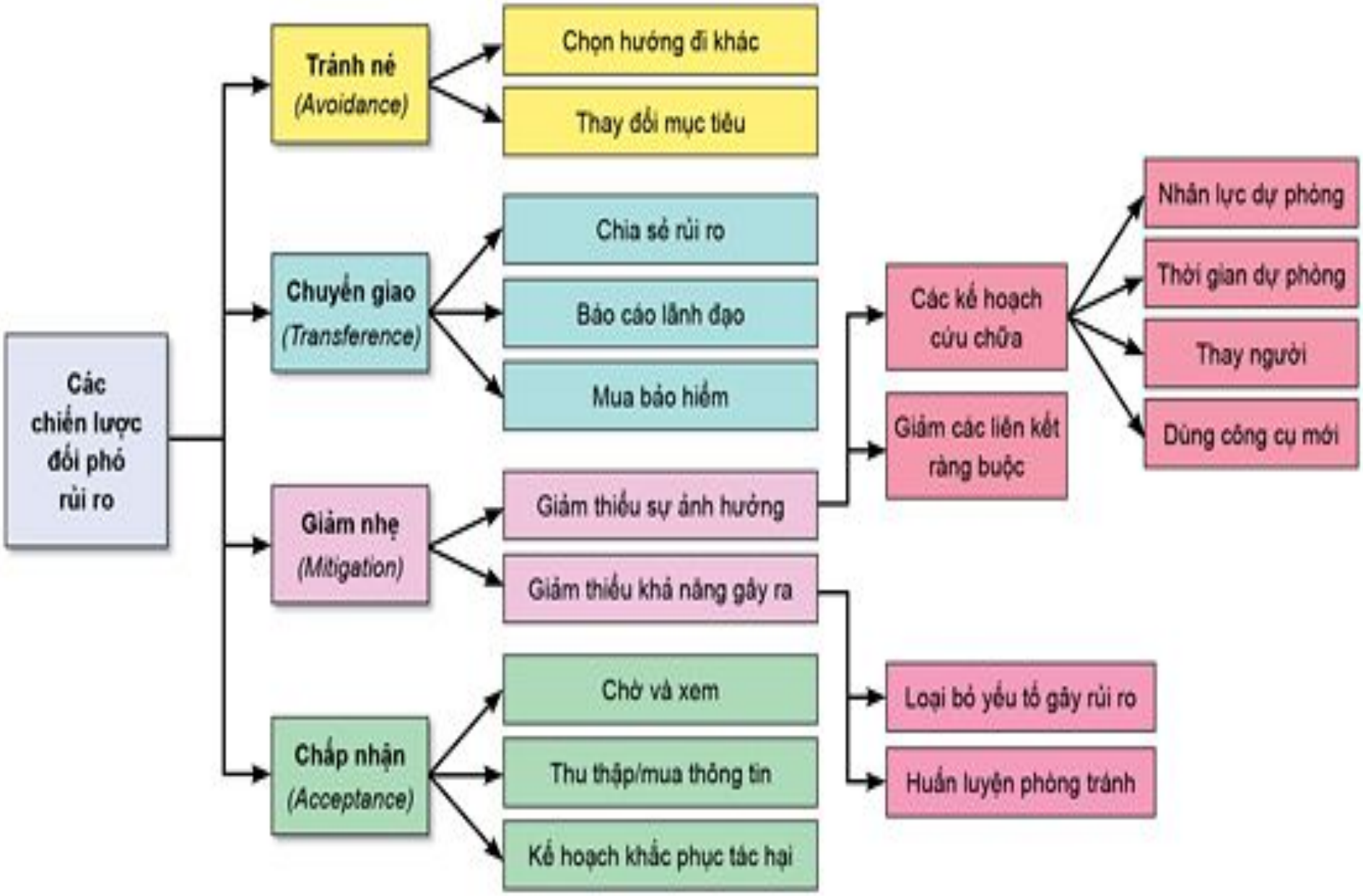
Mức độ THẤP:
. Số lượng rủi ro nhiều
. Độ phức tạp thấp
. Mức ảnh hưởng không cao

Mức độ TRUNG BÌNH:
. Số lượng rủi ro khá nhiều
. Độ phức tạp đa dạng
. Mức ảnh hưởng trung bình

Mức độ CAO:
. Số lượng rủi ro ít
. Độ phức tạp cao
. Mức ảnh hưởng lớn

HĐ3: Lập kế hoạch rủi ro

- **Chiến thuật Tránh**
  - Làm giảm xác suất phát sinh rủi ro
- **Chiến thuật Tối giản**
  - Làm giảm ảnh hưởng của rủi ro
- **Kế hoạch Dự phòng**
  - Tạo ra những kế hoạch dự phòng


Các chiến lược đối phó rủi ro
Tránh né (Avoidance)
Chọn hướng đi khác
Thay đổi mục tiêu
Chuyển giao (Transference)
Chia sẻ rủi ro
Báo cáo lãnh đạo
Mua bảo hiểm
Giảm nhẹ (Mitigation)
Giảm thiểu sự ảnh hưởng
Giảm thiểu khả năng gây ra
Các kế hoạch cứu chữa
Giảm các liên kết ràng buộc
Nhân lực dự phòng
Thời gian dự phòng
Thay người
Dùng công cụ mới
Loại bỏ yếu tố gây rủi ro
Huấn luyện phòng tránh
Chấp nhận (Acceptance)
Chờ và xem
Thu thập/mua thông tin
Kế hoạch khắc phục tác hại

Chiến thuật Quản trị rủi ro

Rủi ro	Chiến thuật
Công ty cắt giảm chi phí	Chuẩn bị một tài liệu tóm lược cho quản lý cấp cao để chỉ ra rằng dự án đang có đóng góp rất quan trọng như thế nào cho các mục tiêu kinh doanh
Nhân viên ốm	Tổ chức lại nhóm sao cho có thêm nhiều chồng lặp trong công việc và con người do đó các thành viên hiểu được công việc của nhau
Thành phần kiếm khuyết	Thay thế những thành phần khiếm khuyết tiềm ẩn bằng cách mua vào những thành phần có độ tin cậy cao

HĐ4: Giám sát rủi ro

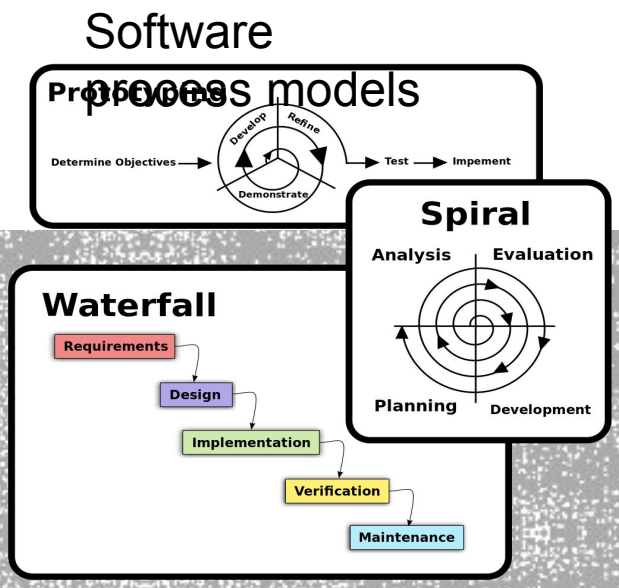
- Đánh giá thường xuyên từng rủi ro để quyết định liệu nó đang ít xảy ra hơn không hay ngược lại
- Đánh giá liệu tác động của rủi ro có thay đổi không

