SOFTWARE ENGINEERING

CO3001

CHAPTER 2 – SOFTWARE PROCESSES

Truong Tuan Anh

Adapted from https://iansommerville.com/software-engineering-book/slides/

TOPICS COVERED

- ✓ Software process models
- ✓ Process activities
- ✓ Coping with change
- ✓ Process improvement

THE SOFTWARE PROCESS

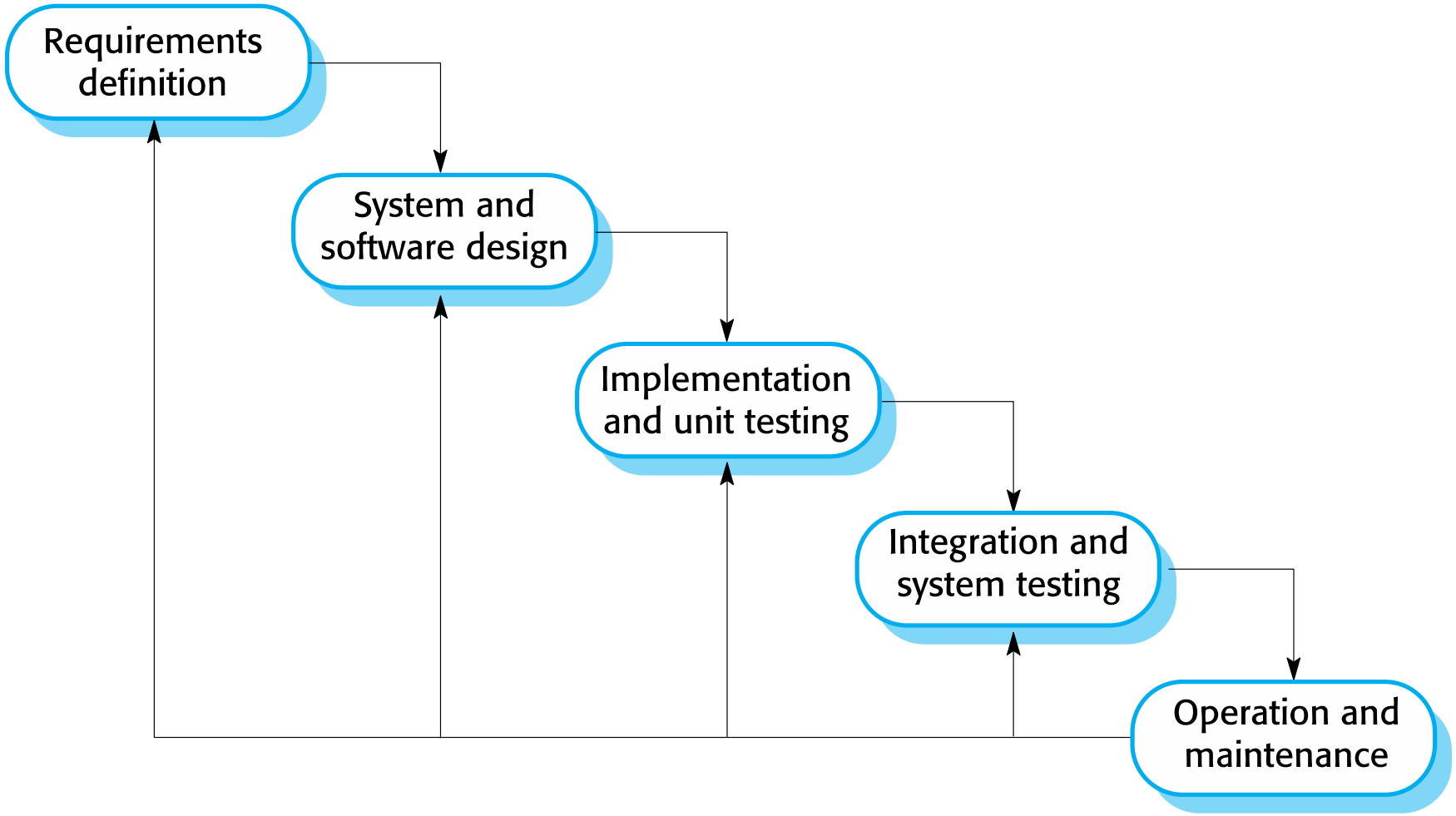
- ✓ A structured set of activities required to develop a software system.
- ✓ Many different software processes but all involve:
  - Specification
  - Design and implementation
  - Validation
  - Evolution.
- ✓ A software process model
  - an abstract representation of a process

SOME SOFTWARE PROCESS MODELS

- The waterfall model
  - Plan-driven model. model theo ke hoach?
  - Separate and distinct phases of specification and development.
- Incremental development Phát trin gia tng
  - Specification, development and validation are interleaved. Xen vào
  - May be plan-driven or agile.
- Integration and configuration tich hop va cau hinh
  - The system is assembled from existing configurable 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.

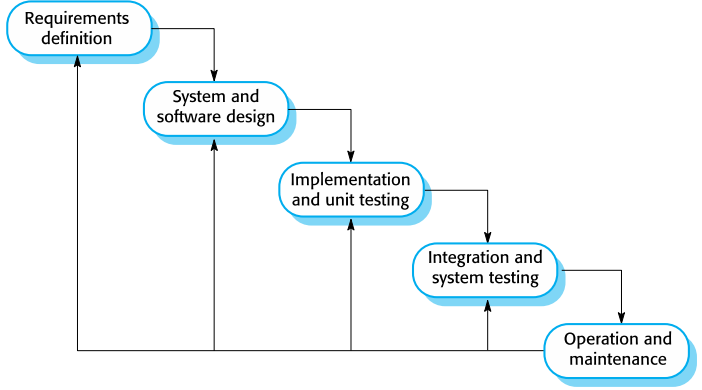
Bang cach su dung mot chuong trinh ket hop voi cac yeu tu nhung model tren

THE WATERFALL MODEL



In principle, a phase has to be complete before moving onto the next phase.

