

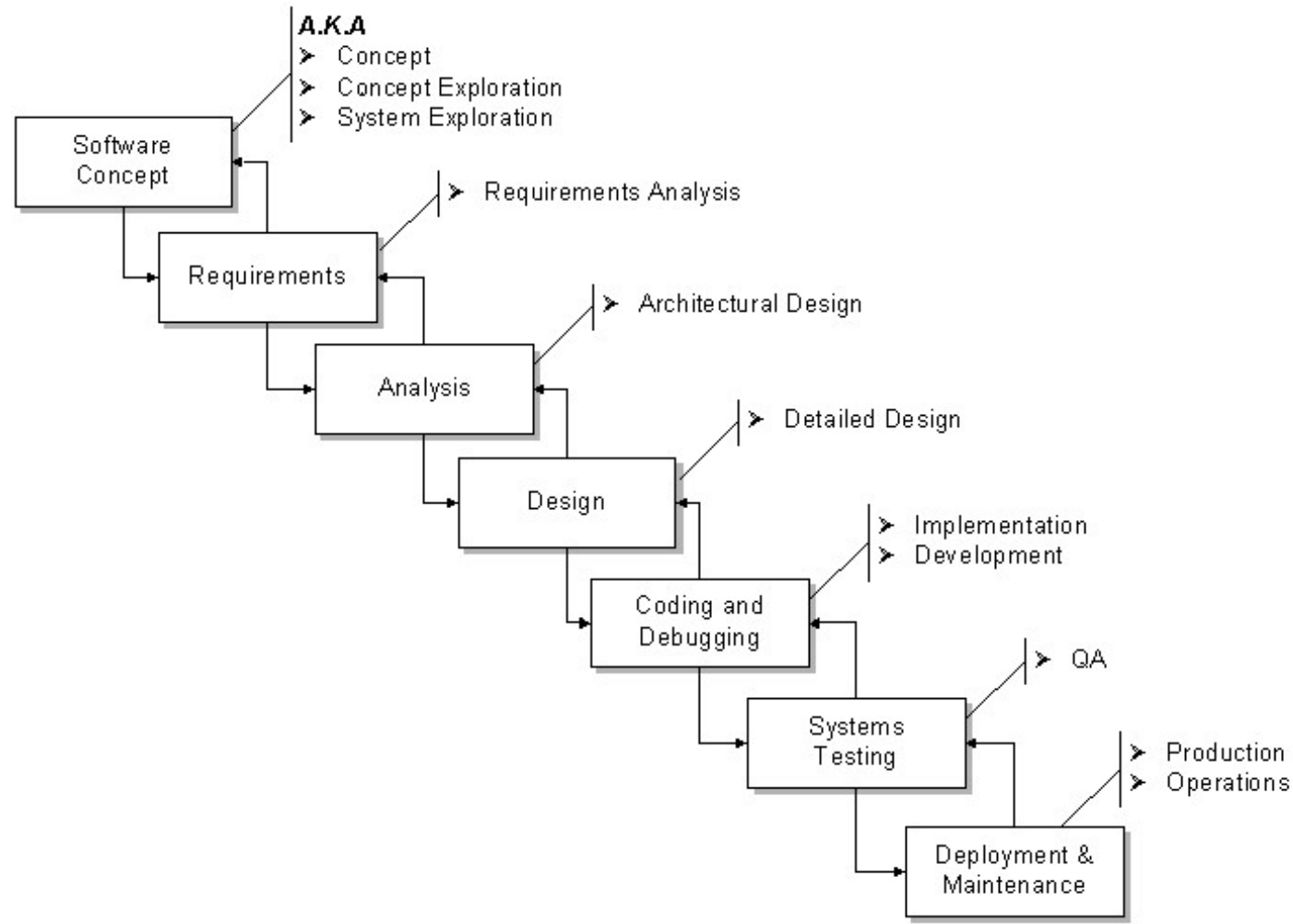
QUẢN LÝ DỰ ÁN PHẦN MỀM

Bài 3: Chuẩn bị, Khởi tạo và Lập kế hoạch dự án

Nội dung bài học

- 1. Các giai đoạn phát triển hệ thống (ôn tập)
 - Các bước điển hình của dự án phần mềm
- 2. Chuẩn bị
- 3. Khởi tạo dự án
- 4. Chu trình Lập kế hoạch

Các giai đoạn của dự án



Phân bố thời gian cho các pha

- Ghi nhớ luật 40-20-40
 - Xác định yêu cầu-Cài đặt-Kiểm thử

	Lập kế hoạch	Viết mã và kiểm thử chức năng	Tích hợp và kiểm thử
Sản phẩm thương mại	25%	40%	35%
Hệ thống Internet	55%	15%	30%
Hệ thống thời gian thực	35%	25%	40%
Hệ thống bảo vệ an toàn	40%	20%	40%

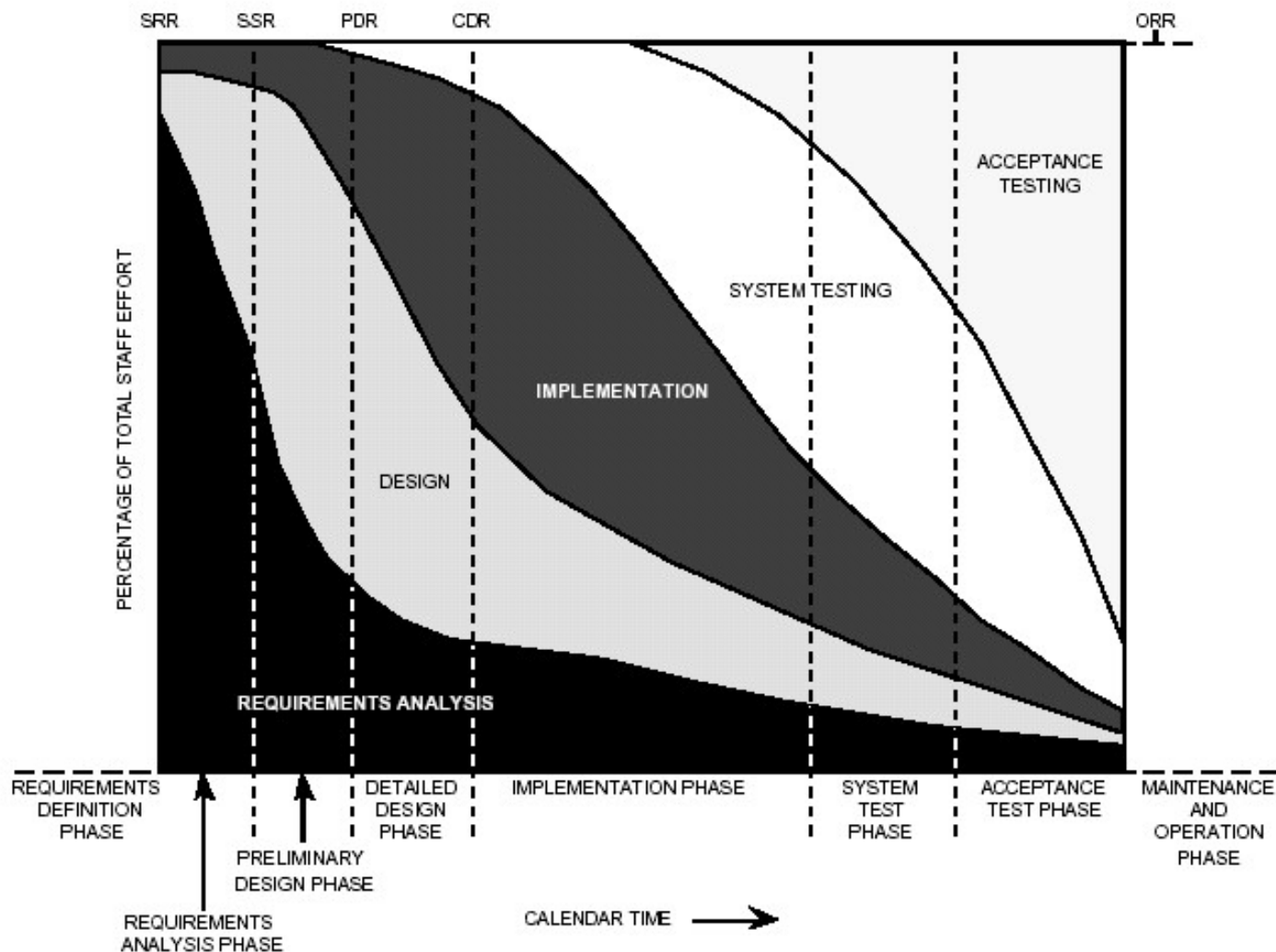
Bennatan, E.M, "On Time Within Budget"

Phân bố thời gian cho các pha

Hoạt động	Dự án nhỏ (2.5K LOC)	Dự án lớn (500K LOC)
Phân tích	10%	30%
Thiết kế	20%	20%
Viết mã	25%	10%
Kiểm thử chức năng	20%	5%
Tích hợp	15%	20%
Kiểm thử hệ thống	10%	15%