Bài tập: Quản trị rủi ro

1. Đề xuất một dự án phần mềm
2. Nhận diện những rủi ro tác động tới dự án
3. Phân tích xác suất xảy ra và mức độ ảnh hưởng của từng rủi ro
4. Đưa ra các biện pháp đối phó với từng rủi ro

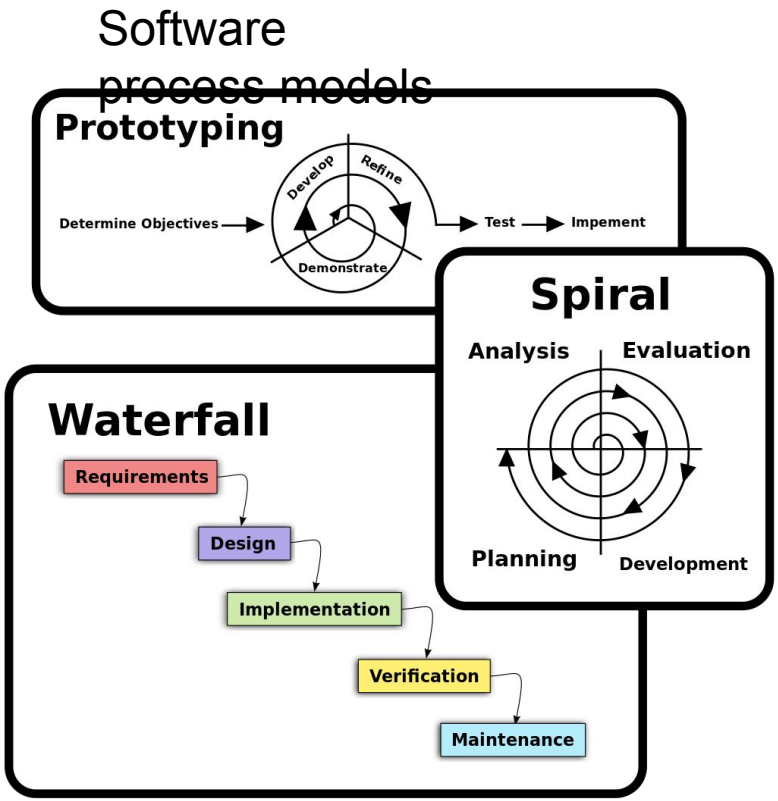
4. MÔ HÌNH QUY TRÌNH PHẦN MỀM

Software Processes



Quy trình phần mềm ?

- Một tập các hành động có cấu trúc, với mục đích phát triển (hoặc tiến hóa của phần mềm)



Mô hình quy trình phần mềm

- Là ***một biểu diễn trừu tượng*** được ***đơn giản hóa*** của một Quy trình phần mềm (QTPM)
- Mô hình QTPM chỉ miêu tả về một vài khía cạnh cụ thể của một quy trình phần mềm:
  - *Khía cạnh luồng công việc*
  - *Khía cạnh luồng dữ liệu*
  - *Khía cạnh vai trò*

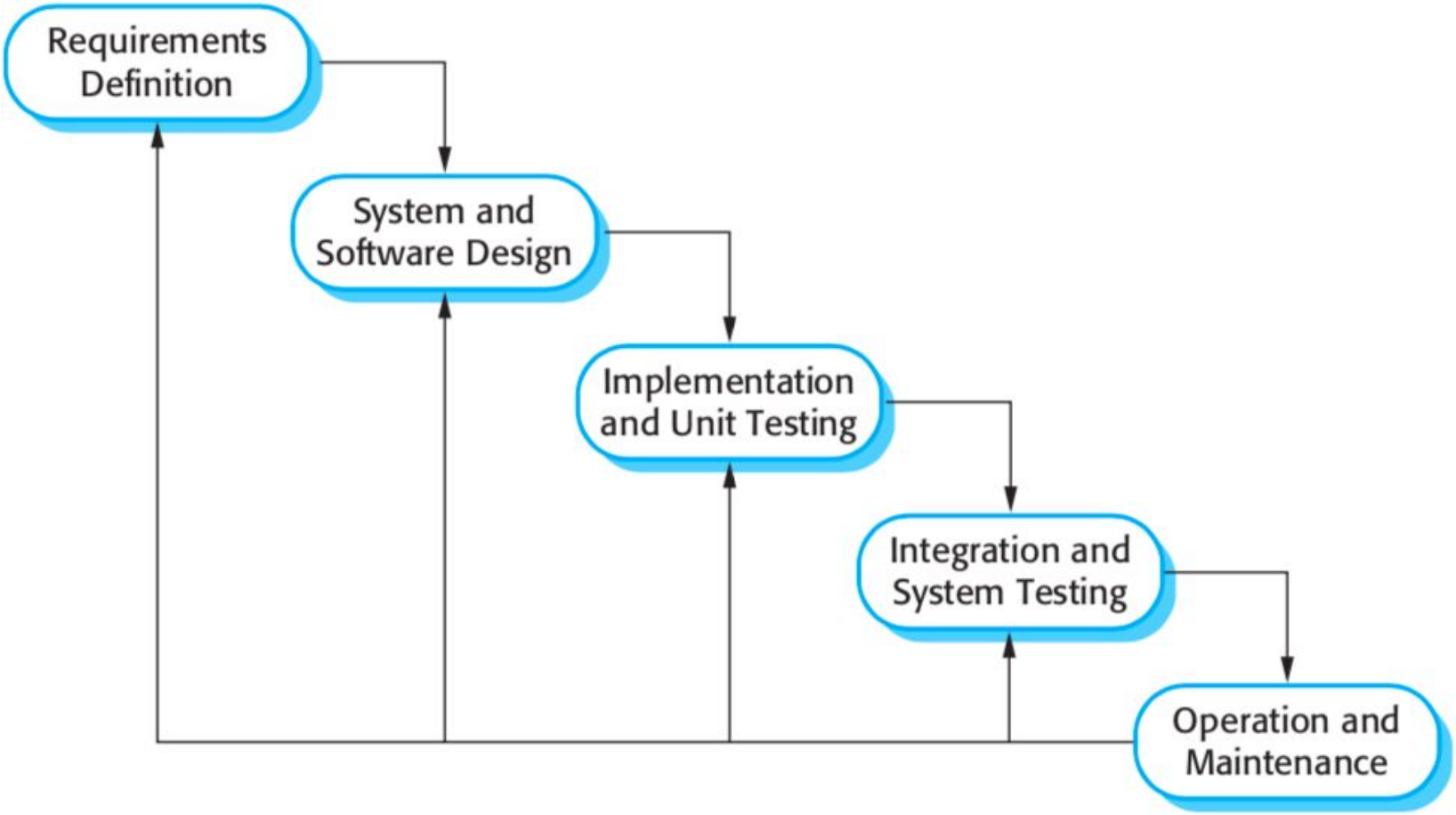
Các mô hình quy trình phần mềm

1. **Mô hình hướng kế hoạch (Plan-driven Models)**
   - Mô hình thác nước
   - Mô hình chữ V
   - Mô hình dựa trên tái sử dụng (dựa trên thành phần)
2. **Mô hình phát triển lặp (& gia tăng)**
   - Mô hình phát triển tiến hóa
   - Mô hình phân phối gia tăng
   - Mô hình xoắn ốc
3. **Mô hình phát triển linh hoạt (Agile Models)**
   - Mô hình SCRUM