WATERFALL MODEL USAGES



Nhc im ln nht

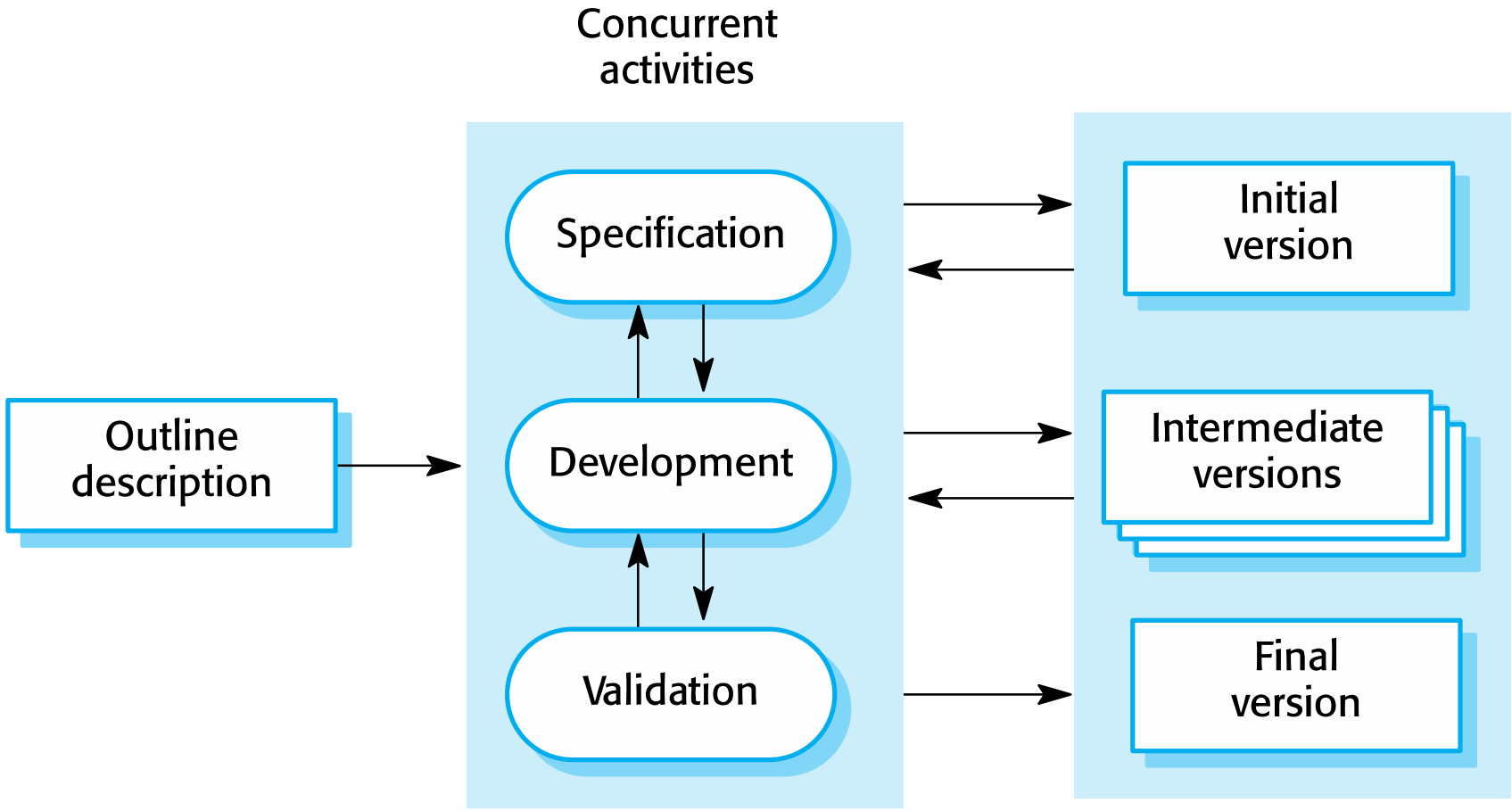
- ✓ The **main drawback:**
  - the difficulty of **accommodating change after the process is underway.**

dang dien ra

- ✓ Mostly used for **large systems** engineering projects
  - a system is developed at several sites.
  - the plan-driven nature of the waterfall model helps coordinate the work.
- ✓ When the requirements are well-understood and **changes** will be **fairly limited during the design process.**
  - Few business systems have stable requirements.

INCREMENTAL DEVELOPMENT

Hoat dong dong thoi



INCREMENTAL DEVELOPMENT BENEFITS

- ✓ Reduce the cost of accommodating changing customer requirements
- ✓ Easier to get customer feedback on the development work that has been done.
- ✓ More rapid delivery and deployment of useful software to the customer

INCREMENTAL DEVELOPMENT PROBLEMS

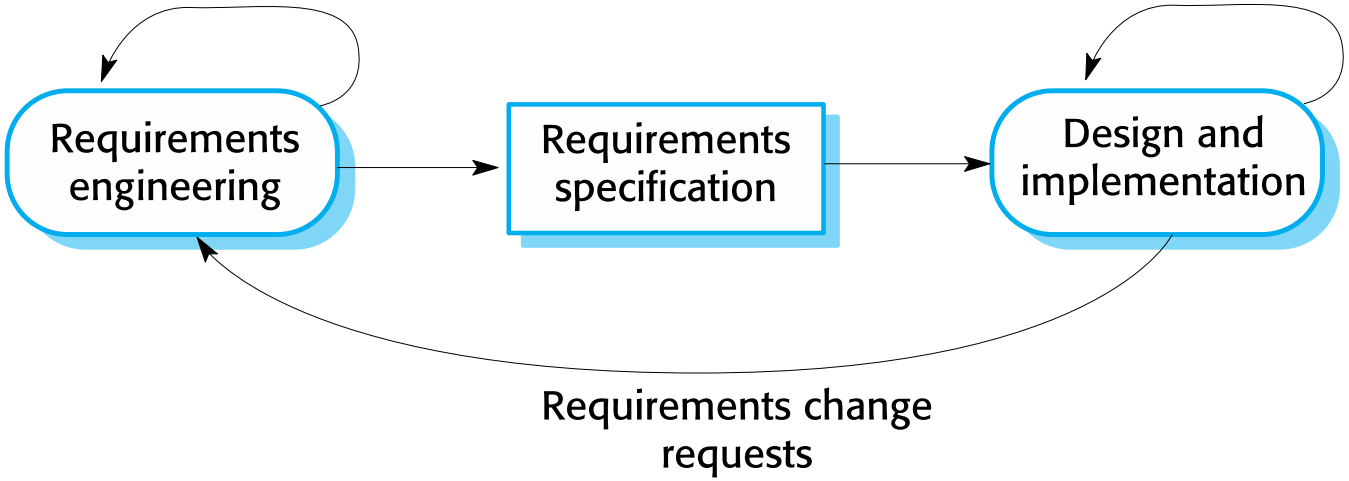
- ✓ The process is not visible.
  - Managers need regular deliverables
  - Not cost-effective to produce documents for every product version

- ✓ System structure tends to degrade as new increments are added.
  - Need time and money on refactoring to improve the software
  - Regular change tends to corrupt the structure.
  - Incorporating further software changes becomes increasingly difficult and costly.