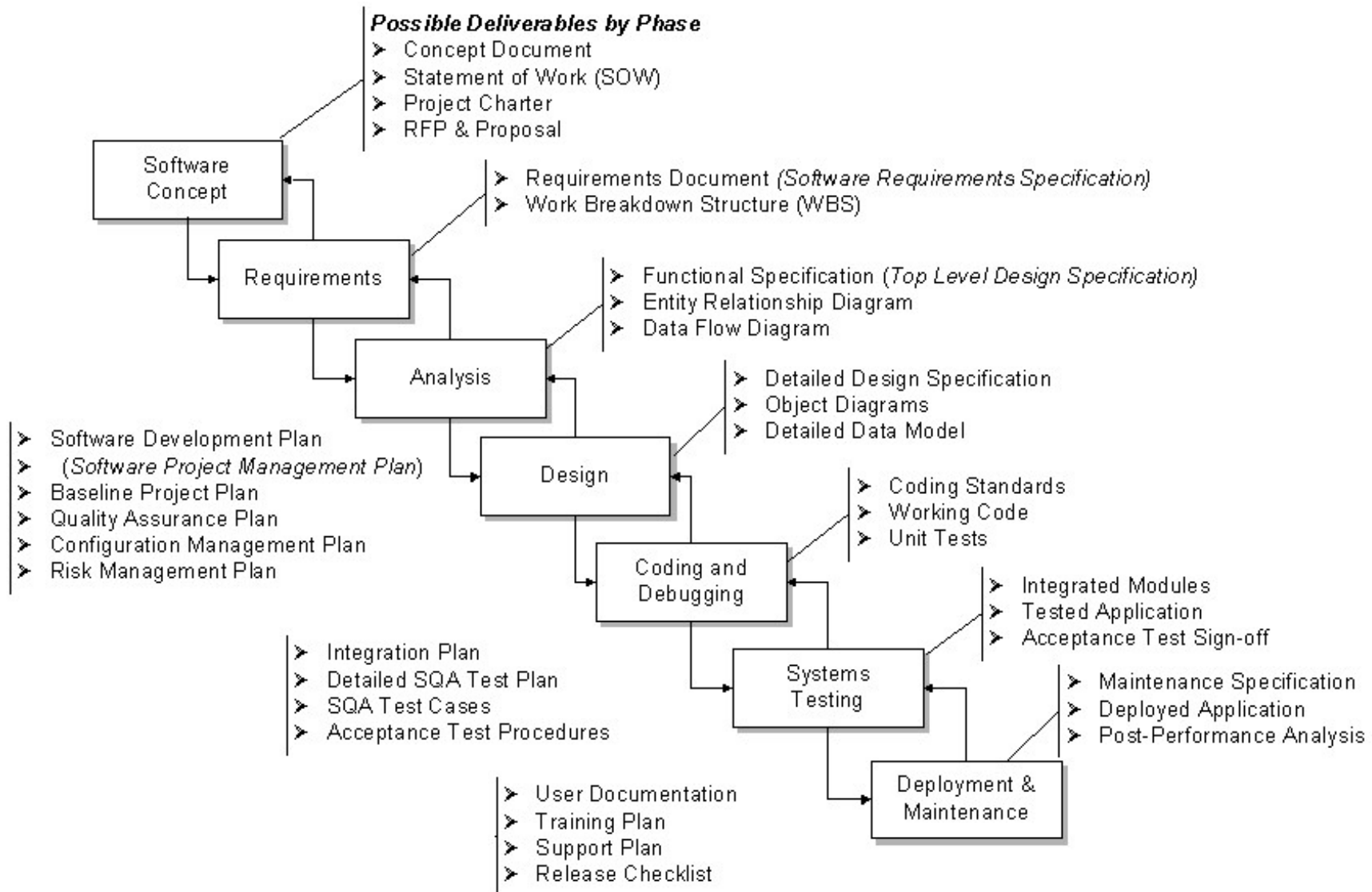
McConnell, Steve, "Rapid Development"

Phân bố công cho các pha



NASA's "Manager's Handbook for Software Development"

Các sản phẩm phân phối của mỗi pha



Tìm hiểu khái niệm

- The “Why” phase
- Not a “mandatory formal” phase
 - Sometimes called the “pre-project” phase
- Collecting project ideas
 - Then the “funneling” process
- Project Justification
 - ROI
 - Cost-benefit analysis
 - Project Portfolio Matrix
- Initial planning and estimates

Concept Exploration

- Possibly includes Procurement Management:
 - RFP Process
 - Vendor selection
 - Contract management
- Gathering the initial team
 - Including PM if not already on-board
- Identify the project sponsor
 - Primary contact for approval and decision making
- Potential Phase Outputs:
 - Concept Document, Product Description, Proposal, SOW, Project Charter

Concept Exploration

- Characteristics & Issues
 - Lack of full commitment and leadership
 - Some frustrations:
 - Management only getting rough estimates from development
 - Development not getting enough specifics from customer
 - Finding a balanced team
 - Budget sign-off may be your 1st major task
 - Achieved via:
 - Good concept document or equivalent
 - Demonstration of clear need (justification)
 - Initial estimates

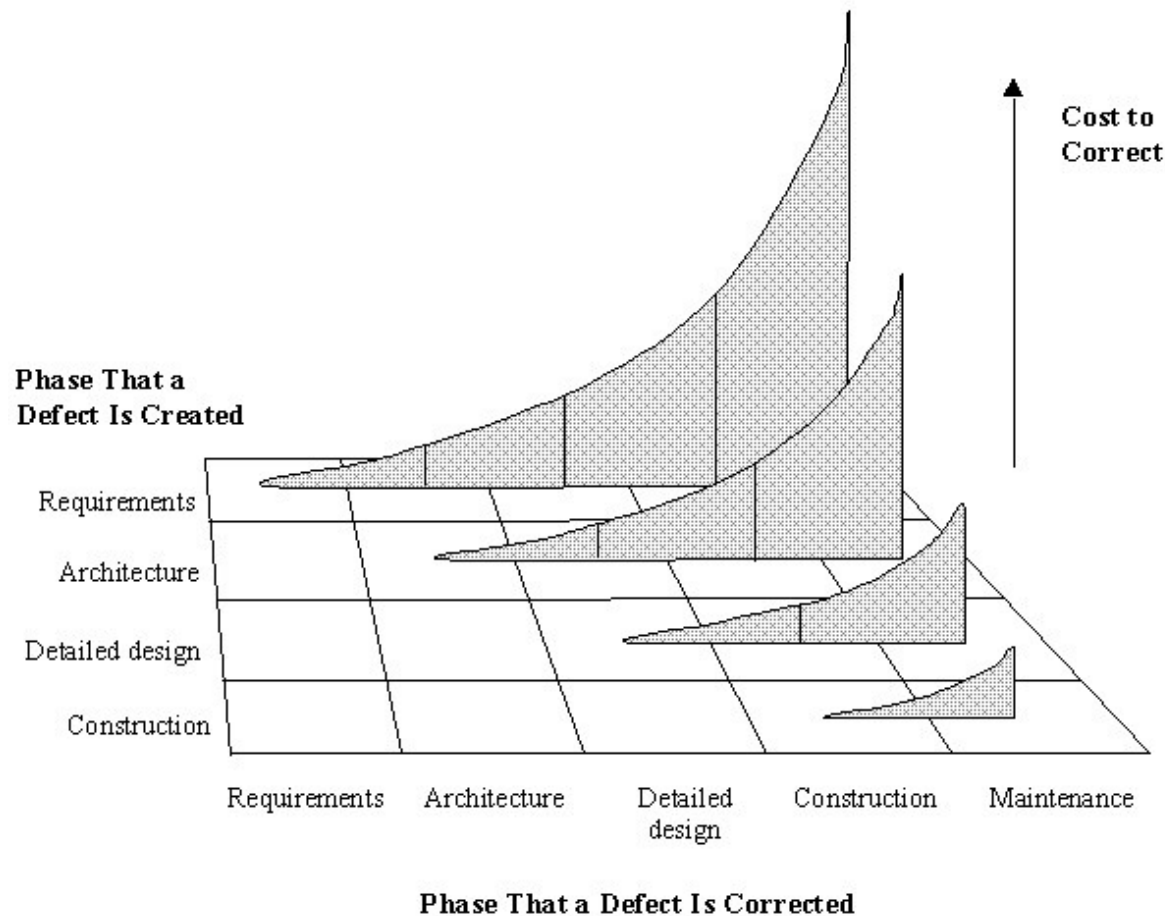
Requirements

- The “What” phase
- Inputs: SOW, Proposal
- Outputs:
 - Requirements Document (RD)
 - a.k.a. Requirements Specification Document (RSD)
 - Software Requirements Specification (SRS)
 - 1st Project Baseline
 - Software Project Management Plan (SPMP)
 - Requirements Approval & Sign-Off
 - Your most difficult task in this phase

Requirements

- Perhaps most important & difficult phase
- Shortchanging it is a ‘classic mistake’
- Can begin with a Project Kickoff Meeting
- Can end with a Software Requirements Review (SRR)
 - For Sponsor and/or customer(s) approval

Why are Requirements so Important?



Copyright 1998 Steven C. McConnell. Reprinted with permission from *Software Project Survival Guide* (Microsoft Press, 1998).

Requirements

- Characteristics & Issues
 - Conflict of interest: developer vs. customer
 - Potential tug-of-war:
 - Disagreement on Features & Estimates
 - Especially in fixed-price contracts
 - Frequent requirements changes
 - Achieving sign-off
- Project planning occurs in parallel

Requirements

- Requirements are capabilities and condition to which the system – more broadly, the project – must conform

2 Types of Requirements

- **Functional** (behavioral)
 - Features and capabilities
- **Non-functional** (a.k.a. “technical”) (everything else)
 - Usability
 - » Human factors, help, documentation
 - Reliability
 - » Failure rates, recoverability, availability
 - Performance
 - » Response times, throughput, resource usage
 - Supportability
 - » Maintainability, internationalization
 - Operations: systems management, installation
 - Interface: integration with other systems
 - Other: legal, packaging, hardware

Requirements

- Other ways of categorizing
 - Go-Ahead vs. Catch-up
 - Relative to competition
 - Backward-looking vs. Forward-looking
 - Backward: address issues with previous version
 - Forward: Anticipating future needs of customers
- Must be prioritized
 - Must-have
 - Should-have
 - Could-have (Nice-to-have: NTH)
- Must be approved

Early Phase Meetings

- Project Kickoff Meeting
- Project Brainstorming Meeting
 - Clarify goals, scope, assumptions
 - Refine estimates
- WBS Meeting

Analysis & Design

- The “How” Phases
- Inputs: Requirements Document
- Outputs:
 - Functional Specification
 - Detailed Design Document
 - User Interface Specification
 - Data Model
 - Prototype (can also be done with requirements)
 - Updated Plan (improved estimates; new baseline)