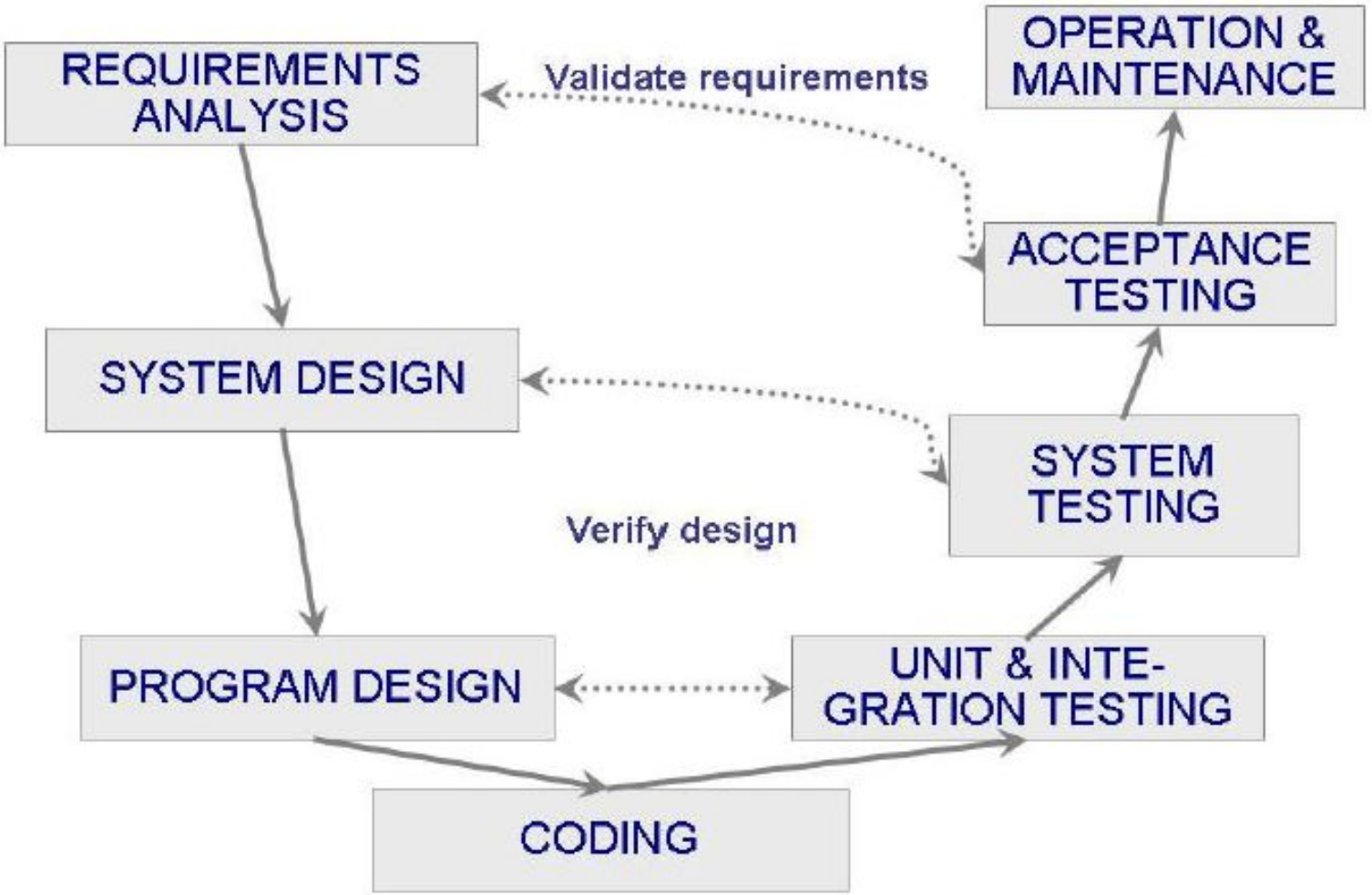
4.1. Các mô hình hướng kế hoạch

- Còn gọi là hướng tài liệu (document driven)
- Dễ hiểu và dễ áp dụng
- Thiếu tính linh hoạt
- Khó khăn và tốn kém khi yêu cầu thay đổi
- Phù hợp với những dự án lớn

Mô hình thác nước

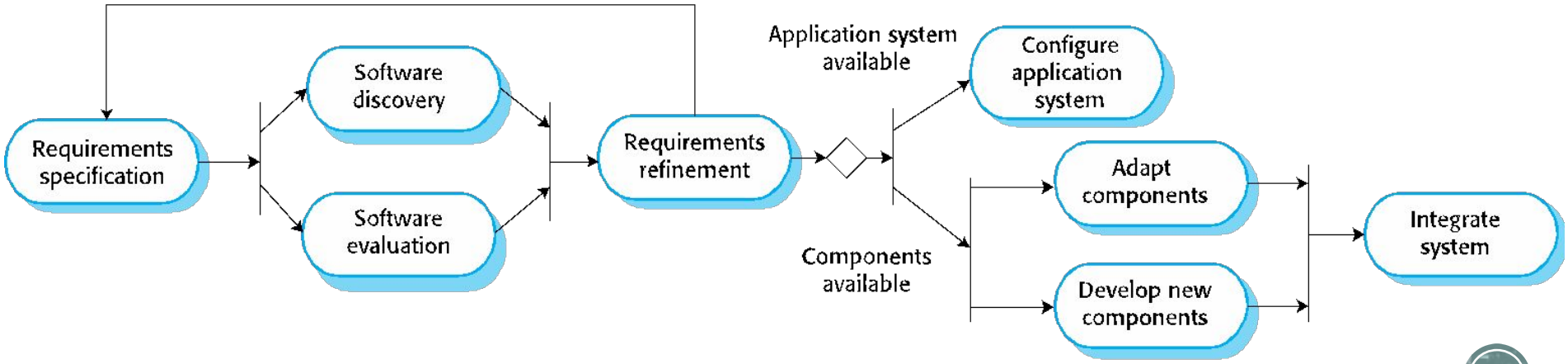


Mô hình chữ V


REQUIREMENTS ANALYSIS
Validate requirements
OPERATION & MAINTENANCE
ACCEPTANCE TESTING
SYSTEM DESIGN
SYSTEM TESTING
Verify design
PROGRAM DESIGN
UNIT & INTE-GRATION TESTING
CODING