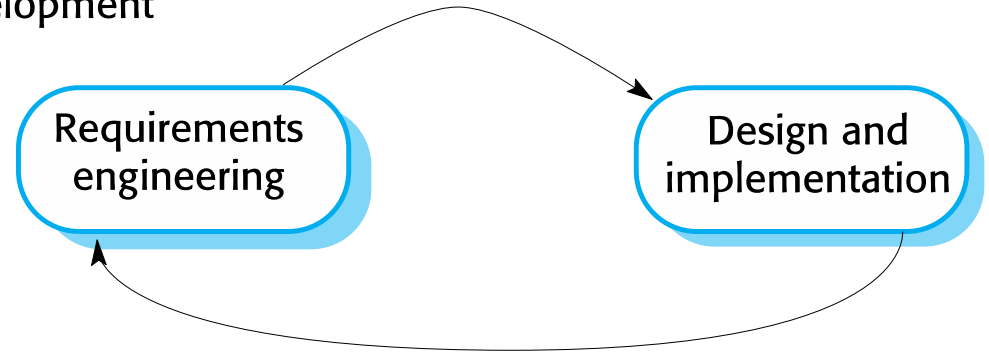
AGILE DEVELOPMENT

Plan-based development

i.e.: waterfall model, incremental development



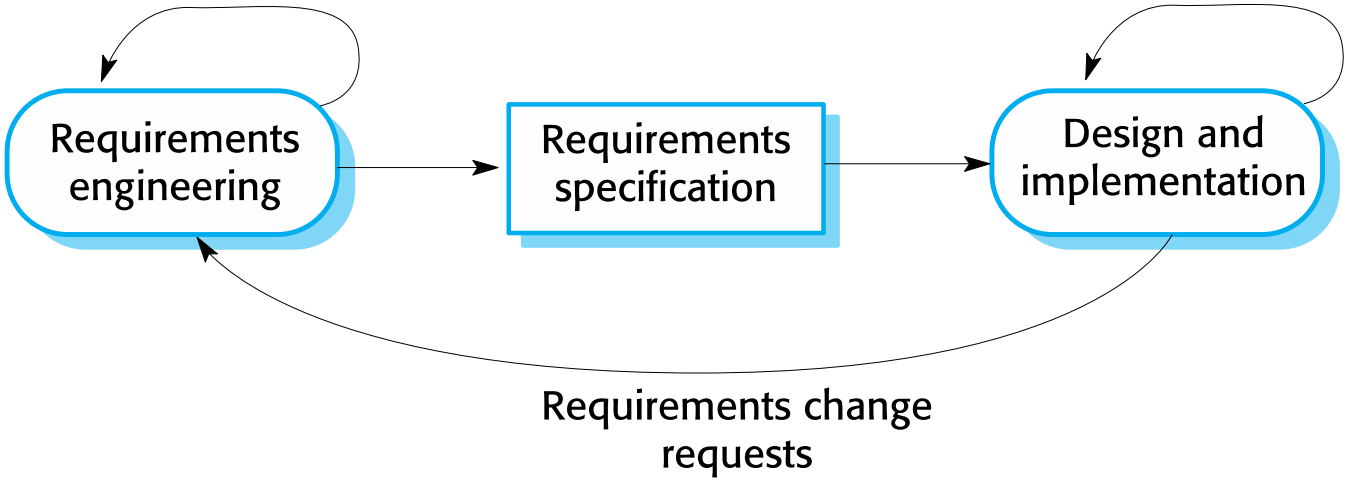
Agile development



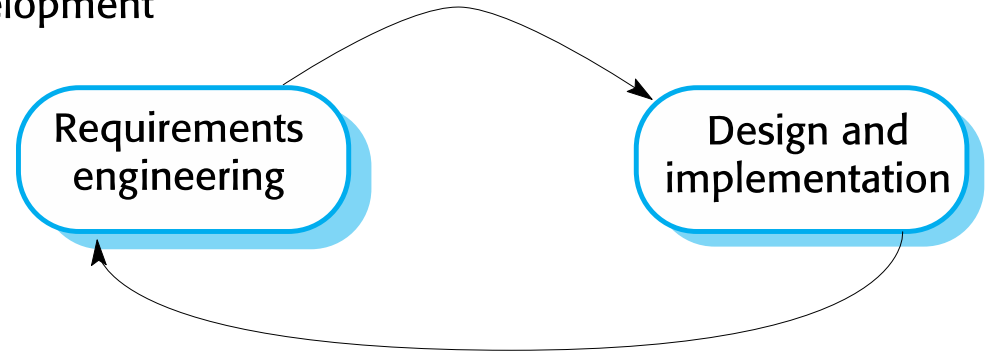
AGILE DEVELOPMENT

Plan-based development

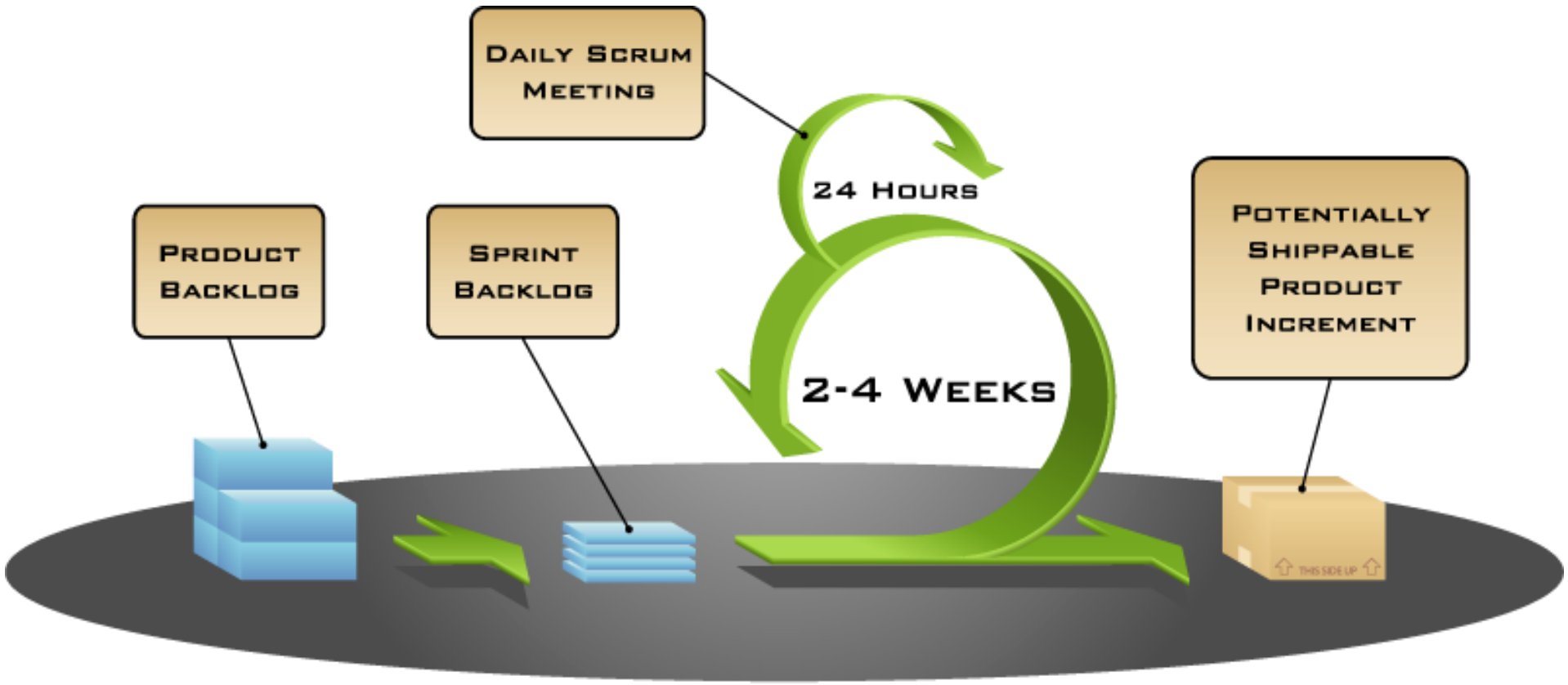
i.e.: waterfall model, incremental development