Analysis & Design

- a.k.a. Top-level design & detailed design
- Continues process from RD
- Ends with Critical Design Review (CDR)
 - Formal sign-off
 - Can also include earlier Preliminary Design Review (PDR) for high level design

Analysis & Design

- Characteristics & Issues
 - Enthusiasm via momentum
 - Team structure and assignments finalized
 - Delays due to requirements changes, new information or late ideas
 - Issues around personnel responsibilities
 - Unfeasible requirements (technical complexity)
 - Resource Issues
 - Including inter-project contention

Development

- The “Do It” phase
- Coding & Unit testing
- Often overlaps Design & Integration phases
 - To shorten the overall schedule
 - PM needs to coordinate this

Development

- Other concurrent activities
 - Design completion
 - Integration begins
 - Unit testing of individual components
 - Test bed setup (environment and tools)
 - Project plans updated
 - Scope and Risk Management conducted

Development

- Characteristics
 - Pressure increases
 - Staffing at highest levels
 - Often a “heads-down” operation
- Issues
 - Last-minute changes
 - Team coordination (esp. in large projects)
 - Communication overhead
 - Management of sub-contractors

Integration & Test

- Evolves from Dev. Phase
- Often done as 2 parallel phases
 - Partial integration & initial test
- Starts with integration of modules
- An initial, incomplete version constructed
- Progressively add more components

Integration & Test

- Integration primarily a programmer task
- Test primarily a QA team task
- Integration:
 - Top-down: Core functionality first, empty shells for incomplete routines (stubs)
 - Bottom up: gradually bind low-level modules
 - Prefer top-down generally

Integration & Test

- Tests
 - Integration testing
 - Black & White-box testing
 - Load & Stress testing
 - Alpha & Beta testing
 - Acceptance testing
- Other activities
 - Final budgeting; risk mgmt.; training; installation preparation; team reduced

Integration & Test

- Characteristics & Issues
 - Increased pressure
 - Overtime
 - Customer conflicts over features
 - Frustration over last-minute failures
 - Budget overruns
 - Motivation problems (such as burnout)
 - Difficulty in customer acceptance
 - Esp. true for fixed-price contracts

Deployment & Maintenance

- Installation depends on system type
 - Web-based, CD-ROM, in-house, etc.
- Migration strategy
- How to get customers up on the system
 - Parallel operation
- Deployment typically in your project plan, maintenance not

Deployment & Maintenance

- Maintenance
 - Fix defects
 - Add new features
 - Improve performance
- Configuration control is very important here
- Documents need to be maintained also
- Sometimes a single team maintains multiple products

Deployment & Maintenance

- Characteristics & Issues
 - Lack of enthusiasm
 - Pressure for quick fixes
 - Insufficient budget
 - Too many patches
 - Personnel turnover
 - Regression testing is critical
 - Preferably through automated tools

Lifecycle Planning

- a.k.a. Lifecycle Management or SDLC
- Greatly influences your chance of success
- Not choosing a lifecycle is a bad option
- Three primary lifecycle model components
 - Phases and their order
 - Intermediate products of each phase
 - Reviews used in each phase

Lập kế hoạch chu trình phát triển

- Các dự án khác nhau đòi hỏi các cách tiếp cận khác nhau
- Không cần biết tất cả các mô hình theo tên
- Nên biết với từng hoàn cảnh thì loại chu trình nào phù hợp nhất
- Một chu trình không phải là một kỹ thuật thiết kế, mô hình hoặc vẽ các lược đồ
 - Cùng một kỹ thuật (UML, DFD, etc) có thể được sử dụng với nhiều loại chu trình phát triển

Pure Waterfall

- The “granddaddy” of models
- Linear sequence of phases
 - “Pure” model: no phases overlap
- Document driven
- All planning done up-front

Waterfall Risk

- Why does the waterfall model “invite risk”?
- Integration and testing occur at the end
 - Often anyone’s 1st chance to “see” the program

Pure Waterfall

- Works well for projects with
 - Stable product definition
 - Well-understood technologies
 - Quality constraints stronger than cost & schedule
 - Technically weak staff
 - Provides structure
 - Good for overseas projects

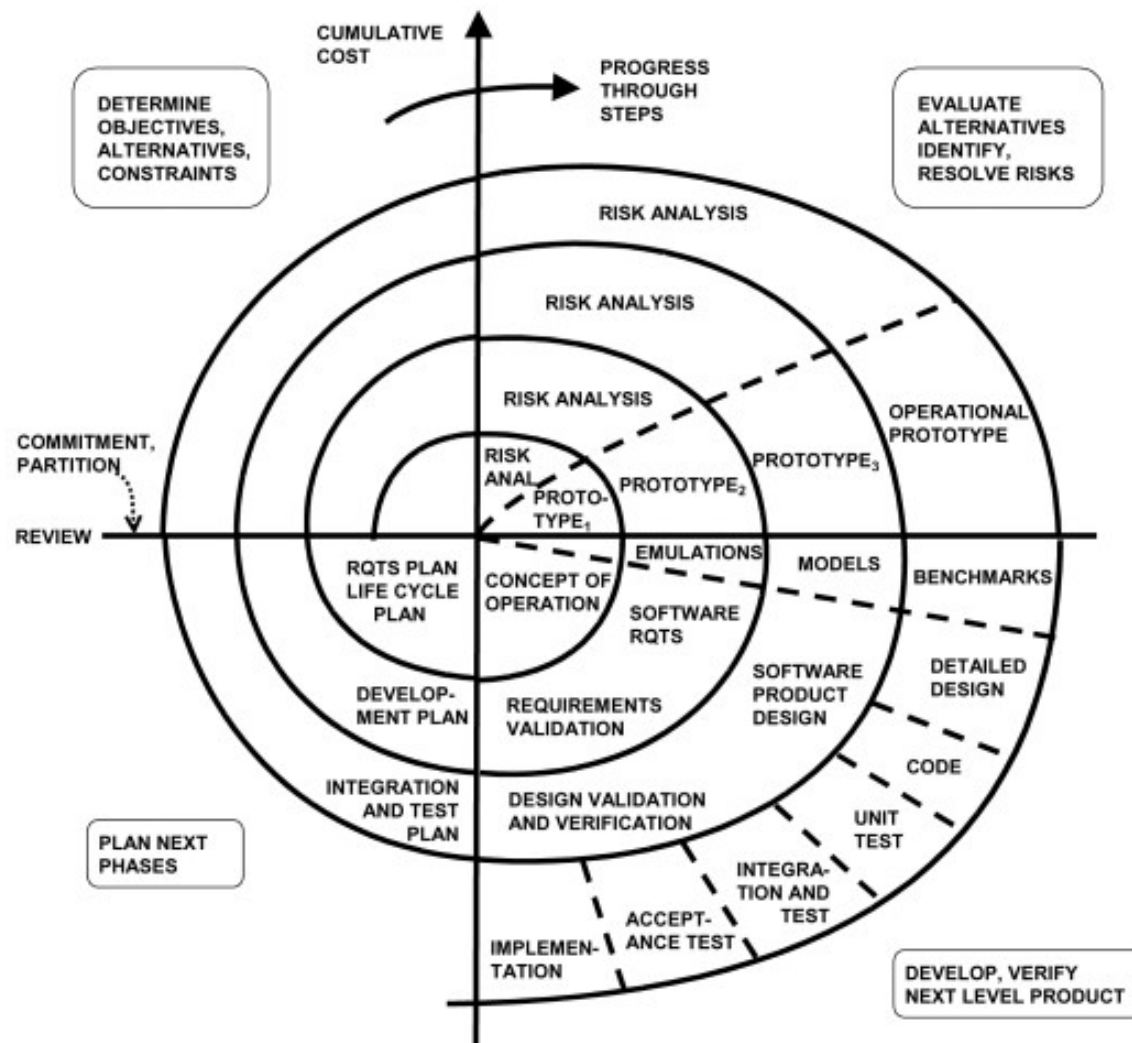
Pure Waterfall

- Disadvantages
 - Not flexible
 - Rigid march from start->finish
 - Difficult to fully define requirements up front
 - Can produce excessive documentation
 - Few visible signs of progress until the end

Code-and-Fix

- “Code-like-Hell”
- Specification (maybe), Code (yes), Release (maybe)
- Advantages
 - No overhead
 - Requires little expertise
- Disadvantages
 - No process, quality control, etc.
 - Highly risky
- Suitable for prototypes or throwaways

Spiral



Spiral

- Emphasizes risk analysis & mgmt. in each phase
- A Series of Mini-projects
- Each addresses a set of “risks”
 - Start small, explore risks, prototype, plan, repeat
- Early iterations are “cheapest”
- Number of spirals is variable
 - Last set of steps are waterfall-like

Spiral

- Advantages
 - Can be combined with other models
 - As costs increase, risks decrease
 - Risk orientation provides early warning
- Disadvantages
 - More complex
 - Requires more management

Modified Waterfall – Sashimi

- Overlapping phases
- Advantages
 - Reduces overall schedule
 - Reduces documentation
 - Works well if personnel continuity
- Disadvantages
 - Milestones more ambiguous
 - Progress tracking more difficult
 - Communication can be more difficult

