

Software Engineering

Lecture 2: Software Development Life Cycle



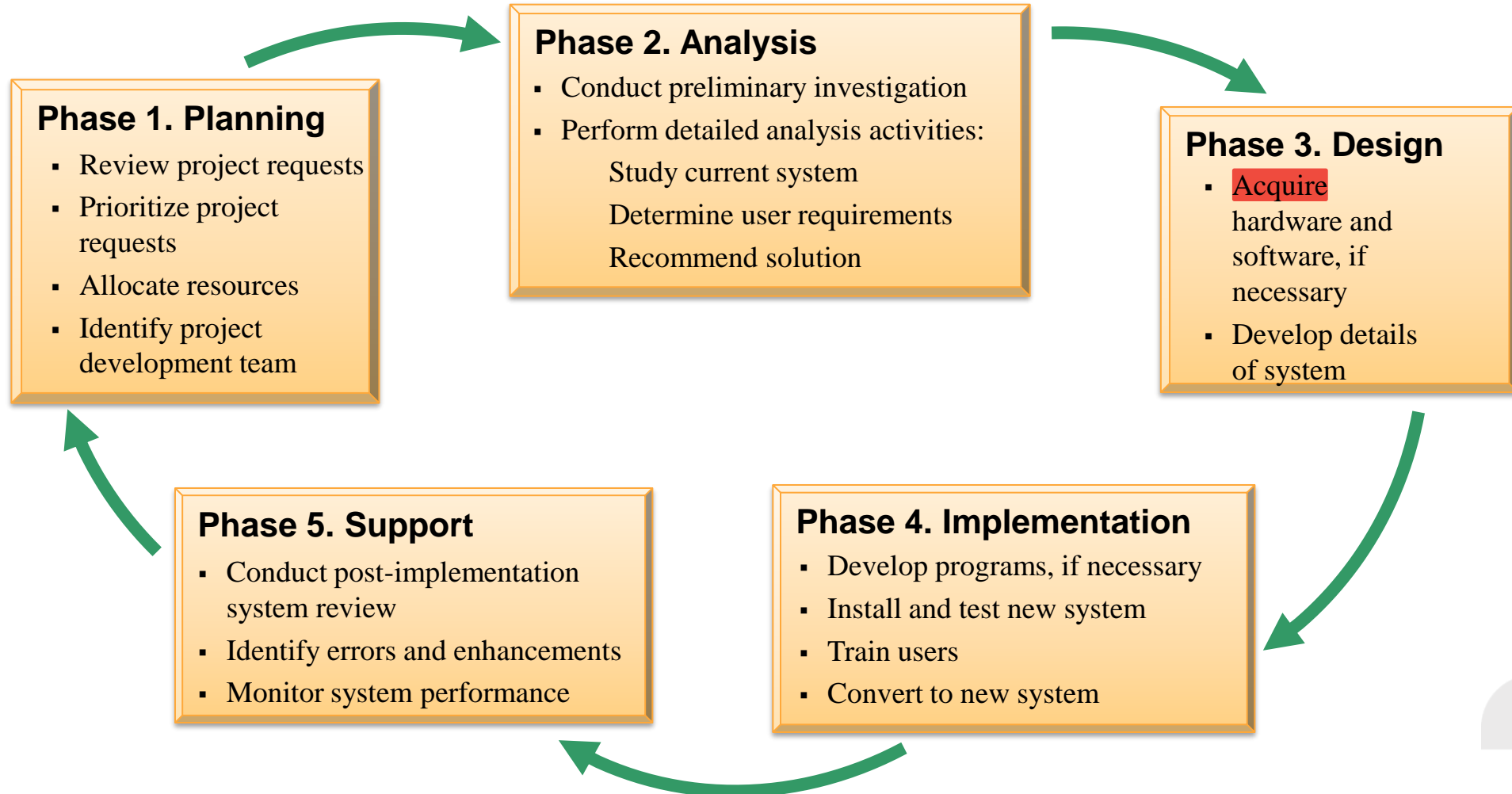
What is an information system (IS)?

Hardware, software, data, people, and procedures that work together to produce quality information

System—Set of components that interact to achieve common goal

Businesses use many types of systems

What are the phases of the system development cycle?