Mô hình dựa trên thành phần

- Phát triển dựa trên việc tái sử dụng, tích hợp từ các thành phần có sẵn hoặc từ các thành phần thương mại COTS (Commercial-off-the-shelf)
- Đang ngày càng được sử dụng nhiều khi các tiêu chuẩn thành phần được đưa vào sử dụng

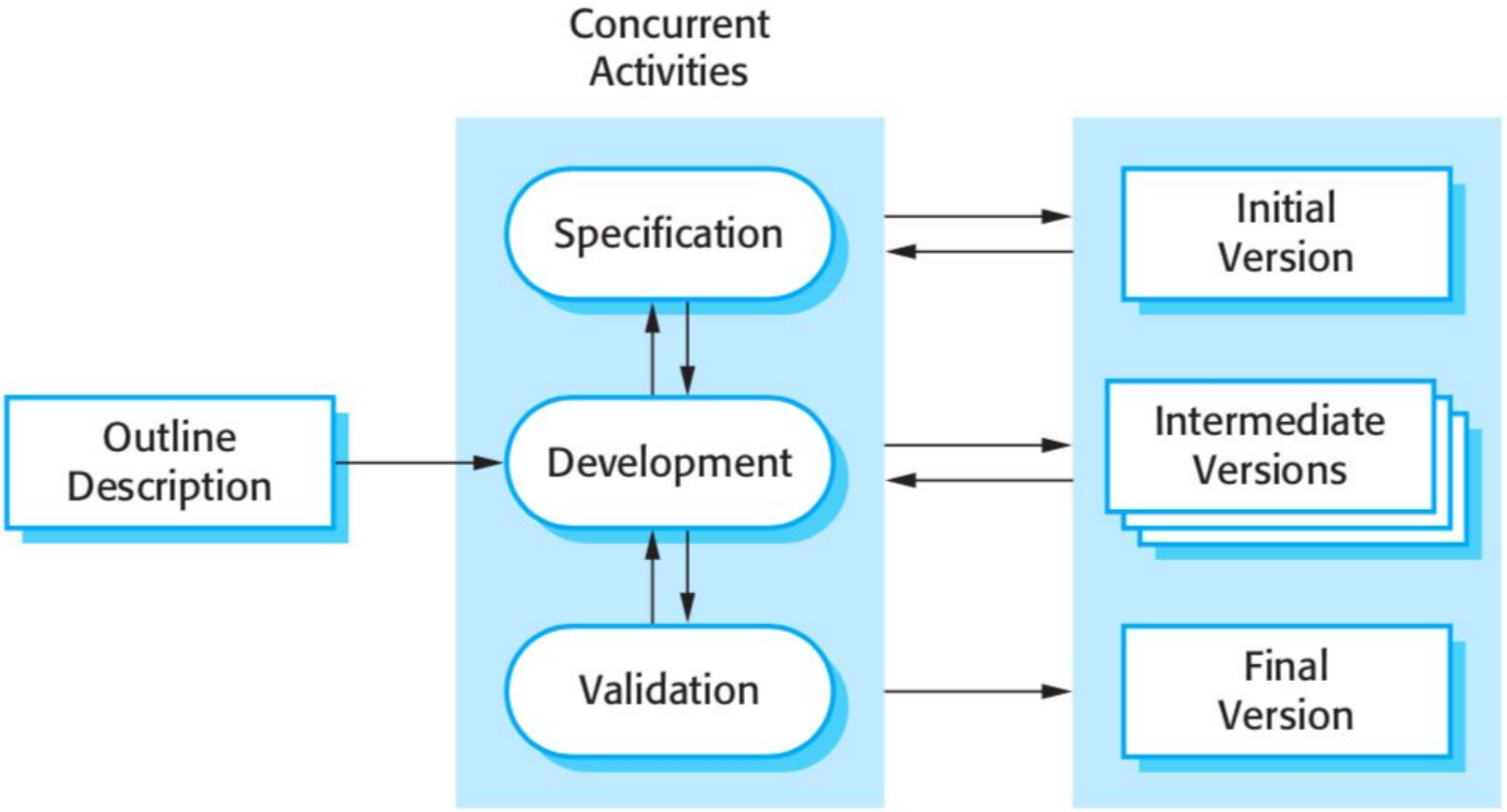


4.2. Mô hình lặp & gia tăng

- Các yêu cầu hệ thống luôn tiến hóa trong quá trình thực hiện dự án
- Do đó, việc lặp lại quy trình phát triển luôn là một phần trong quá trình phát triển những hệ thống lớn