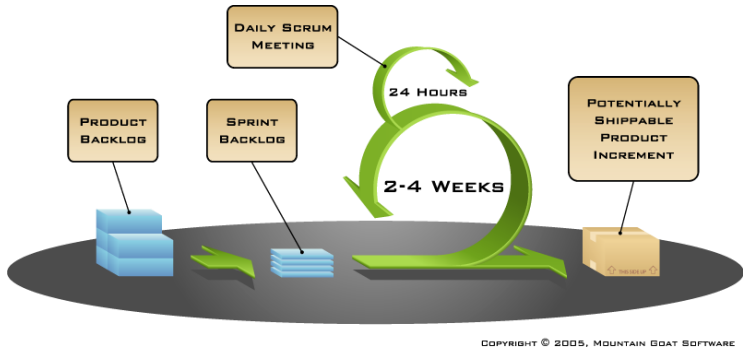
Agile development



SCRUM – THE MOST POPULAR AGILE DEVELOPMENT APPROACHES



AGILE DEVELOPMENT



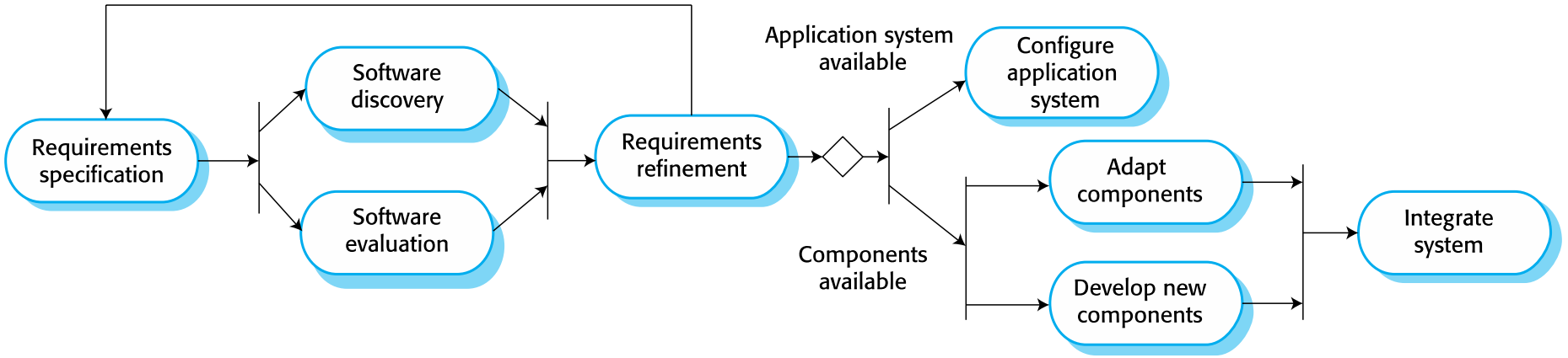
PROs	**CONs**
More flexible	Hard to predict
Product get to market faster	Final product is not released first
Better communication	Documentation gets left behind

REUSE-ORIENTED SOFTWARE ENGINEERING

he thong duoc tich hop tu nhung thanh phan hay ung dung hien co

- ✓ Based on software reuse where systems are integrated from existing components or application systems (COTS - Commercial-off-the-shelf) systems).
  - Stand-alone application systems (COTS)
  - Package objects / component framework such as .NET or J2EE.
  - Web services
- ✓ Reused elements may be configured to adapt their behaviour and functionality to a user's requirements
- ✓ Reuse is now the standard approach for building many types of business system

REUSE-ORIENTED SOFTWARE ENGINEERING



ADVANTAGES AND DISADVANTAGES

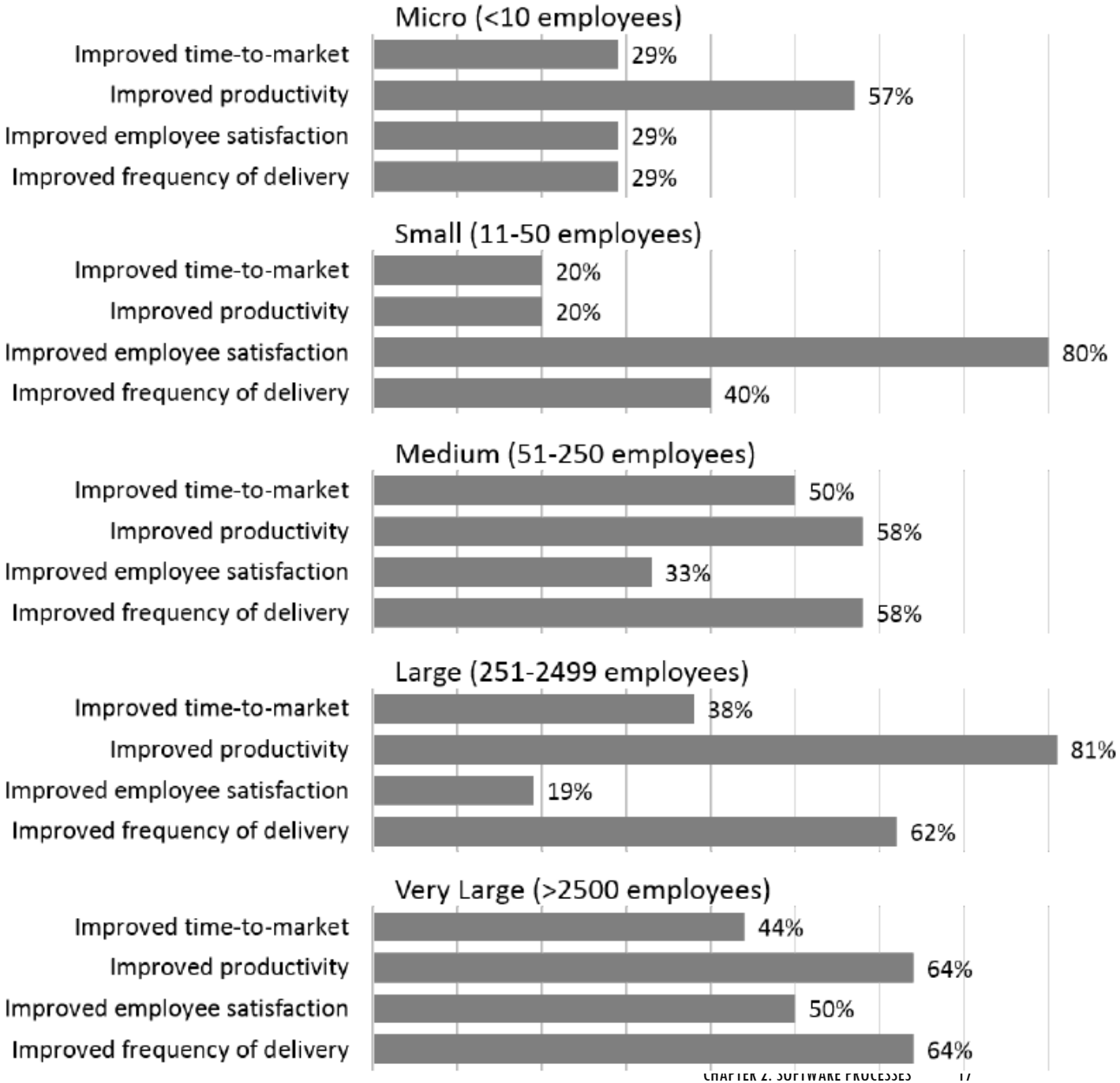
- ✓ Reduced costs and risks as less software is developed from scratch
- ✓ Faster delivery and deployment of system
- ✓ But requirements compromises are inevitable so system may not meet real needs of users

khong the tranh khoi

- ✓ Loss of control over evolution of reused system elements

"HYBRID DEVELOPI APPROACHES IN S SYSTEMS DEVELOP

Many companies face the pro development approach fitting



Kuhrmann, M., P. Diebold, J.
"Hybrid Software Developme
Software 36 (4): 20–31. https