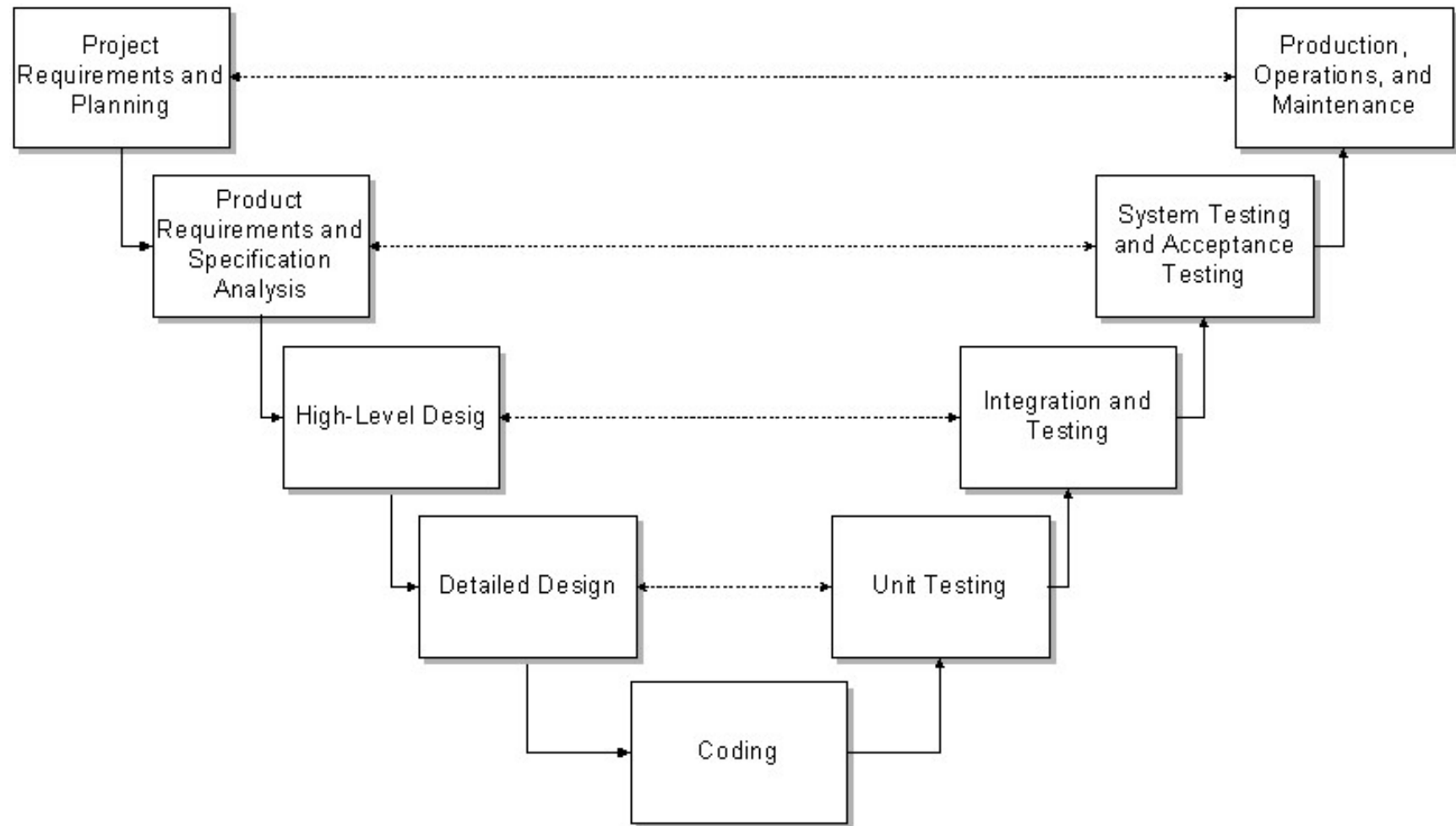
Evolutionary Prototyping

- Design most prominent parts first
 - Usually via a visual prototype
- Good for situations with:
 - Rapidly changing requirements
 - Non-committal customer
 - Vague problem domain
- Provides steady, visible progress
- Disadvantages
 - Time estimation is difficult
 - Project completion date may be unknown
 - An excuse to do “code-and-fix”

Staged Delivery

- Waterfall steps through architectural design
- Then detailed design, code, test, deliver in stages
- Advantages
 - Customers get product much sooner
 - Tangible signs of progress sooner
 - Problems discovered earlier
 - Increases flexibility
 - Reduces: status reporting overhead & estimation error
- Disadvantages
 - Requires more planning (for you the PM)
 - More releases increase effort (and possible feature creep)
- How's this differ from Evolutionary Prototyping?

V Process Model



V Process Model

- Designed for testability
 - Emphasizes Verification & Validation
- Variation of waterfall
- Strengths
 - Encourages V&V at all phases
- Weaknesses
 - Does not handle iterations
 - Changes can be more difficult to handle
- Good choice for systems that require high reliability such as patient control systems

RAD

- Rapid Application Development
- Popular in the 80's
 - 1. Joint Requirements Planning (JRP)
 - 2. Joint Application Design (JAD)
 - 3. Construction
 - Heavy use of tools: code generators
 - Time-boxed; many prototypes
 - 4. Cutover
- Good for systems with extensive user input available

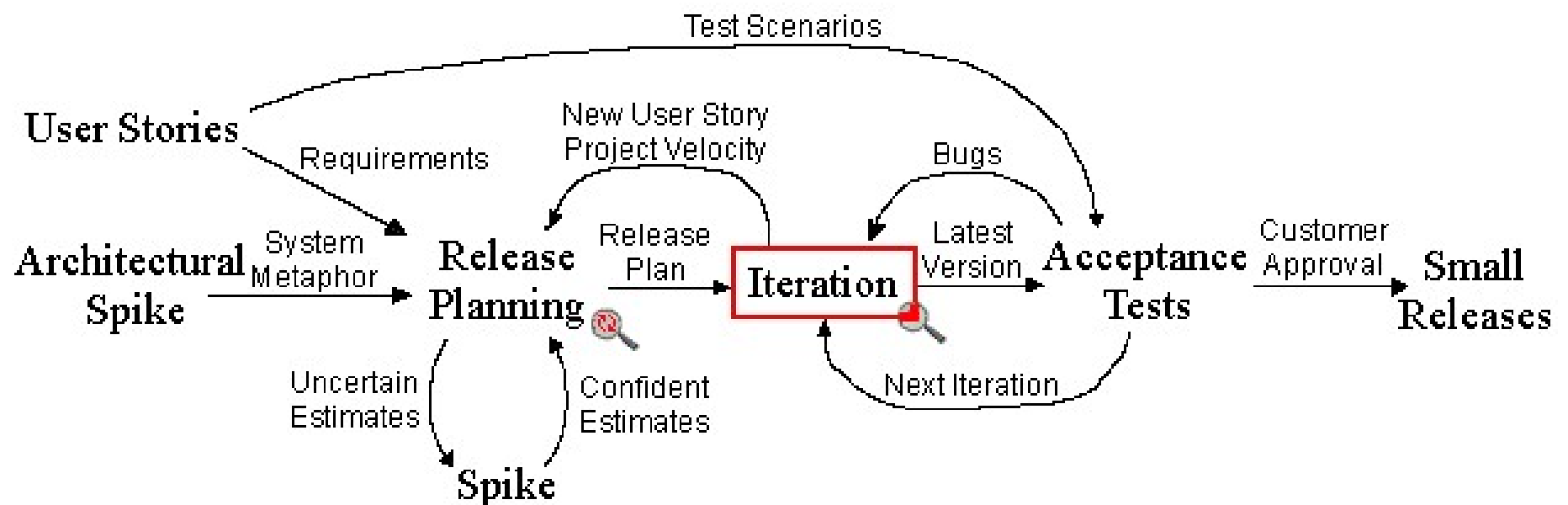
COTS

- Commercial Off-The-Shelf software
- Build-vs.-buy decision
- Advantages
 - Available immediately
 - Potentially lower cost
- Disadvantages
 - Not as tailored to your requirements
- Remember: custom software rarely meets its ideal (so compare that reality to COTS option)

XP: eXtreme Programming

- Not a Microsoft product
- Part of movement called “Agile Development”
- A “Lightweight” methodology
- A bit counter-culture
- Currently in vogue
- Motto: “Embrace Change”
- Highly Incremental / Iterative

eXtreme Programming



eXtreme Programming

- Suitable for small groups
- Attempts to minimize unnecessary work
- Uses an “on-site” customer
- Small releases
- Pair programming
- Refactoring
- Stories as requirements
- You want good developers if you use this

Other “Agile” Methodologies

- Agile here means “lite”, reduced docs, highly iterative
- Agile Software Development
 - Alliance, their “manifesto”, their book
- SCRUM
 - Features 30-day “Sprint” cycles
- Feature Driven Development (FDD)
 - XP with more emphasis on docs and process

Other “Agile” Methodologies

- Adaptive Software Development (ASD)
 - Book, site
- Dynamic System Development Method (DSDM)
 - Popular in Europe
- Homegrown: developers often hide their “agile adventures” from management