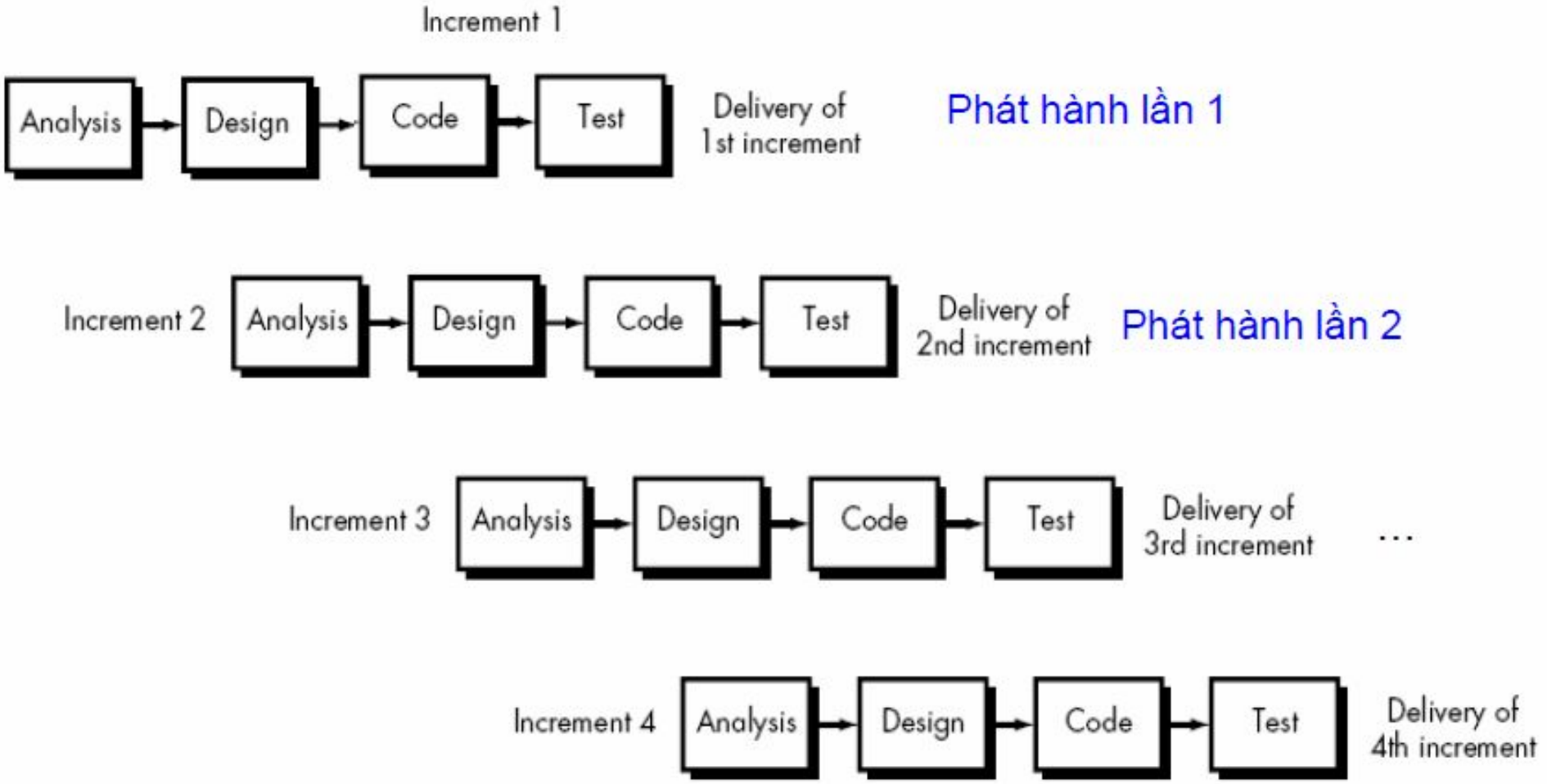
Mô hình phát triển tiến hóa

- Ý tưởng:
  1. Xây dựng một mẫu thử ban đầu và đưa cho người dùng xem xét
  2. Tinh chỉnh mẫu thử qua nhiều phiên bản cho đến khi thỏa mãn yêu cầu của người dùng thì dừng lại
- Các hoạt động *Đặc tả*, *Phát triển* và *Xác thực* được thực hiện đan xen


Concurrent Activities
Specification
Development
Validation
Outline Description
Initial Version
Intermediate Versions
Final Version

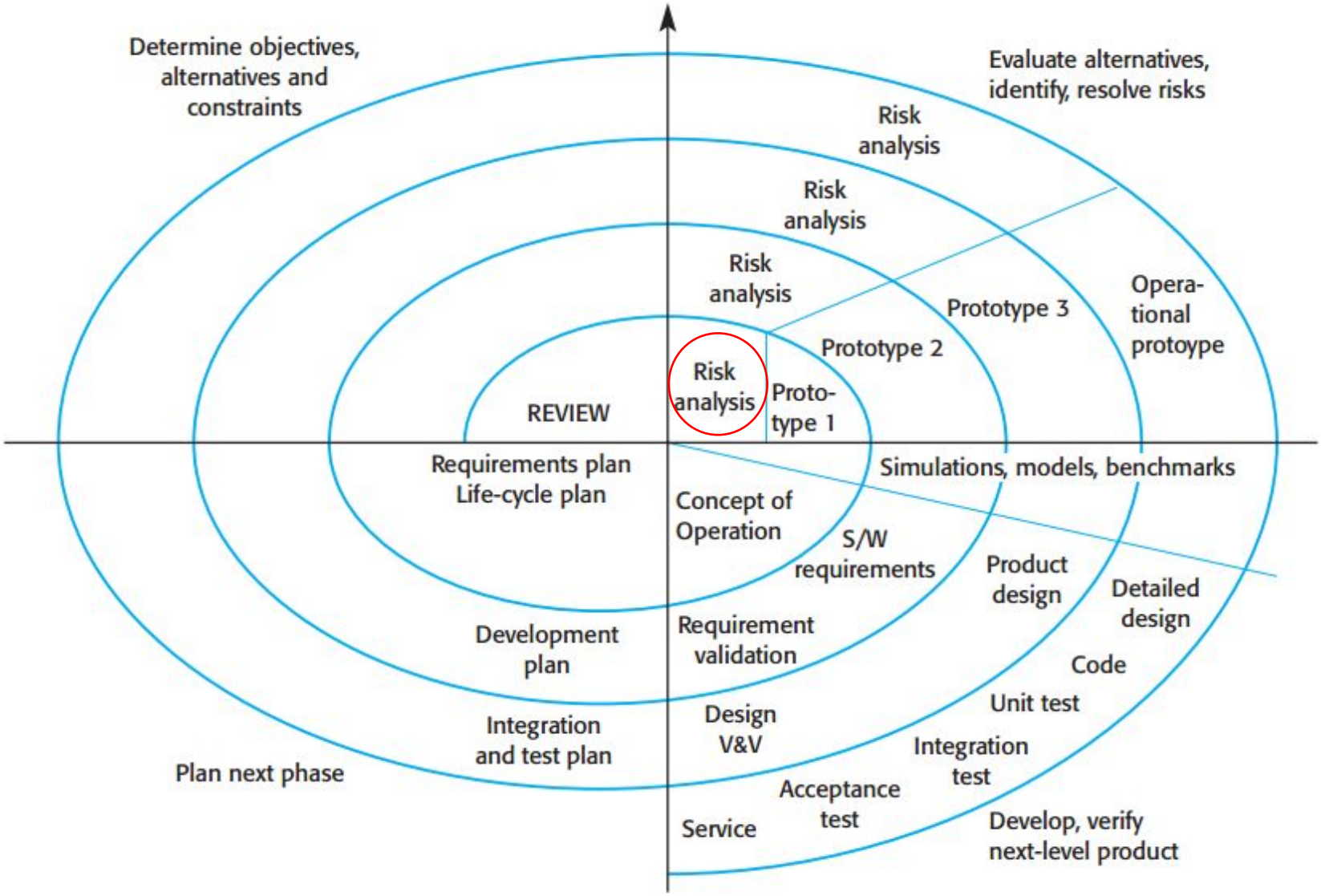
Mô hình phân phối gia tăng

- Sự phát triển và phân phối được chia ra thành nhiều vòng tăng dần, được gọi là *các gia số (increment)*
- Mỗi gia số phân phối một tập con các chức năng được yêu cầu
- Những yêu cầu người dùng được sắp thứ tự ưu tiên và yêu cầu ưu tiên cao nhất được bao gồm trong những gia số đầu tiên


Increment 1
Analysis
Design
Code
Test
Delivery of
1st increment
Phát hành lần 1
Increment 2
Analysis
Design
Code
Test
Delivery of
2nd increment
Phát hành lần 2
Increment 3
Analysis
Design
Code
Test
Delivery of
3rd increment
...
Increment 4
Analysis
Design
Code
Test
Delivery of
4th increment