PROCESS ACTIVITIES

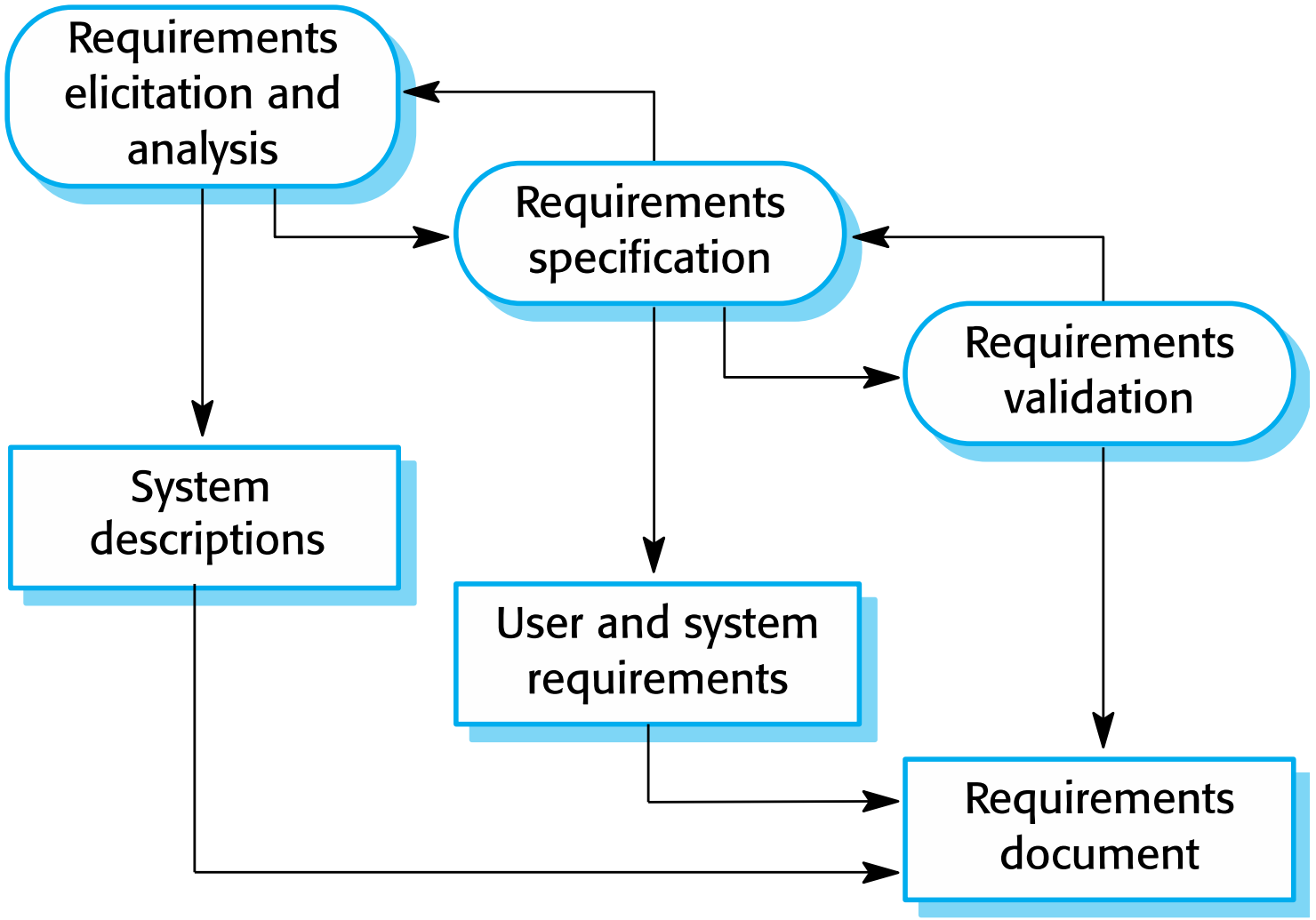
ACTIVITY: SOFTWARE SPECIFICATION

- ✓ The process of establishing what services are required and the constraints on the system's operation and development.

  Qua trinh cung co nhung gi dich vu yeu cau va nhung han che trong cac hoat dong va phat trien cua he thong

- ✓ Use: Requirements engineering process
  - Requirements elicitation and analysis thu thap yeu cau va phan tich
  - Requirements specification dac ta
  - Requirements validation xac thuc

THE REQUIREMENTS ENGINEERING PROCESS



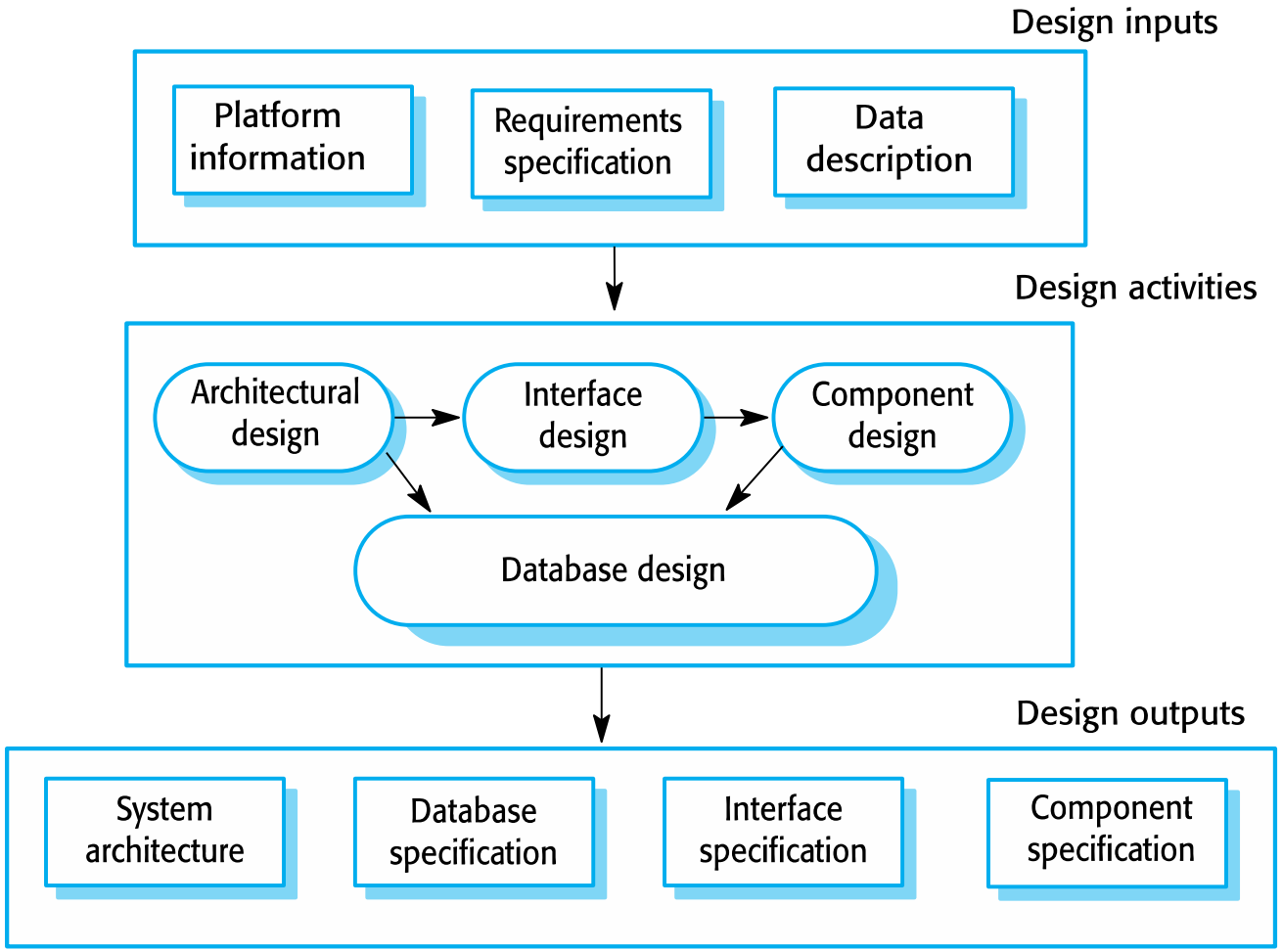
ACTIVITY: SOFTWARE DESIGN AND IMPLEMENTATION ~ SOFTWARE DEVELOPMENT

- ✓ The process of converting the system specification into an executable system.

  Qa trinh chuyen doi system specification thanh he thong thuc thi

- ✓ Two (sub) activities:
  - Software design
    - Design a software structure that realises the specification;
  - Implementation
    - Translate this structure into an executable program;
  - The activities of design and implementation are closely related and may be inter-leaved.

A GENERAL MODEL OF THE DESIGN PROCESS



SYSTEM IMPLEMENTATION Trien khai he thong

- ✓ The software is implemented either by developing a program or programs or by configuring an application system.
- ✓ Design and implementation are interleaved activities for most types of software system.
- ✓ Programming is an individual activity with no standard process.
- ✓ Debugging is the activity of finding program faults and correcting these faults.