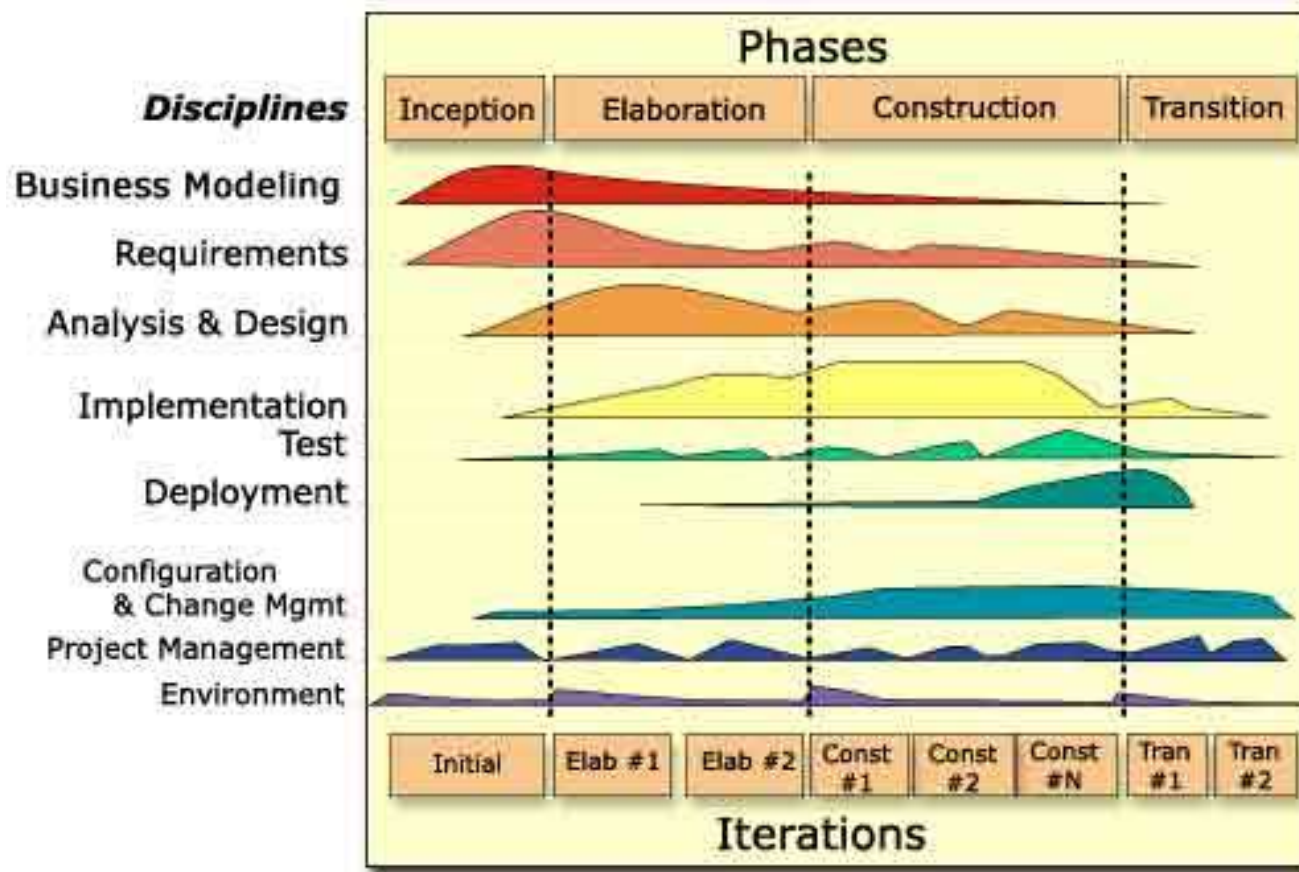
Other “Agile” Methodologies

- Pros
 - Similar to XP, can reduce process overhead
 - Responsive to user feedback
 - Amenable to change
- Cons
 - Requires close monitoring by PM
 - May not “scale” to large projects
 - Often requires better quality developers

Rational Unified Process

- RUP
- From Rational Corporation
- “Generic” version is the Unified Process
- Commercial
- Extensive tool support (expensive)
- Object-oriented
- Incremental
- Newer

Rational Unified Process



Rational Unified Process

- Develop Iteratively
- Manage Requirements
- Uses UML (Unified Modeling Language)
- Produces “artifacts”
- Use component-based architecture
- Visually model software
- Complex process
- A “framework”
- Suitable for large scale systems

Lựa chọn chu trình phát triển

- Thay đổi từng dự án
- Lựa chọn theo việc lặp lại hay tăng thêm
- Yêu cầu được hiểu rõ như thế nào?
- Các rủi ro là gì?
- Có hạn kết thúc ấn định không?
- Đội dự án và khách hàng có kinh nghiệm thế nào?

IEEE 1074

- Một chuẩn cho quá trình phát triển phần mềm
 - Lựa chọn chu trình phát triển
 - Tiến trình quản lý dự án
 - Tiến trình tiền phát triển
 - Các quá trình phát triển
 - Các quá trình sau khi phát triển xong hệ thống
 - Quá trình tích hợp

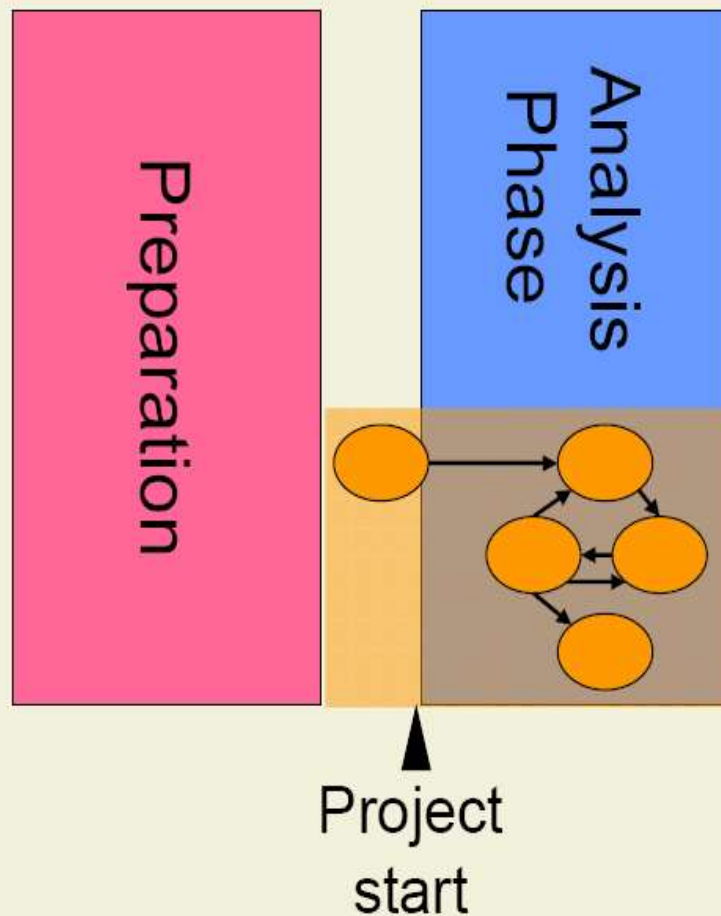
Chuẩn bị: Product sketch

- Các câu hỏi cần thiết
 - Sản phẩm làm gì?
 - Sản phẩm hoạt động thế nào
 - Ai bị tác động bởi sản phẩm và những công việc gì được thực hiện bởi nhóm người này
- Mức độ chi tiết
 - Đánh giá tính khả thi của kỹ thuật
 - Cho phép ước lượng chi phí ban đầu
- Product sketch được viết cho những người không chuyên gia (quản lý)

Chuẩn bị: Tóm tắt

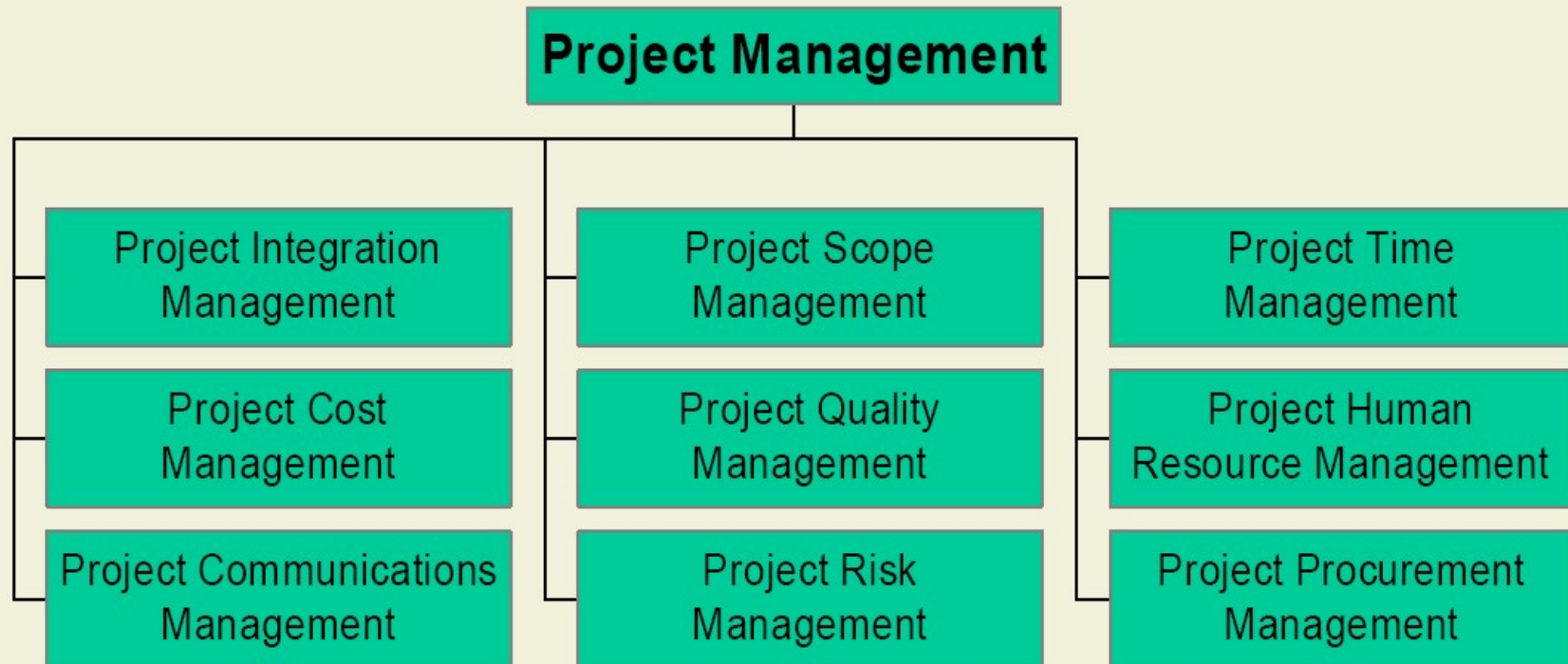
- Chuẩn bị không phải là một phần của dự án
- Mục đích
 - Phát triển các mục tiêu đúc kết được từ những ý tưởng mơ hồ ban đầu
- Các sản phẩm phân phối chính
 - Product sketch cho giấy tờ quyết định
- Các người tham gia chính
 - Khách hàng, giám đốc dự án, phân tích nghiệp vụ
- Các công cụ và kỹ thuật
 - Truyền thông
 - Nghiên cứu chuẩn bị ví dụ: phân tích ảnh hưởng