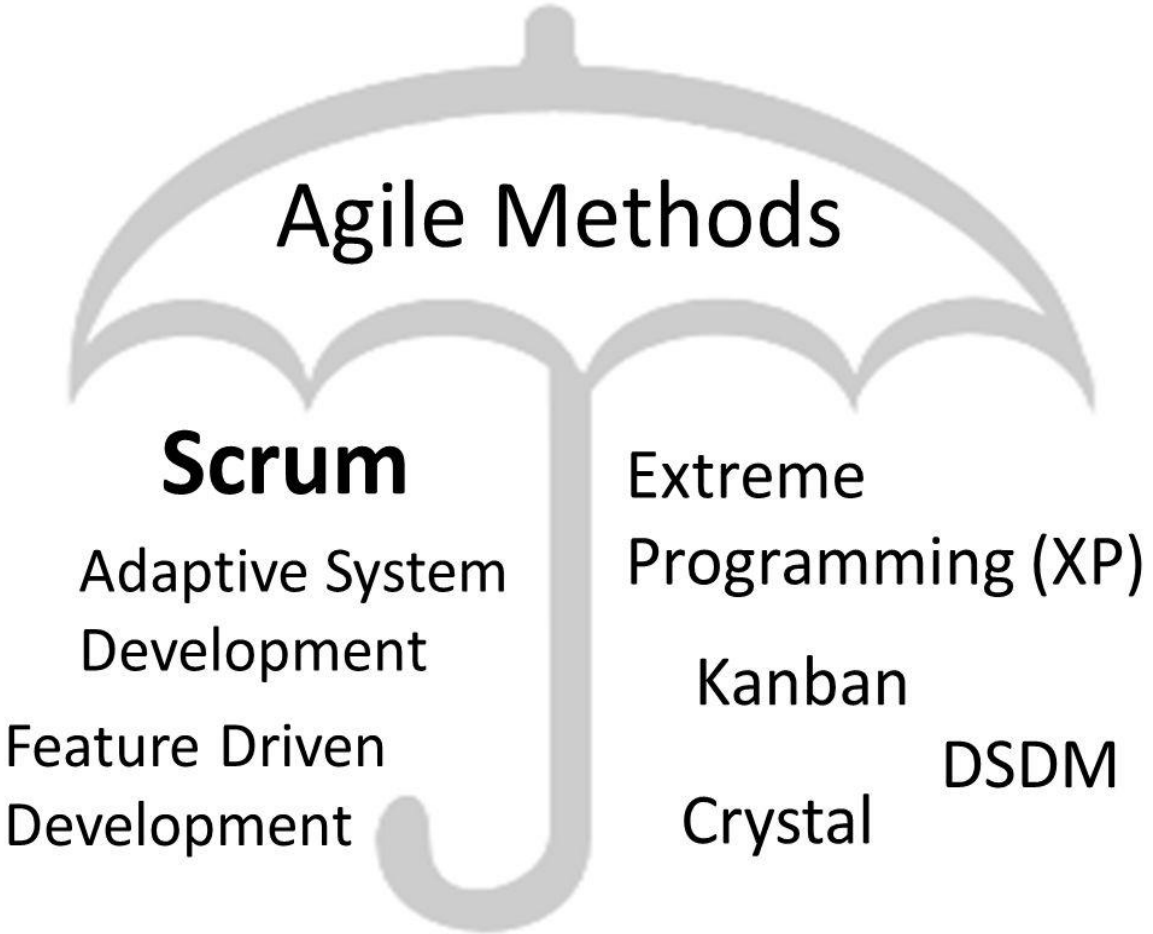
Mô hình xoắn ốc

- Quy trình phát triển được biểu diễn theo ***một hình xoắn ốc*** thay vì một chuỗi tuần tự những hành động với cơ chế truy vết ngược
- Mỗi vòng lặp trong sơ đồ xoắn ốc biểu diễn một pha của quá trình phát triển
- **<u>Rủi ro</u> được đánh giá và giải quyết trong suốt quá trình phát triển**

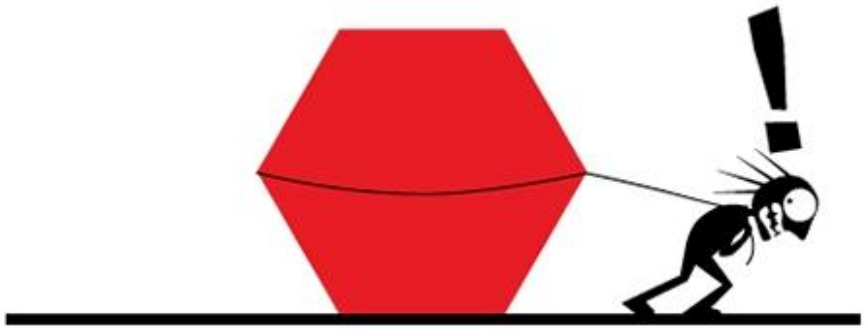


Boehm's spiral model, 1988

4.3. Mô hình phát triển linh hoạt

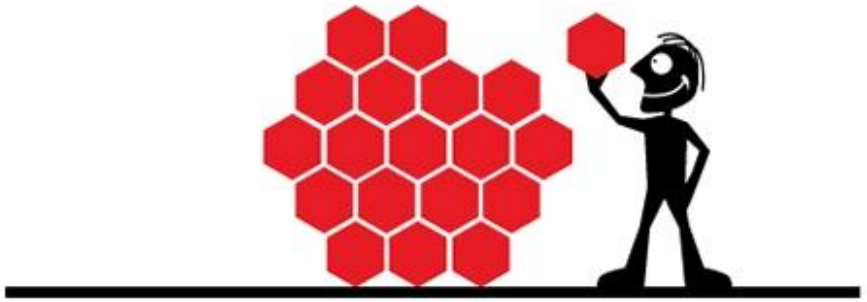


THE WATERFALL PROCESS



'This project has got so big, I'm not sure I'll be able to deliver it!'