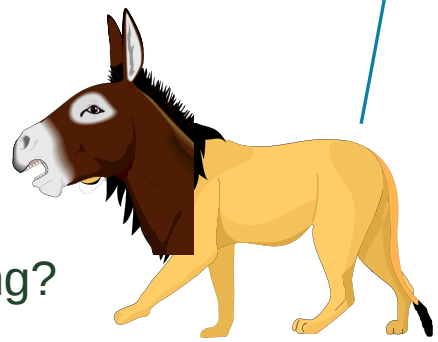
ACTIVITY: SOFTWARE VALIDATION

building the thing right?

- ✓ Verification and validation (V & V)
  - to show that a system conforms to its specification and meets the requirements of the system customer.

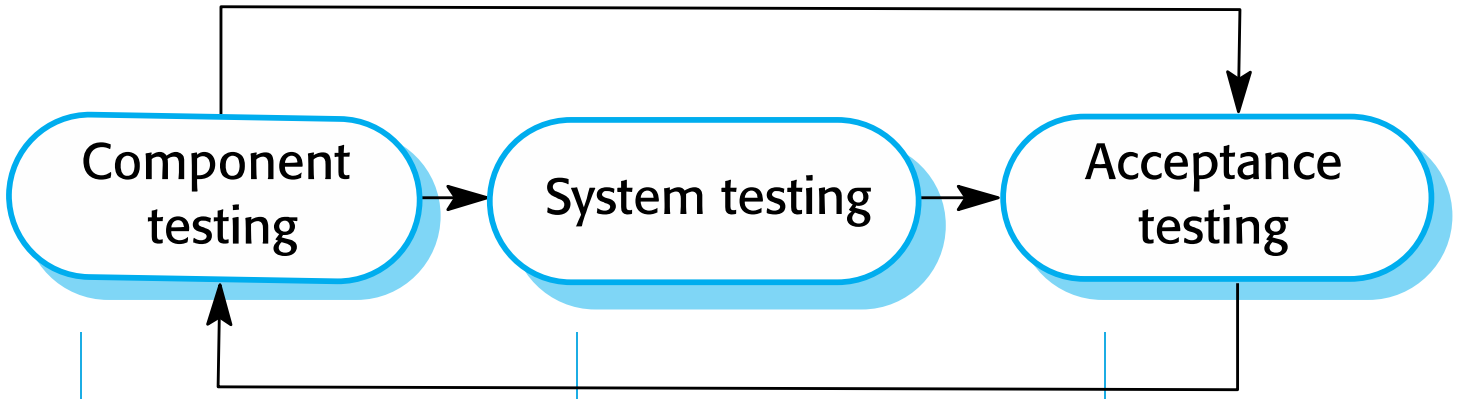
phu hop

- ✓ Involves checking and review processes and system testing.
  - System testing: executing the system with test cases
  - Testing: the most commonly used V & V activity.



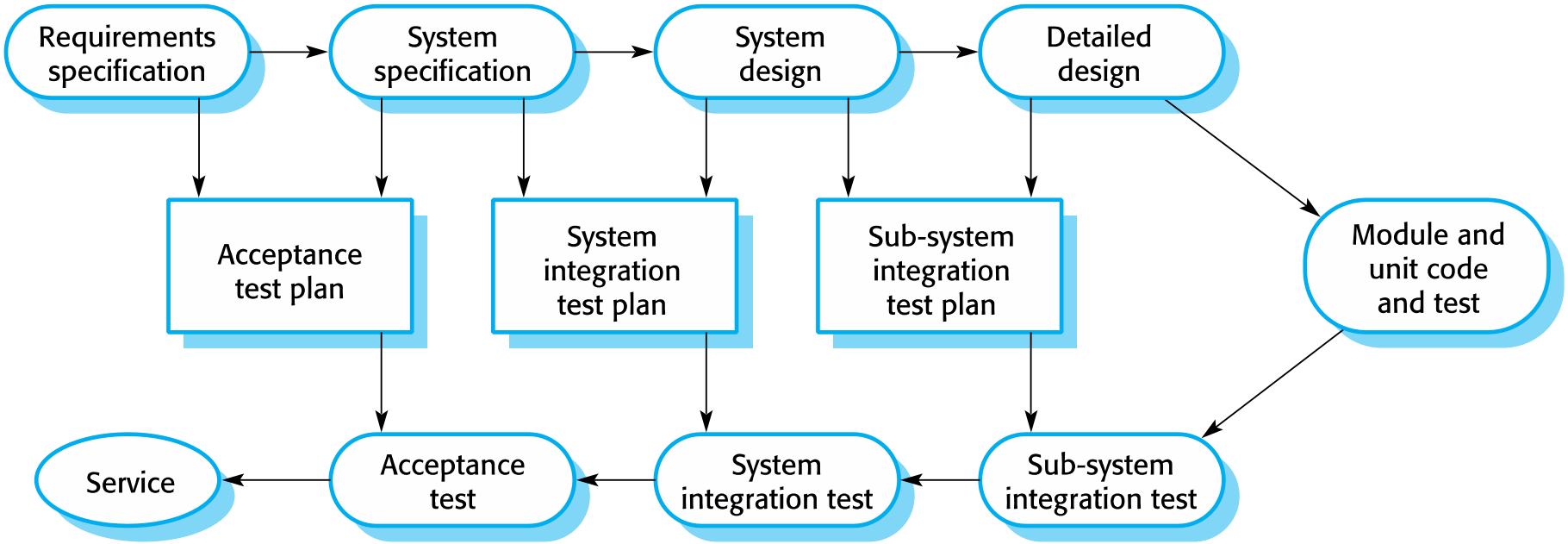
building the right thing?

STAGES OF TESTING



- Test individual components independently
- Testing of the system as a whole
- Testing with customer data

TESTING PHASES IN A PLAN-DRIVEN SOFTWARE PROCESS

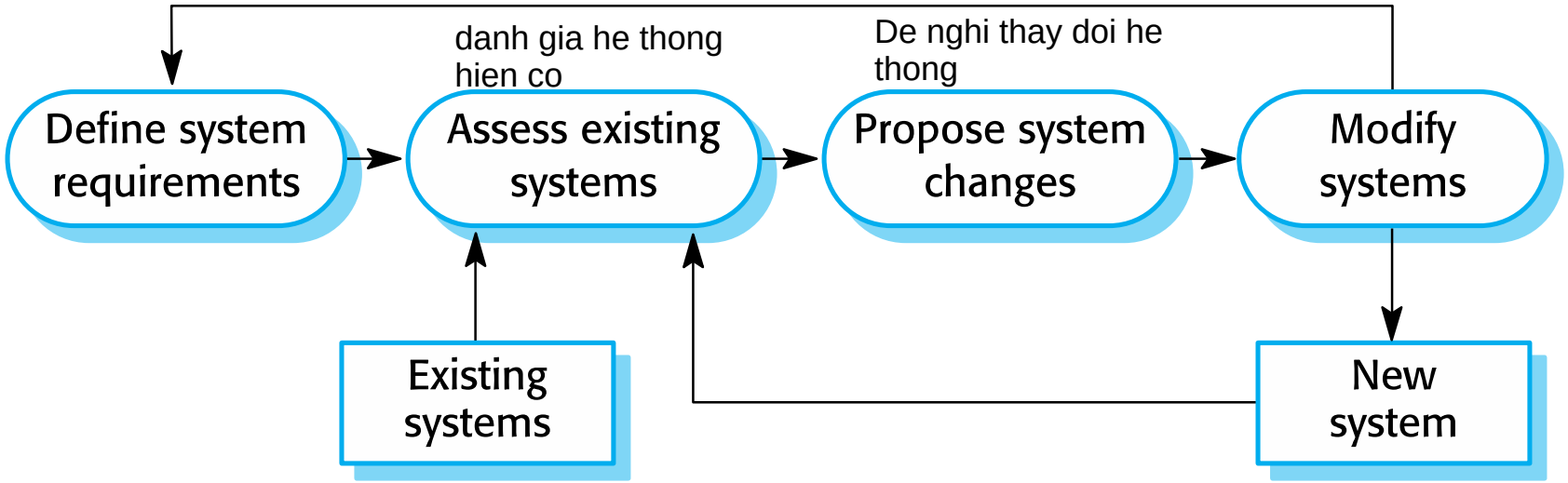


ACTIVITY: SOFTWARE EVOLUTION

von co

- ✓ Software is inherently flexible and can change.
- ✓ Requirements can change
  - (changing business circumstances) => the software must also evolve and change.