الحصول على



What are **guidelines** for system development?

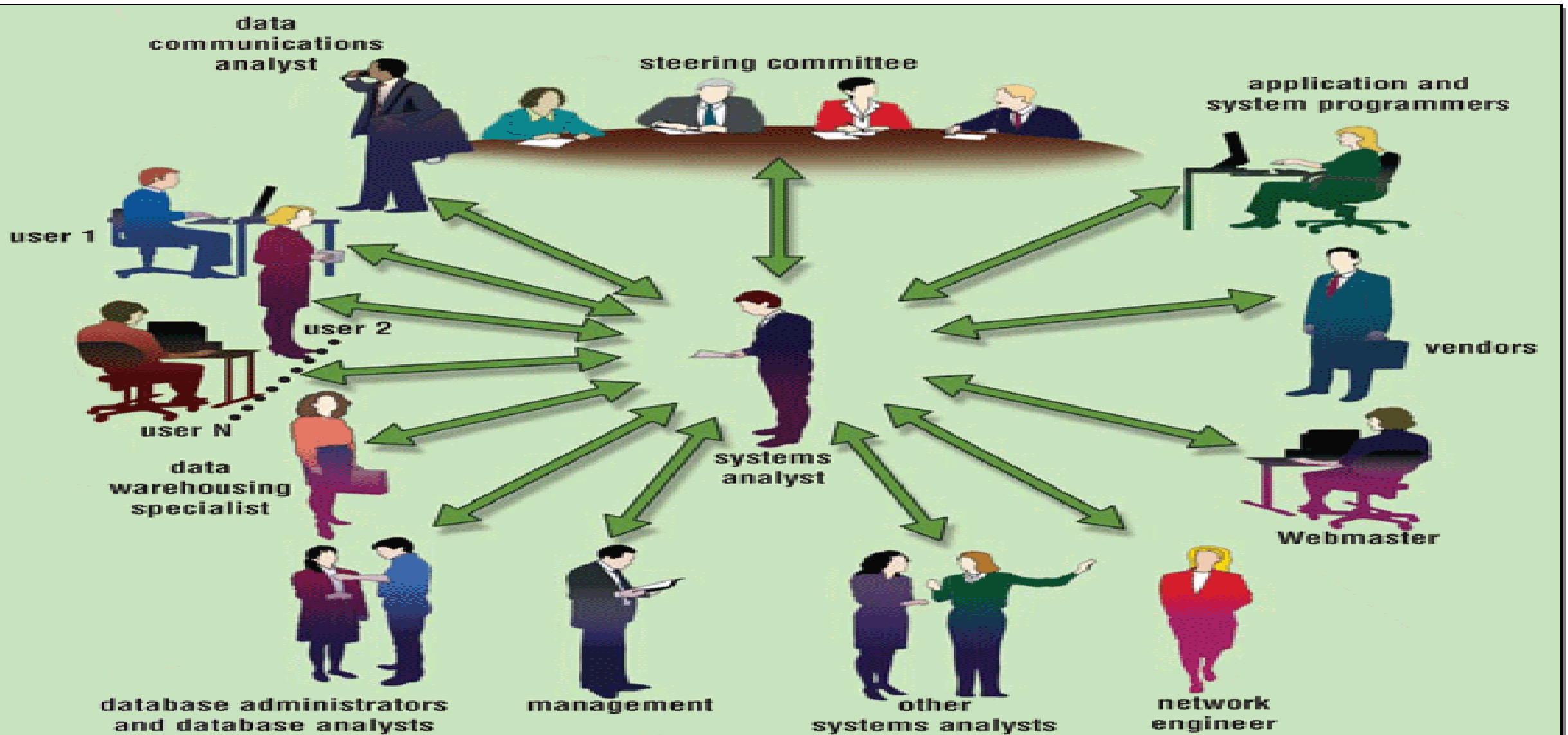
Arrange tasks into **phases**
(**groups of activities**)

Involve **users** (anyone for whom
system is being built)

Develop clearly defined **standards** (procedures
company expects employees to follow)



Who participates in the system development life cycle?



What is a systems analyst?

Responsible for designing and developing information system

Communicate between users and IT professionals



What is the project team?