THE AGILE PROCESS

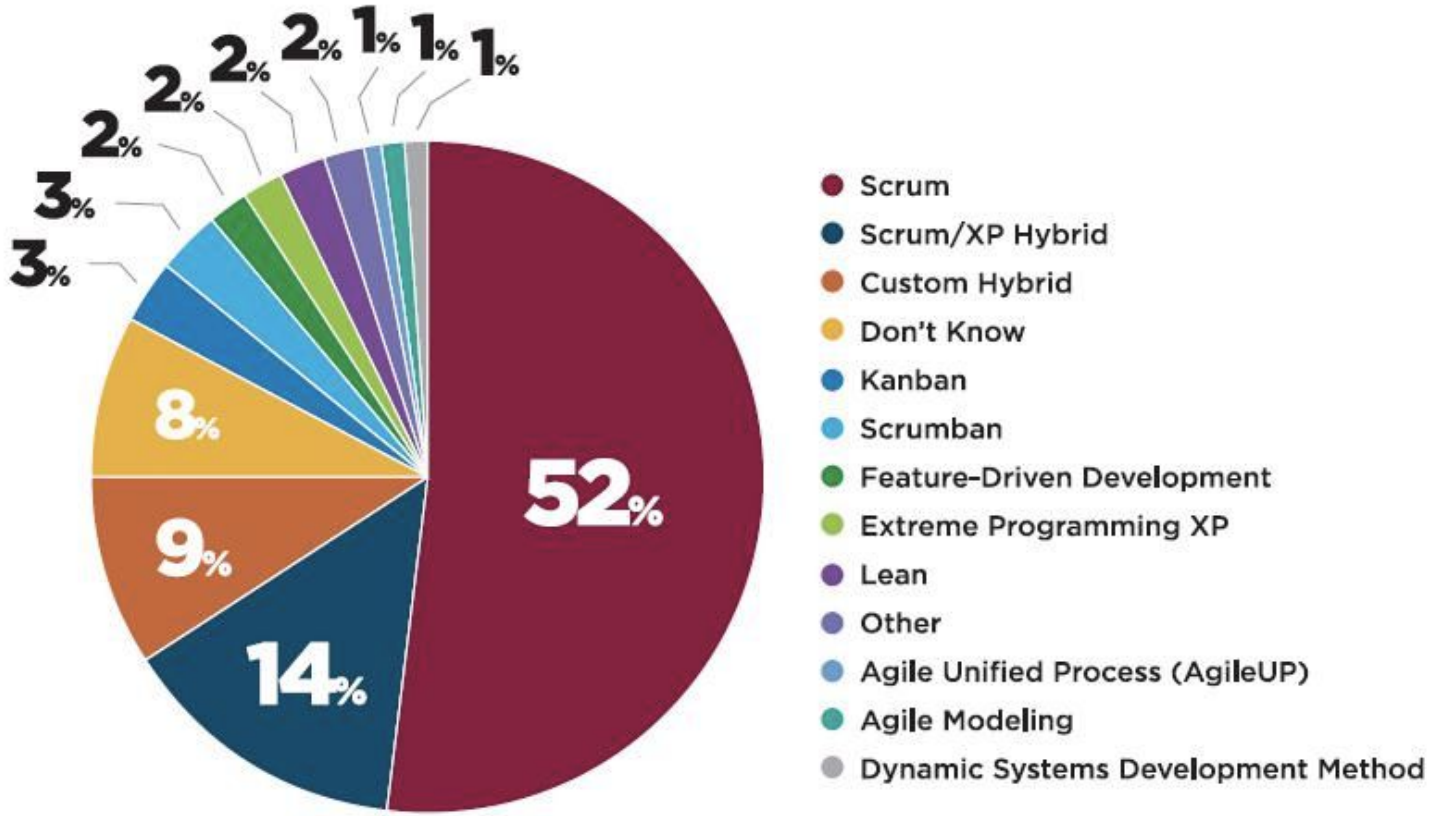


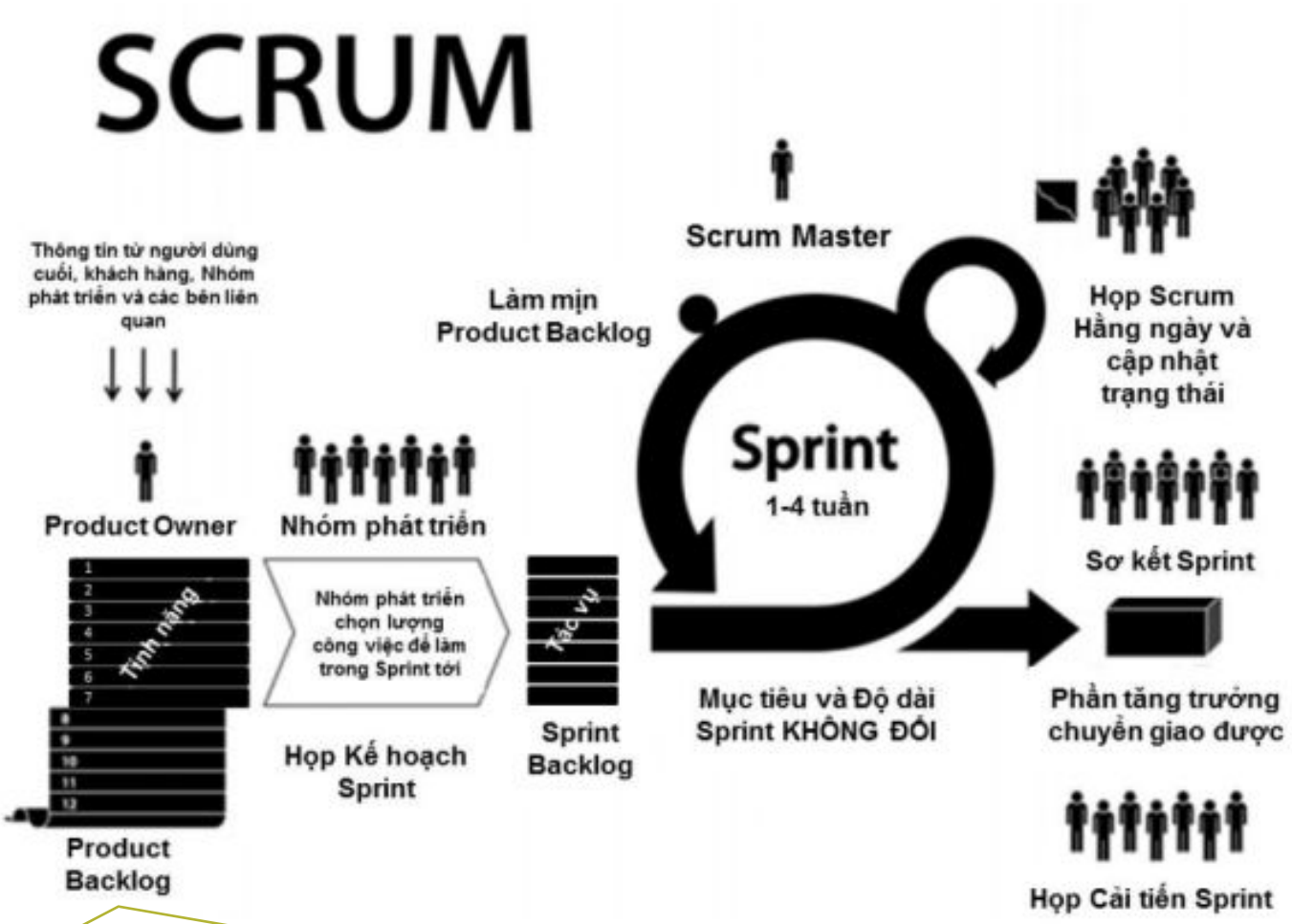
'It's so much better delivering this project in bite-sized sections'

Tuyên ngôn Agile

1. Cá nhân và sự tương tác hơn là quy trình và công cụ
2. Chương trình chạy tốt hơn là tài liệu đầy đủ
3. Cộng tác với khách hàng hơn là đàm phán hợp đồng
4. Phản hồi với sự thay đổi hơn là bám theo kế hoạch

Tỉ lệ các phương pháp Agile được dùng





User Story: Là *<người dùng cụ thể \ vai trò>* , tôi muốn *<làm gì đó>* để *<phục vụ mục đích nào đó>*

Câu chuyện người dùng (User story)

- **User Story là** một bản tóm tắt nhu cầu người dùng
- **Cú pháp: Là *<người dùng cụ thể \ vai trò>*, tôi muốn *<làm gì đó>* để *<phục vụ mục đích nào đó>***
  - Ví dụ: *Là quản trị của diễn đàn, tôi muốn xóa một người dùng phạm quy nghiêm trọng để tránh gây gại cho diễn đàn.*

CÔNG CỤ CASE

Computer-Aided Software Engineering

CASE là gì? (1)