truong hop



COPING WITH CHANGE

COPING WITH CHANGE

- ✓ Change is inevitable in all large software projects.
  - Business changes
  - New technologies
  - Changing platforms

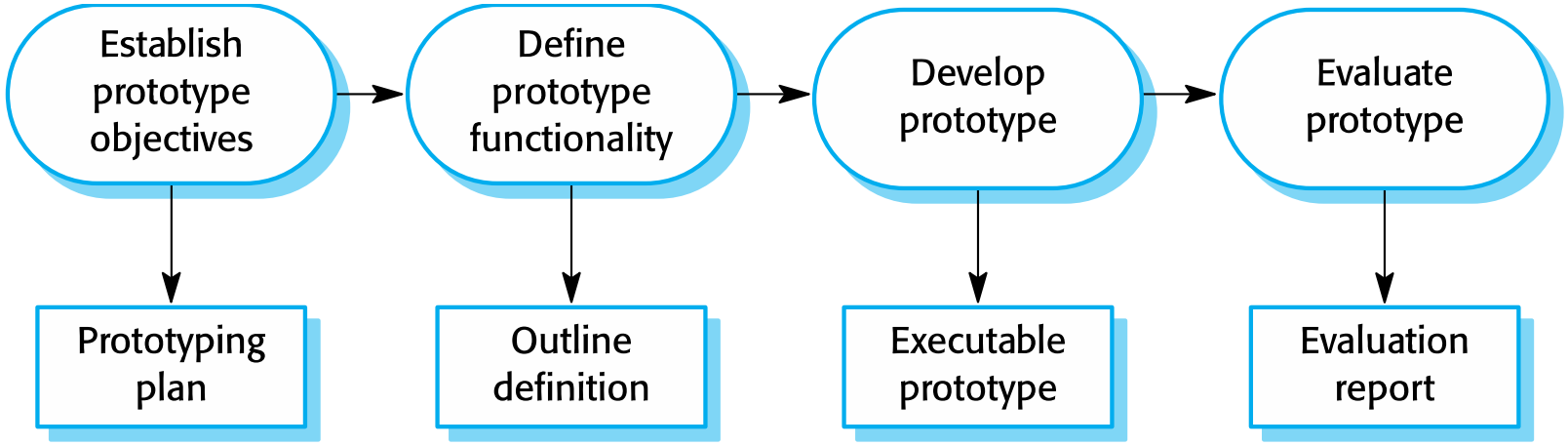
- ✓ Change leads to rework
  - costs include rework (re-analysing requirements) and implementing new functionality

SOFTWARE PROTOTYPING nguyen mau

- ✓ A prototype is an initial version of a system used to demonstrate concepts and try out design options.
- ✓ A prototype can be used in:
  - requirements engineering process: requirements elicitation and validation;
  - design processes: options and develop UI design;
  - testing process: run back-to-back tests.

Benefits:
- Improved system usability.
- A closer match to users' real needs.
- Improved design quality.
- Improved maintainability.
- Reduced development effort.

THE PROCESS OF PROTOTYPE DEVELOPMENT



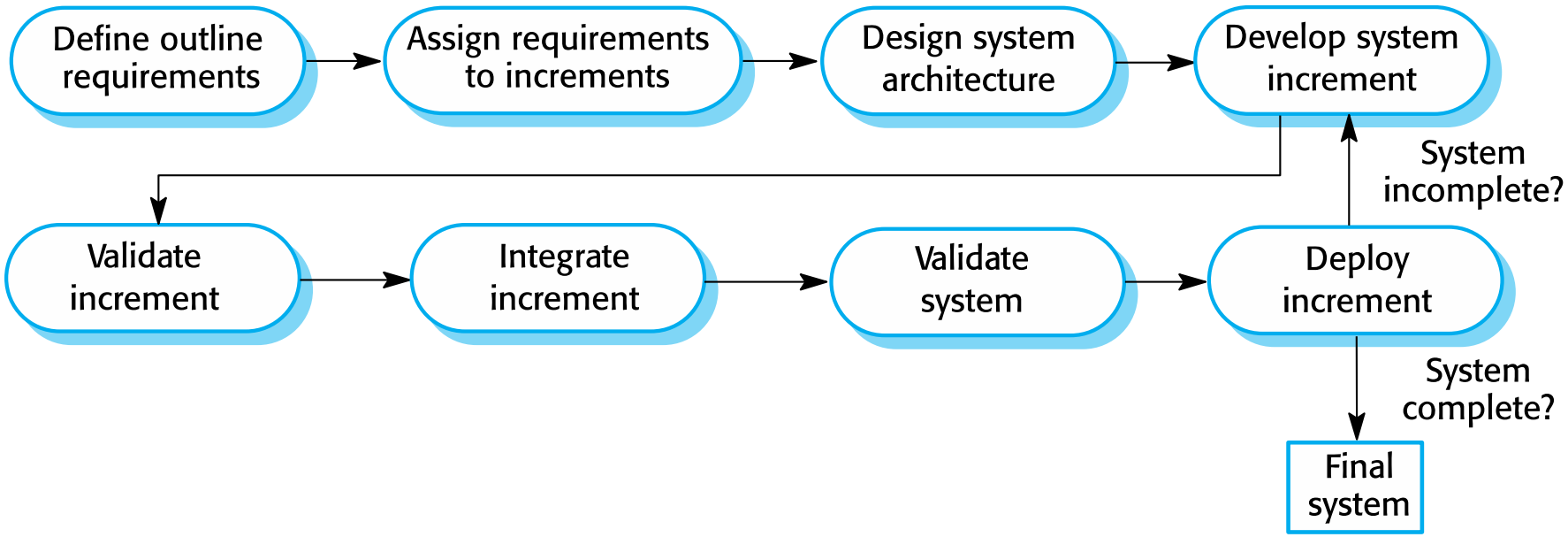
Prototype development:

- May be based on rapid prototyping languages or tools
- May involve leaving out functionality

INCREMENTAL DELIVERY

- ✓ The development and delivery is broken down into increments
  - each increment delivering part of the required functionality.
  - user requirements are prioritised and the highest priority requirements are included in early increments.
- ✓ Two approaches:
  - Incremental development: by developer
  - Incremental delivery: for end-user

INCREMENTAL DELIVERY



Advantages:

- system functionality is available earlier.
- early increments act as a prototype
- lower risk of overall project failure.
- highest priority system services receive most testing.