Starting a Project

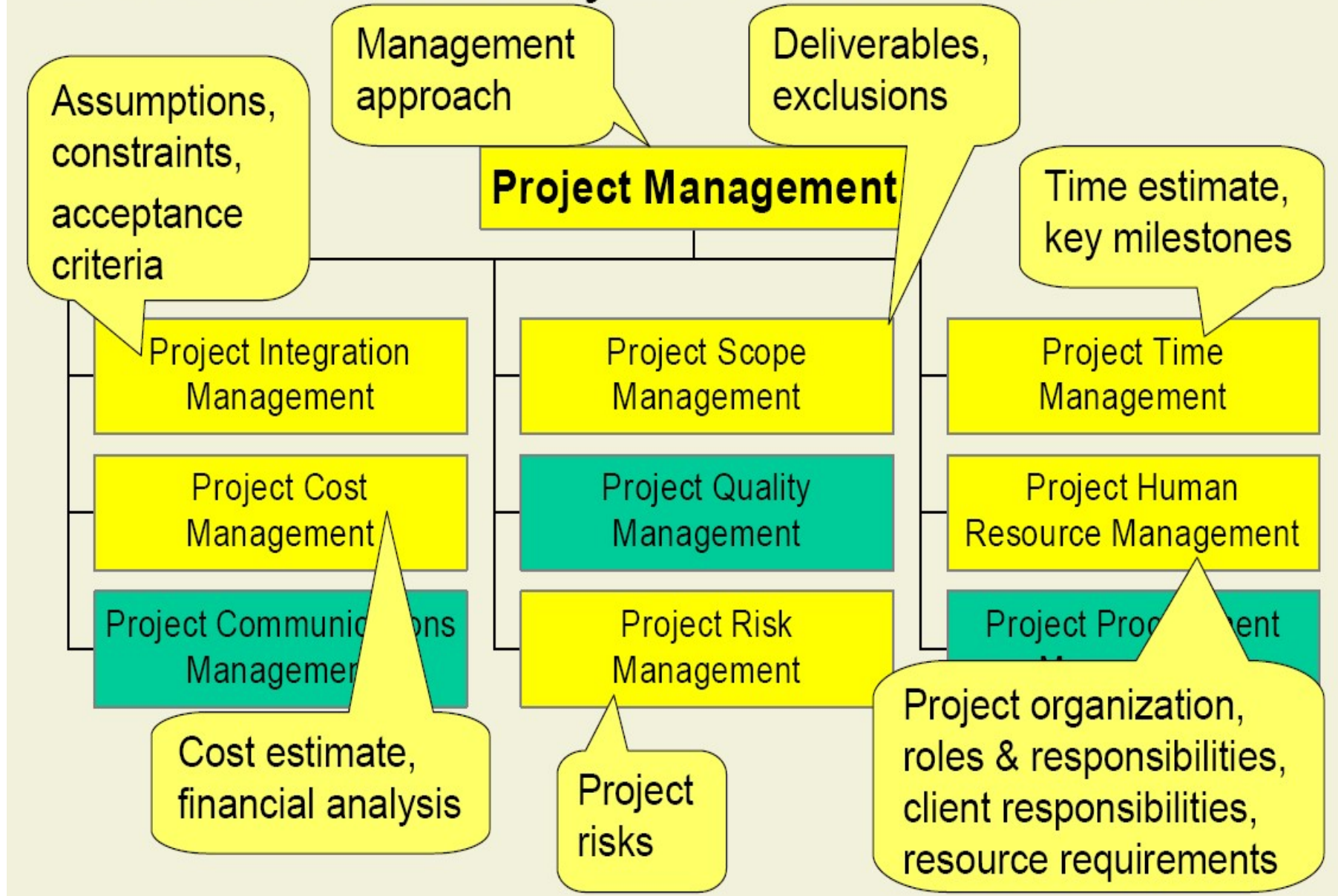


- Starting a project is mainly a project management activity
- Initiation prepares a decision by clients and management

What is Necessary to Take a Decision?



What is Necessary to Take a Decision?



Giả thiết

- Định nghĩa: các giả thiết là các yếu tố được cho là đúng, có thật hoặc chắc chắn cho mục đích lập kế hoạch
- Các giả thiết ảnh hưởng tới tất cả các khía cạnh của việc lập kế hoạch và là một phần thúc đẩy tiến độ của dự án
- Đội dự án thường xuyên xác định, lập tài liệu và xác định tính đúng đắn của giả thiết như một quá trình lập kế hoạch
- Giả thiết thường liên quan tới một mức độ rủi ro

Ràng buộc

- Định nghĩa: Ràng buộc là các yếu tố làm hạn chế sự lựa chọn của đội dự án
- Một dự án có thể bao gồm chi phí, thời gian, tài nguyên con người, kỹ thuật và các ràng buộc khác
- Ví dụ:
 - Các mốc thời gian ngoại cảnh
 - Cạn trên của ngân sách
 - Phụ thuộc với các dự án khác, v.v...

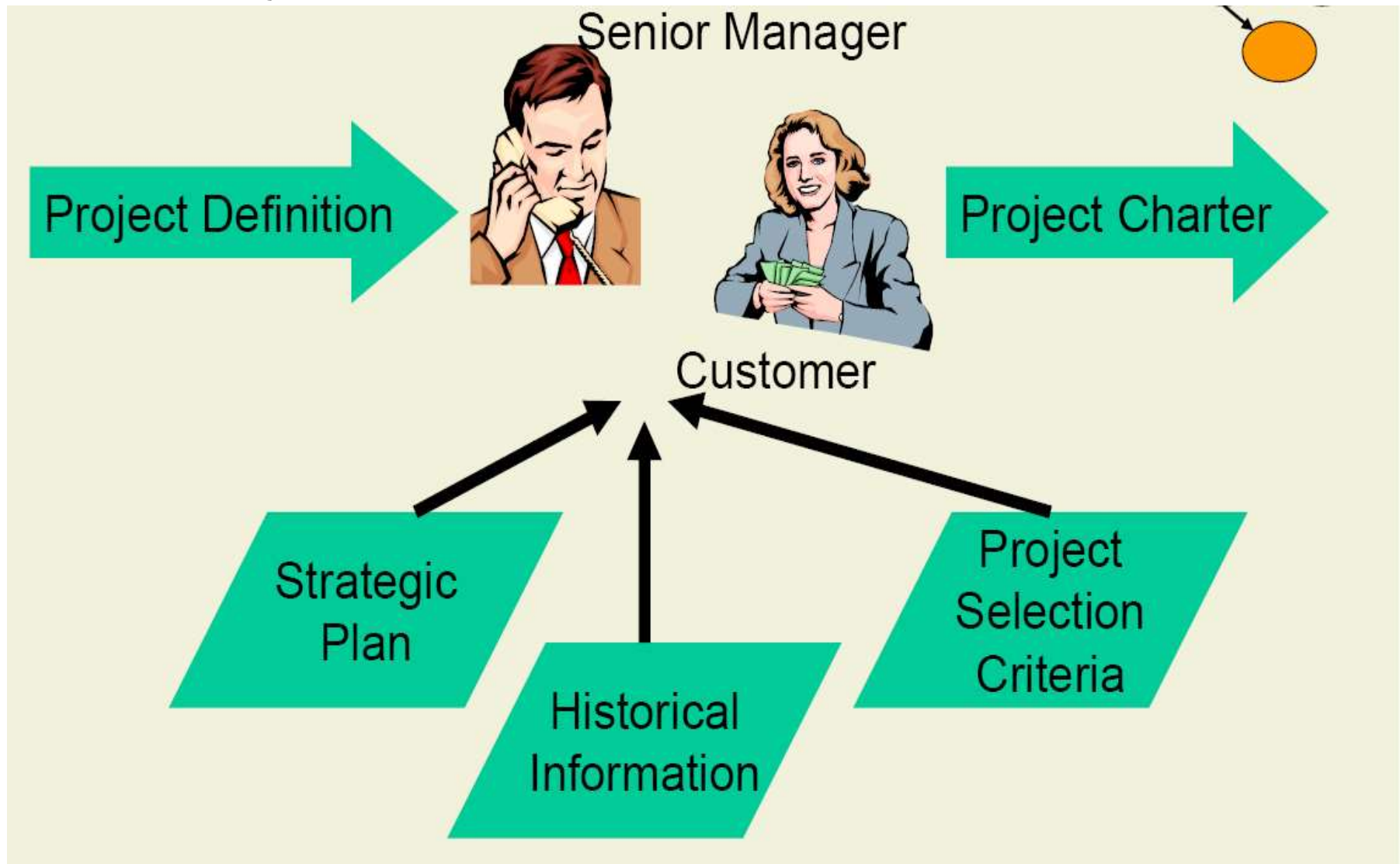
Người tham gia dự án

- Định nghĩa: các cá nhân và tổ chức tham gia tích cực vào dự án hoặc lợi nhuận của họ bị ảnh hưởng tốt hoặc xấu bởi quá trình thực thi hoặc kết thúc dự án; họ có thể ảnh hưởng nhiều tới dự án và kết quả của nó
- Những người tham gia dự án:
 - Giám đốc dự án
 - Khách hàng
 - Tổ chức thực hiện
 - các thành viên đội dự án
 - Người tài trợ