- Những hệ thống phần mềm hỗ trợ cho quá trình phát triển và tiến hóa phần mềm
- Giúp tự động hóa các hoạt động
  - Trình soạn thảo bằng đồ họa cho quá trình phát triển mô hình hệ thống
  - Từ điển dữ liệu để quản lý những thực thể trong thiết kế
  - Trình tạo giao diện đồ họa cho việc xây dựng giao diện người dùng
  - Trình gỡ rối để hỗ trợ việc tìm lỗi chương trình
  - Trình dịch tự động để tạo những phiên bản mới cho chương trình

CASE là gì? (2)

- Upper-CASE:
  - Những công cụ hỗ trợ những hoạt động ở giai đoạn đầu: *xác định yêu cầu và thiết kế*
- Lower-CASE:
  - Những công cụ hỗ trợ những hoạt động ở giai đoạn sau như: *lập trình, debugging và testing*

Phân loại CASE

- **Khía cạnh chức năng**
  - Phân loại dựa theo chức năng cụ thể của chúng
- **Khía cạnh quy trình**
  - Phân loại dựa theo những hoạt động quy trình được hỗ trợ

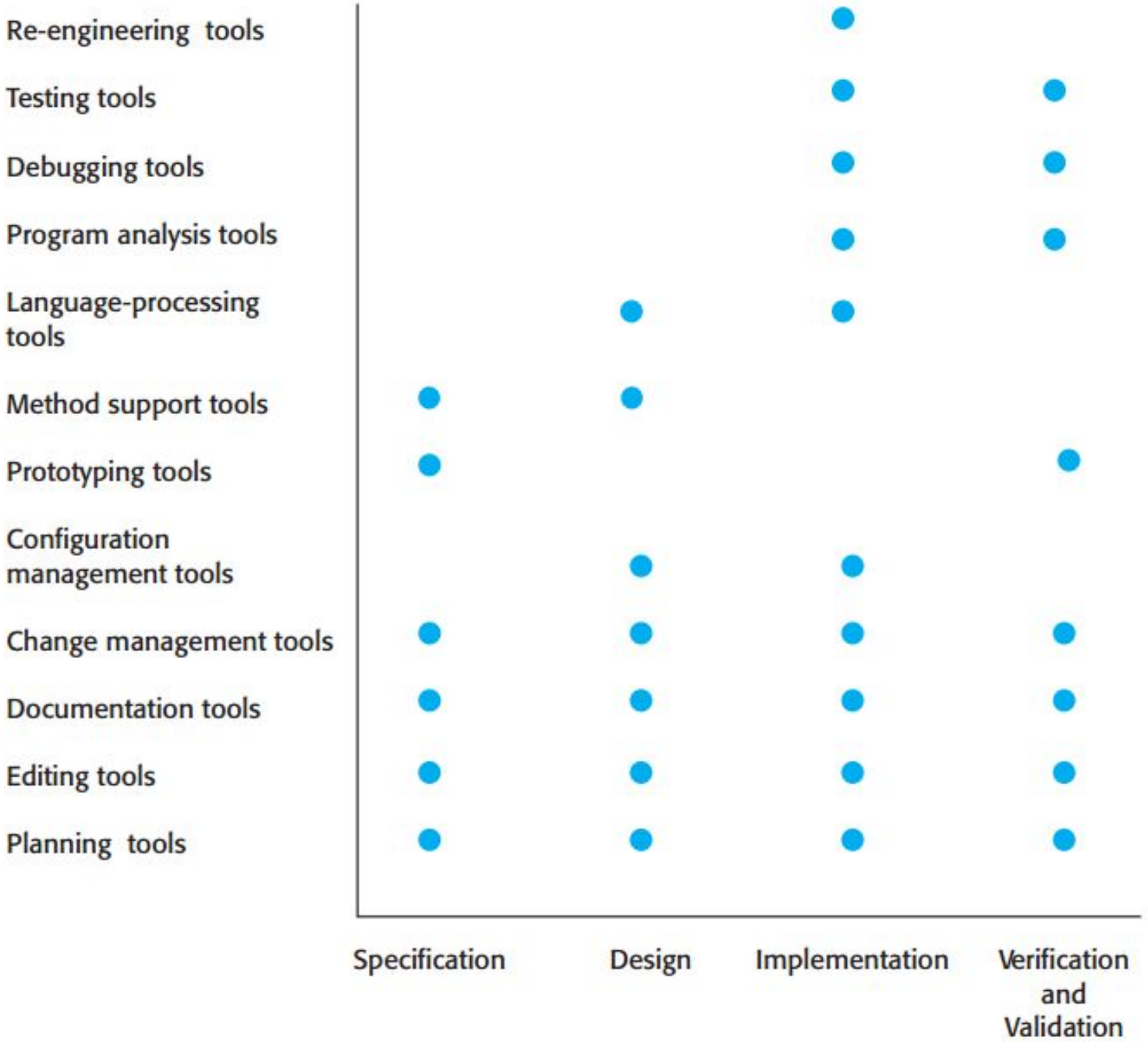
Phân loại CASE theo chức năng (1)

Kiểu công cụ	Ví dụ
Công cụ lập kế hoạch	Công cụ PERT, công cụ ước lượng, bảng tính
Công cụ soạn thảo	Trình soạn thảo văn bản, biểu đồ, bộ xử lý từ ngữ
Công cụ quản lý sự thay đổi	Công cụ truy xuất yêu cầu, kiểm soát sự thay đổi
Công cụ quản lý cấu hình	Hệ thống quản lý phiên bản, công cụ xây dựng hệ thống
Công cụ tạo nguyên mẫu	Ngôn ngữ ở mức rất cao, bộ tạo giao diện người dùng
Ngôn ngữ hỗ trợ phương thức	Trình thiết kế, từ điển dữ liệu, bộ sinh mã

Phân loại CASE theo chức năng (2)

Kiểu công cụ	Ví dụ
Công cụ xử lý ngôn ngữ	Trình biên dịch (compilers), trình thông dịch (interpreters)
Công cụ phân tích chương trình	Bộ tạo tham chiếu chéo, trình phân tích tĩnh, trình phân tích động
Công cụ kiểm thử	Bộ tạo dữ liệu kiểm thử, trình so sánh tệp tin
Công cụ gỡ rối	Hệ thống gỡ rối tương tác
Công cụ tài liệu hóa	Chương trình bố trí trang, trình xử lý hình ảnh
Công cụ tái kỹ thuật	Hệ thống tham chiếu chéo, hệ thống tái cấu trúc chương trình

Phân loại CASE theo hoạt động



Tài liệu tham khảo

- ***Software engineering: A practitioner's approach***, Part 1 & Part 2, Roger S. Pressman, McGraw-Hill Higher Education, 2010. (#000021579)
- *Software Engineering*, Ian Sommerville, 10th Edition, 2016
- *Kỹ nghệ Phần mềm*, TS Lê Văn Phùng, Nhà xuất bản thông tin và truyền thông, 2014
- *Nhập Môn Công Nghệ Phần Mềm*, Phạm Thị Quỳnh