Problems:

- may require a set of basic facilities
- the specification is developed in conjunction with the software.

giao thoa

A DISCUSSION CASE

https://docs.google.com/document/d/1S05m-iNqgjqhPTAkPJM6FfoXdnmv6EO4q_drPjeXqTA/edit?usp=sharing

PROCESS IMPROVEMENT

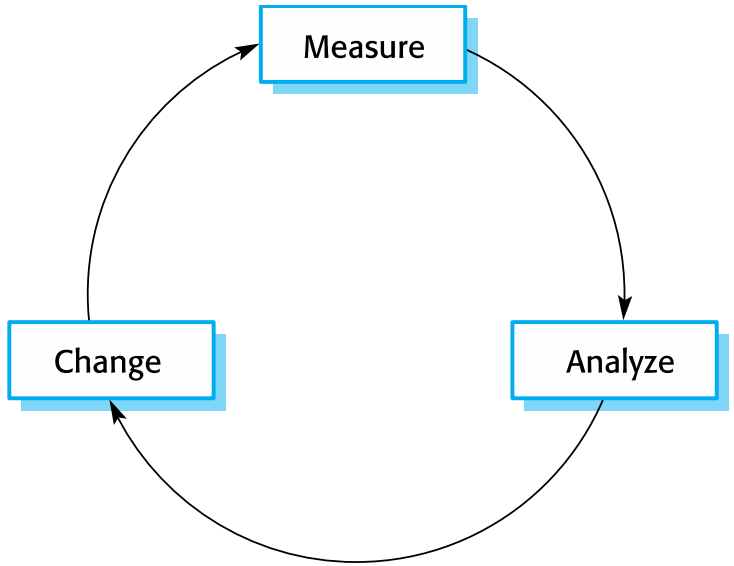
PROCESS IMPROVEMENT

- ✓ Software process improvement
  - enhancing the quality of software,
  - reducing costs
  - or accelerating development processes.

tang cuong

day nhanh

- ✓ Process improvement
  - understanding existing processes
  - and changing these processes



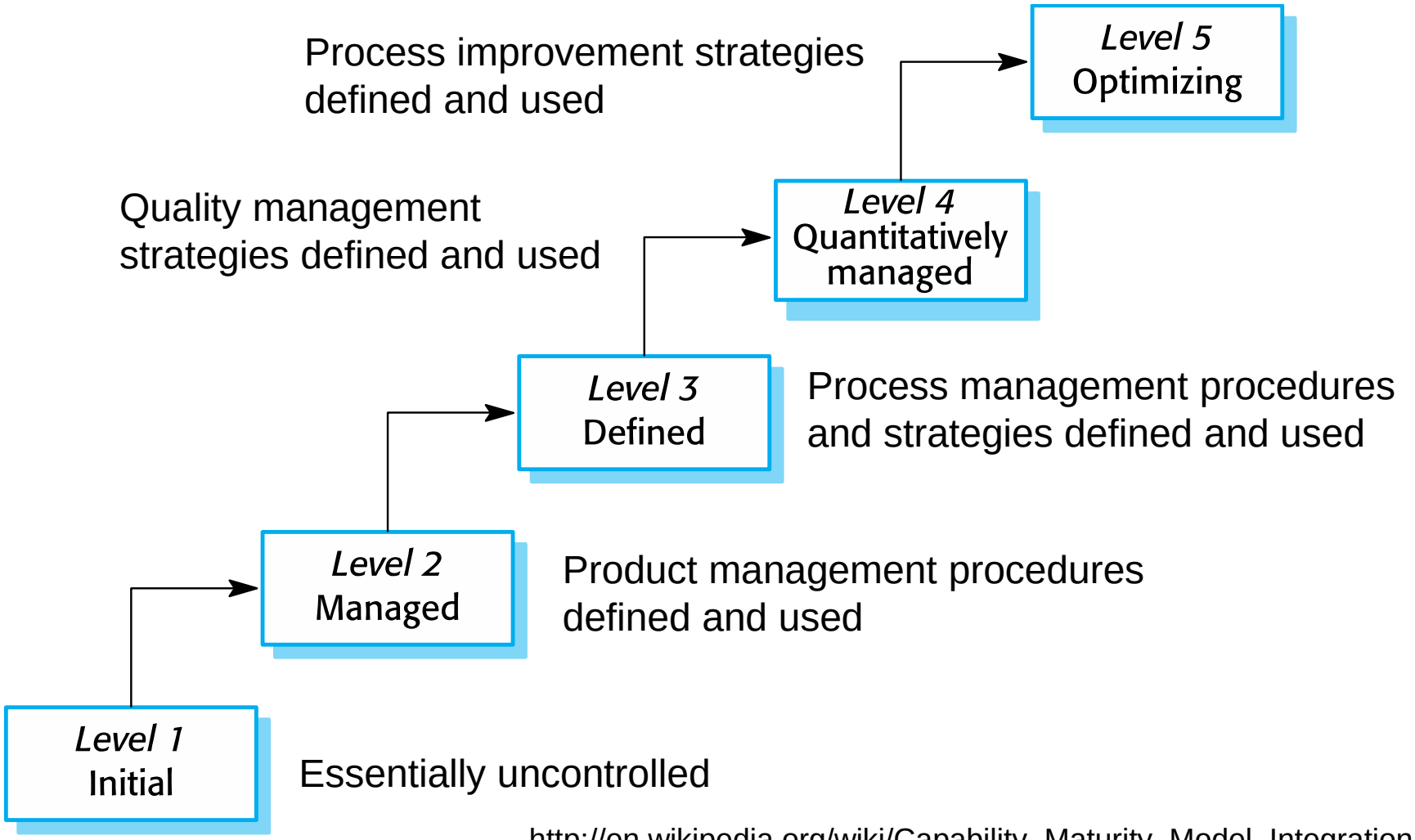
PROCESS IMPROVEMENT ACTIVITIES

- ✓ **Process measurement** do luong qa trinh
  - You measure one or more attributes of the software process or product. These measurements forms a baseline that helps you decide if process improvements have been effective. baseline = duong co so
- ✓ **Process analysis** phan tich qa trinh
  - The current process is assessed, and process weaknesses and bottlenecks are identified. Process models (sometimes called process maps) that describe the process may be developed.
- ✓ **Process change** thay doi qua trinh proposed = de xuat
  - Process changes are proposed to address some of the identified process weaknesses. These are introduced and the cycle resumes to collect data about the effectiveness of the changes. chu trinh thu thap

THE CAPABILITY MATURITY MODEL (CMM)

- ✓ Capability Maturity Model Integration (CMMI) is a process level improvement training and appraisal program chuong trinh tham dinh
- ✓ CMMI defines the most important elements that are required to build great products, or deliver great service
- ✓ It is required by many U.S. Government contracts, especially in software development.

THE CAPABILITY MATURITY MODEL (CMM)



http://en.wikipedia.org/wiki/Capability_Maturity_Model_Integration

SOFTWARE PROJECT DOCUMENTATION

Activity	Document
Validation & Verification	**SVVP** - Software Validation & Verification Plan
Quality Assurance	**SQAP** - Software Quality Assurance Plan
Configuration	**SCMP** - Software Configuration Management Plan
Project status	**SPMP** - Software Project Management Plan
Requirements	**SRS** - Software Requirements Specifications
Design	**SDD** - Software Design Document / Software Detail Design Document
Code	Source **Code**
Testing	**STD** - Software Test Document
Operation	User's **Manual**

SUMMARY

- ✓ Software processes
- ✓ Software process models
  - waterfall, incremental development, reuse-oriented development.
- ✓ Fundamental activities:
  - Requirements engineering: developing specification.
  - Design and implementation: transforming a requirements specification into an executable software system
  - Software validation: checking that the system conforms to its specification.
  - Software evolution: change existing software systems to meet new requirements

SUMMARY (CONT.)

- ✓ Coping with change
  - prototyping
  - iterative development and delivery
- ✓ Process improvement
  - agile approaches, geared to reducing process overheads,
  - maturity-based approaches based on better process management
  - and the use of good software engineering practice.
- ✓ The SEI process maturity framework (CMM)
  - identifies maturity levels that essentially correspond to the use of good software engineering practice.