Định nghĩa dự án

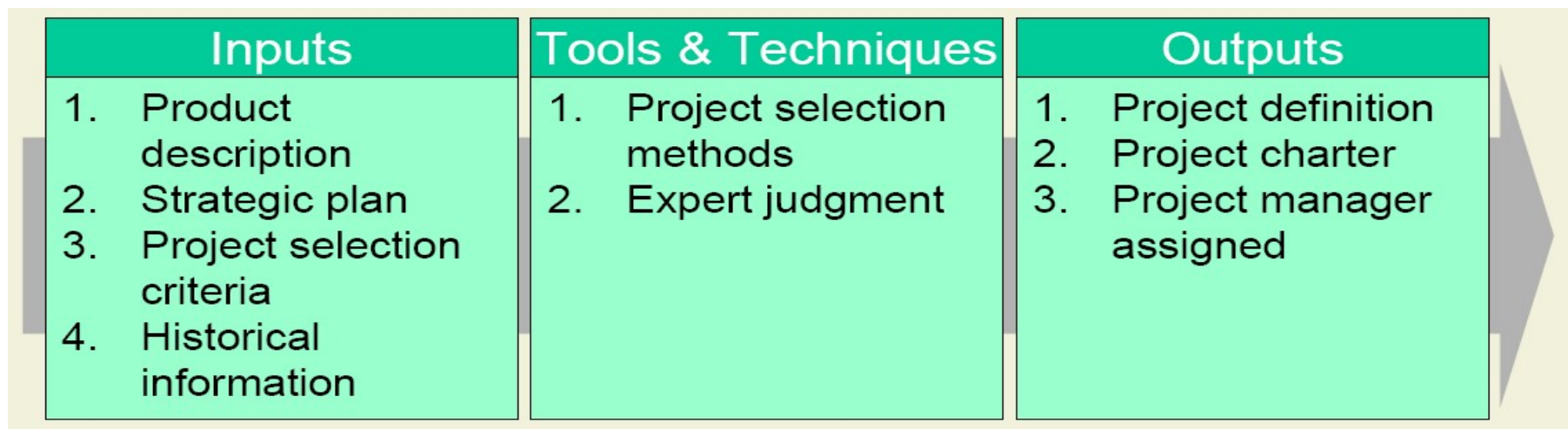
- Định nghĩa dự án là một tài liệu tổng quan thiết lập phạm vi của dự án
- Nó là cơ sở cho một quyết định của khách hàng và quản lý

Quá trình lựa chọn dự án



Quá trình khởi tạo: tóm tắt

- Mục đích:
 - chính thức khởi tạo một dự án mới hoặc pha tiếp theo của một dự án đã có
 - Lặp lại quá trình khởi tạo tại thời điểm bắt đầu của mỗi pha giúp cho dự án tập trung vào những nhu cầu nghiệp vụ



Lập kế hoạch

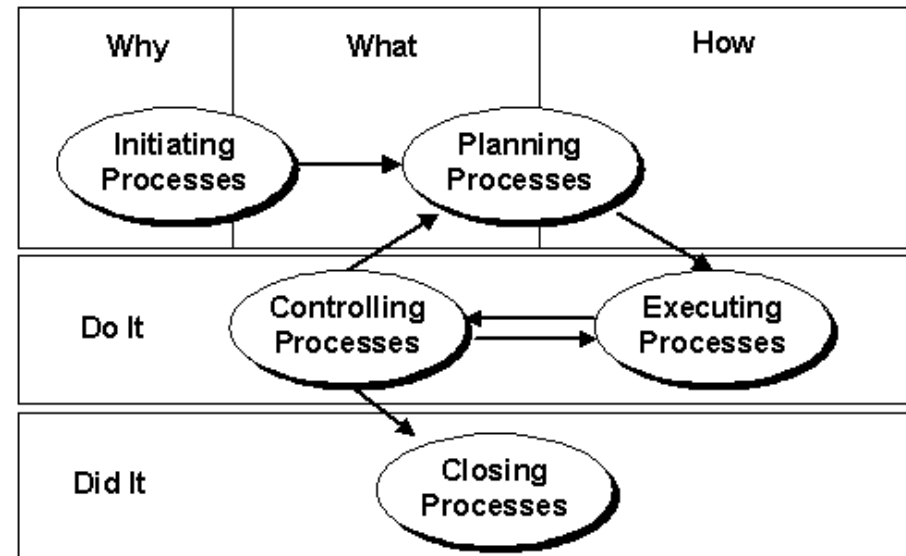
- “Kế hoạch không là gì. Nhưng kế hoạch là mọi thứ.” Gen. Dwight Eisenhower

Tại sao cần một kế hoạch dự án

- Một sản phẩm hoặc dịch vụ duy nhất
- Hướng dẫn thực thi dự án
- Lập tài liệu các giả thiết lập kế hoạch dự án
- Lập tài liệu các quyết định kế hoạch liên quan tới các phương án thay thế được chọn
- Tạo môi trường giao tiếp thuận lợi giữa người tham gia dự án
- Cung cấp bản kế hoạch gốc cho việc đo tiến độ và kiểm soát dự án

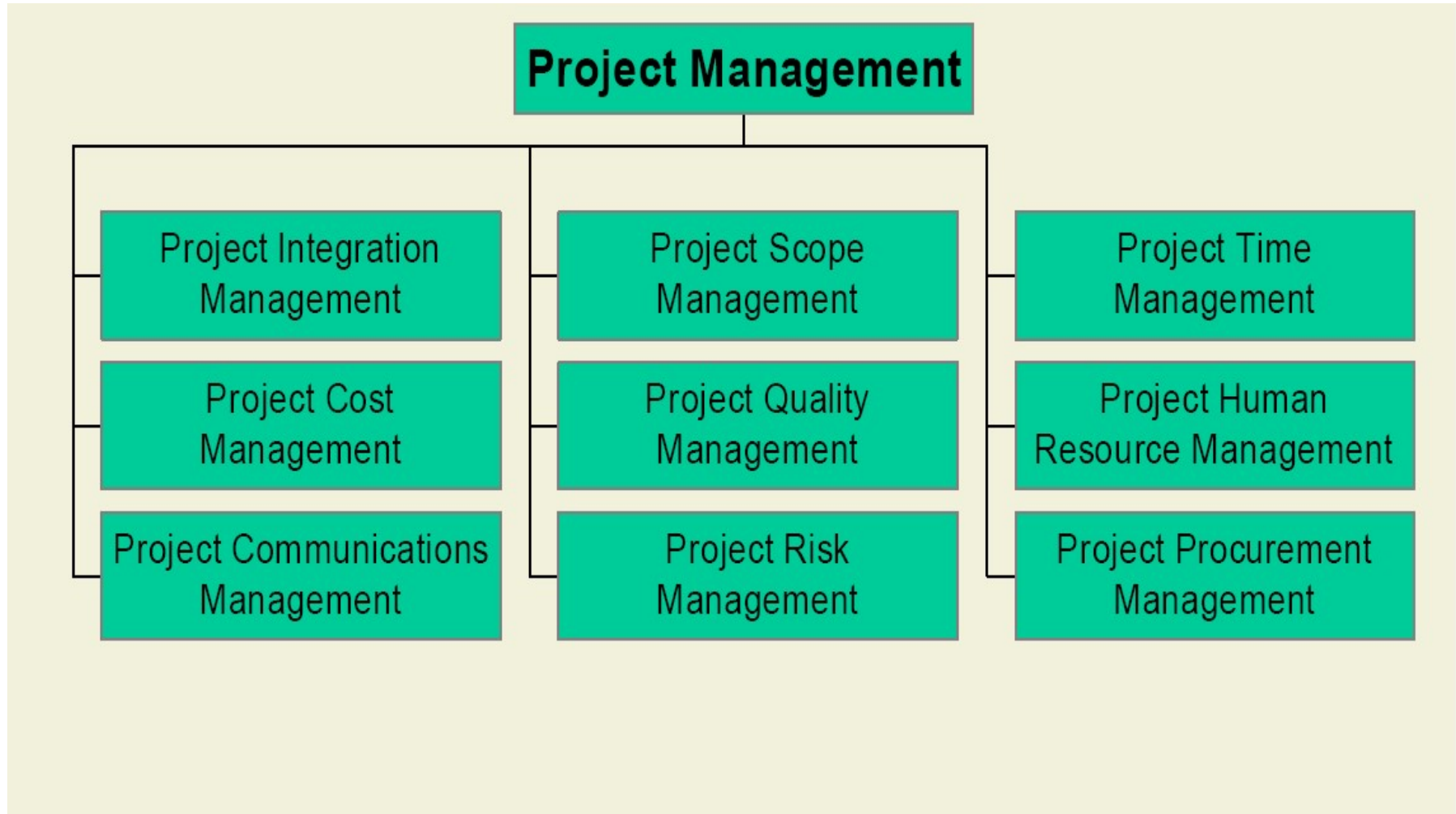
Quá trình quản lý dự án của bạn

- Tại sao
 - Sản phẩm phân phối: ROI
- Cái gì
 - Mô tả công việc, Các yêu cầu
- Như thế nào
 - Bản thiết kế cụ thể, SDP, chu trình sống
- Thực hiện
 - Tiến hành
- Xong
 - xem xét cuối dự án

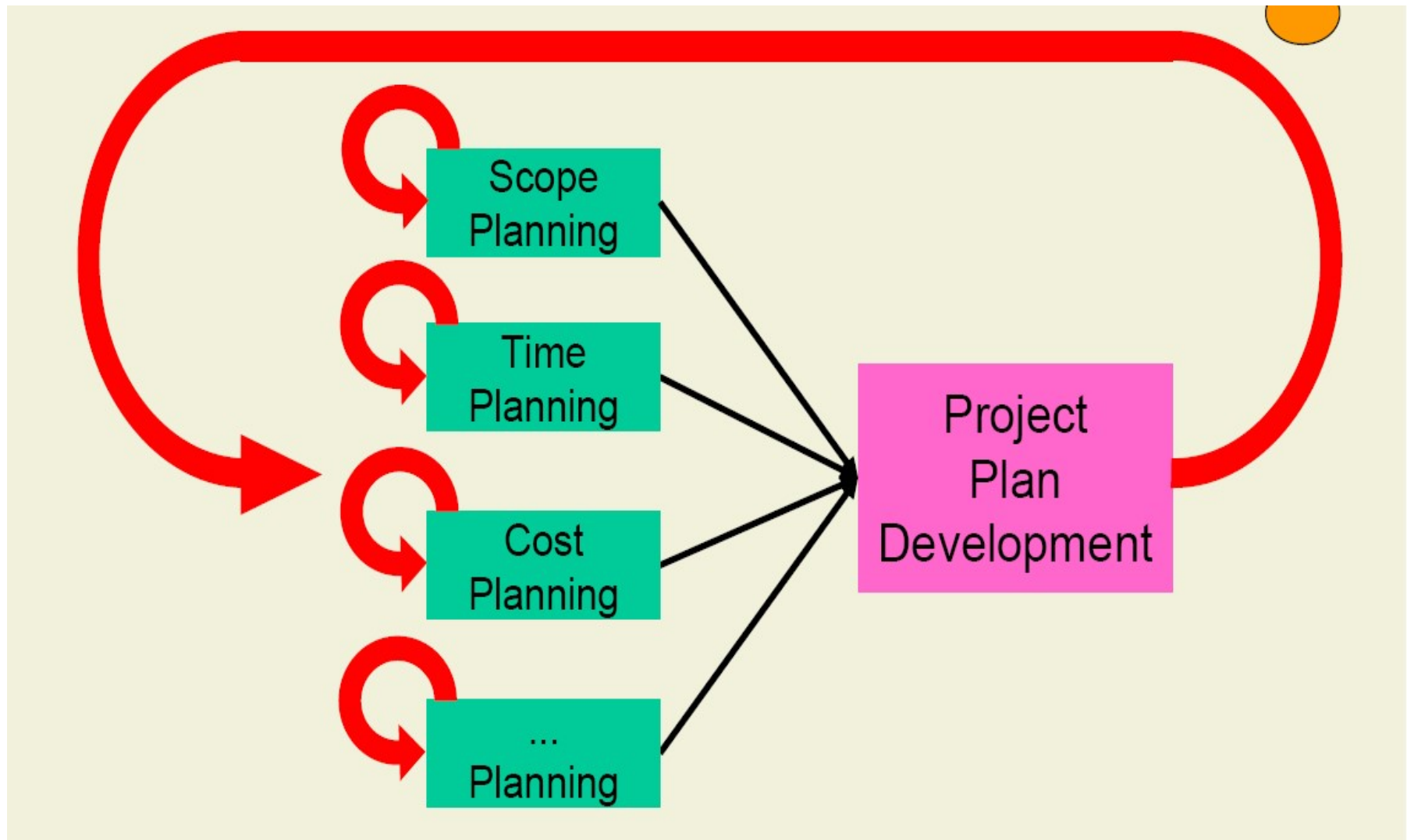


Futrell, Shafer, Shafer, "Quality Software Project Management"

Các khía cạnh của lên kế hoạch



Vòng lặp của việc lập kế hoạch



Tài liệu kế hoạch dự án

- Một tài liệu chính thức được thông qua
- Một kế hoạch dự án không chỉ là một lịch thực hiện
- Bao gồm
 - Cách tiếp cận quản lý dự án
 - Phạm vi, lập lịch, ước lượng chi phí, tài nguyên, trách nhiệm
 - kế hoạch quản lý tài trợ cho các khía cạnh trên
 - kế hoạch đo đạc năng suất cho phạm vi, lịch và chi phí
 - Các vấn đề mở và quyết định còn bị trì hoãn

Các tài liệu

- Lập kế hoạch
- Sản phẩm

Các tài liệu lập kế hoạch

- Bản kế hoạch phát triển phần mềm (SDP)
- Bản kế hoạch đảm bảo chất lượng phần mềm (SQAP)
- Bản kế hoạch quản lý cấu hình phần mềm (SCMP)
- Bản kế hoạch quản lý rủi ro
- Bản kế hoạch cải thiện tiến trình phần mềm
- Bản kế hoạch quản lý truyền thông
- Bản kế hoạch chuyển đổi hệ thống-Migration Plan
- Bản kế hoạch vận hành-Operations Plan

Các tài liệu lập kế hoạch

- Giám đốc dự án cần lựa chọn các tài liệu nào là thích hợp
- Các tài liệu không cần quá dài
- Một tập nhỏ:
 - Bản kế hoạch phát triển phần mềm
 - Bản kế hoạch quản lý rủi ro
 - Bản kế hoạch đảm bảo chất lượng phần mềm
 - Bản kế hoạch quản lý cấu hình phần mềm

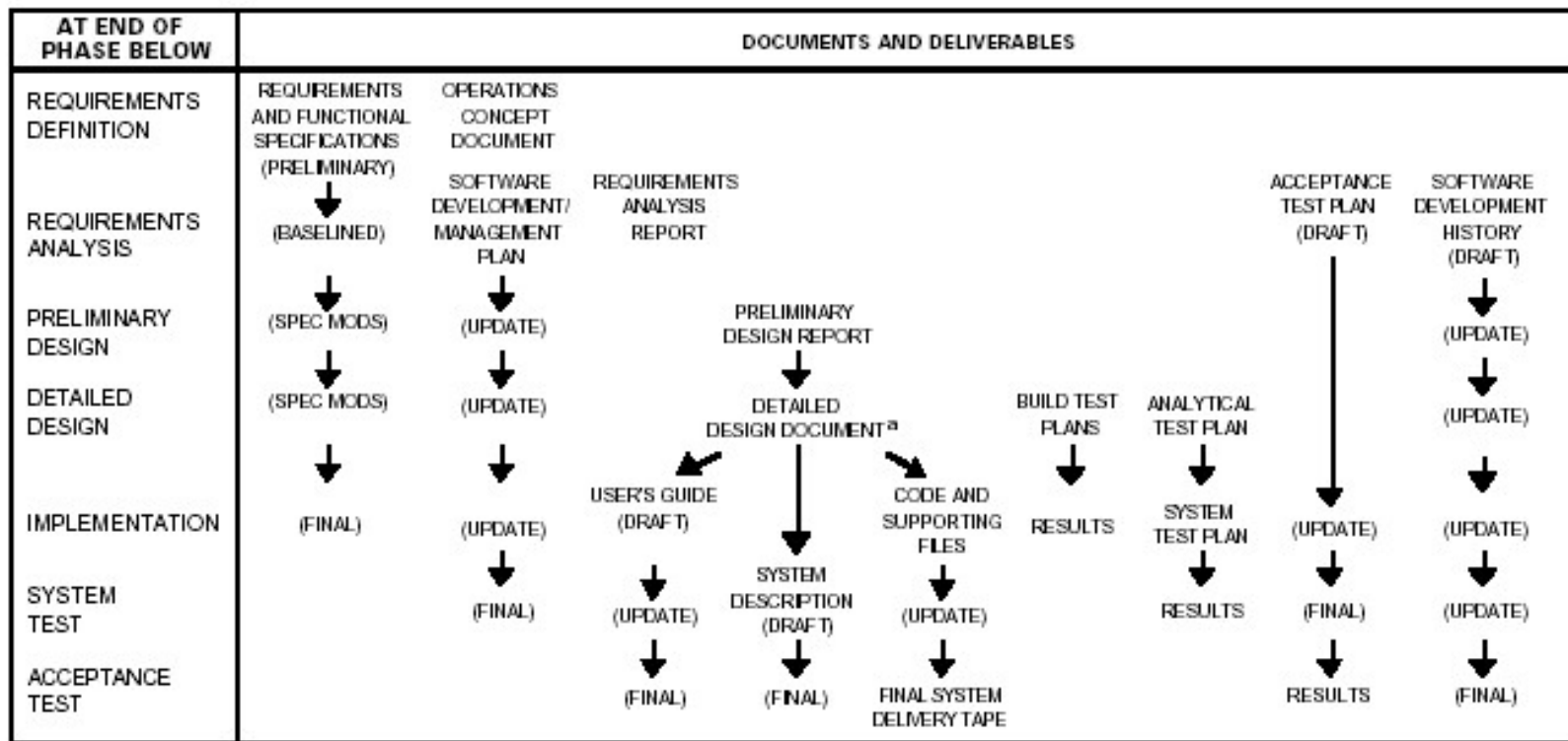
Các tài liệu lập kế hoạch

- Phân tích ROI của dự án
- Phát biểu bài toán
- Tôn chỉ dự án (Project Charter)
- Kế hoạch quản lý dự án phần mềm (SPMP)
- Ngân sách
- Ma trận gán trách nhiệm (RAM)
- Kế hoạch quản lý rủi ro

Các tài liệu cho sản phẩm

- Phát biểu nhu cầu
- Mô tả giao diện hệ thống
- Mô tả yêu cầu phần mềm
- Mô tả thiết kế phần mềm
- Kế hoạch xác thực phần mềm
- Tài liệu người dùng
- Kế hoạch hỗ trợ
- Tài liệu bảo dưỡng

Kế hoạch phát triển qua thời gian²



NASA's "Manager's Handbook for Software Development"

Kế hoạch phát triển phần mềm

- Kế hoạch quản lý dự án phần mềm
- Được coi là tài liệu quan trọng nhất trong dự án (cùng với SRS)
 - Có thể được xem như một bản tích hợp các tài liệu cơ bản khác
- Phát triển dần qua thời gian bằng cách thêm các phần nhỏ vào

Kế hoạch phát triển/quản lý dự án phần mềm (SDP / SPMP)

- Các phần cơ bản
 - Tổng quan dự án
 - Các sản phẩm phân phối
 - Tổ chức dự án
 - Các tiến trình quản lý
 - Các tiến trình kỹ thuật
 - Ngân sách
 - Lập lịch

Kế hoạch quản lý truyền thông

- Thường là một phần trong kế hoạch quản lý dự án phần mềm
- Mô tả luồng thông tin tới tất cả các bên liên quan
 - Thu thập và phân phối thông tin
- Hợp trạng thái
 - Hàng tháng, hàng tuần, hàng ngày?
 - Các báo cáo trạng thái là rất cần thiết