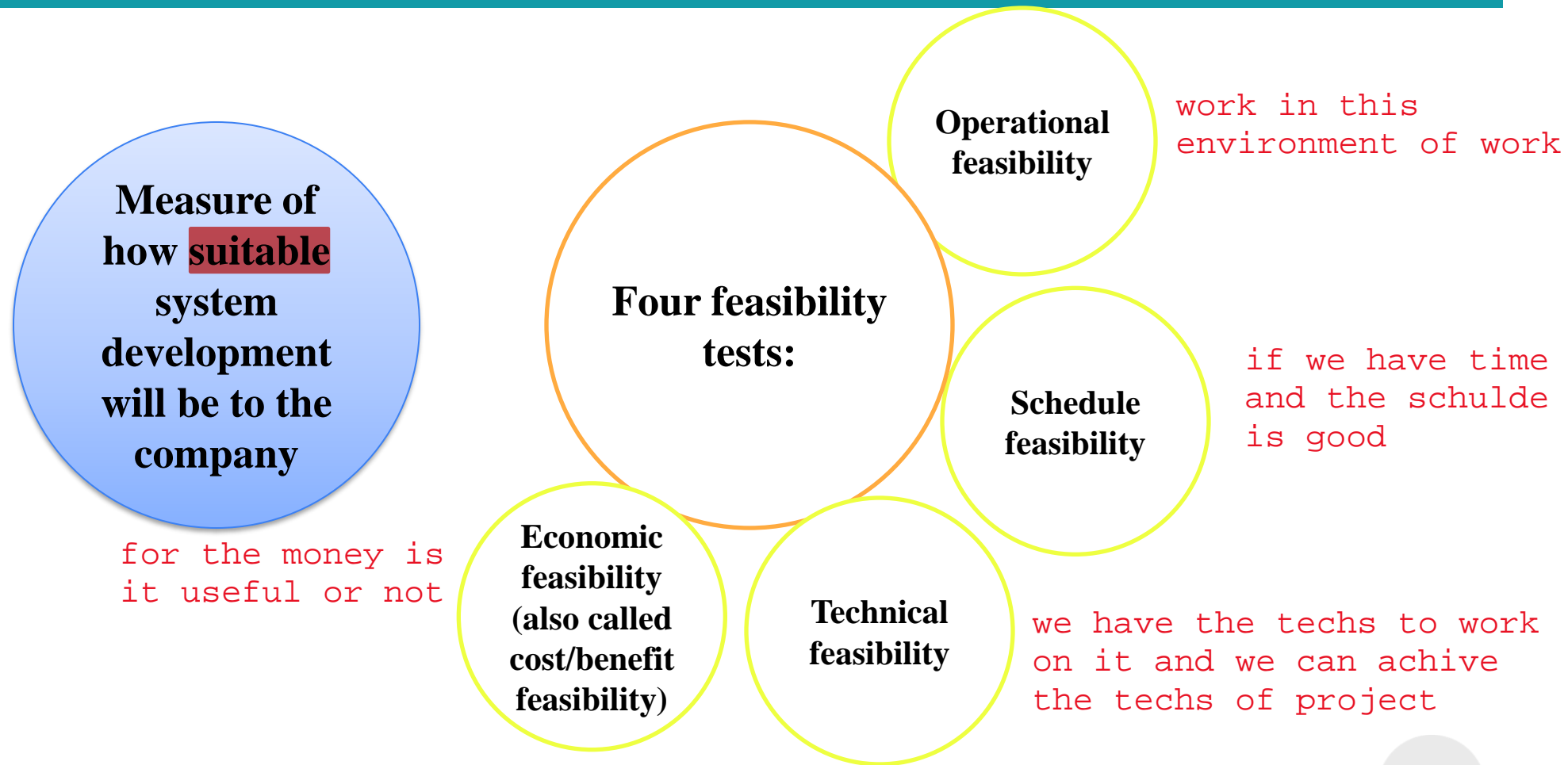
Formed to work on project from beginning to end

Consists of users, systems analyst, and other IT professionals

Project leader—one member of the team who manages and controls project budget and schedule



What is feasibility?



What is documentation?

**Collection and summarization
of data and information**

**Includes reports, diagrams,
programs, and other deliverables**

→ مواد



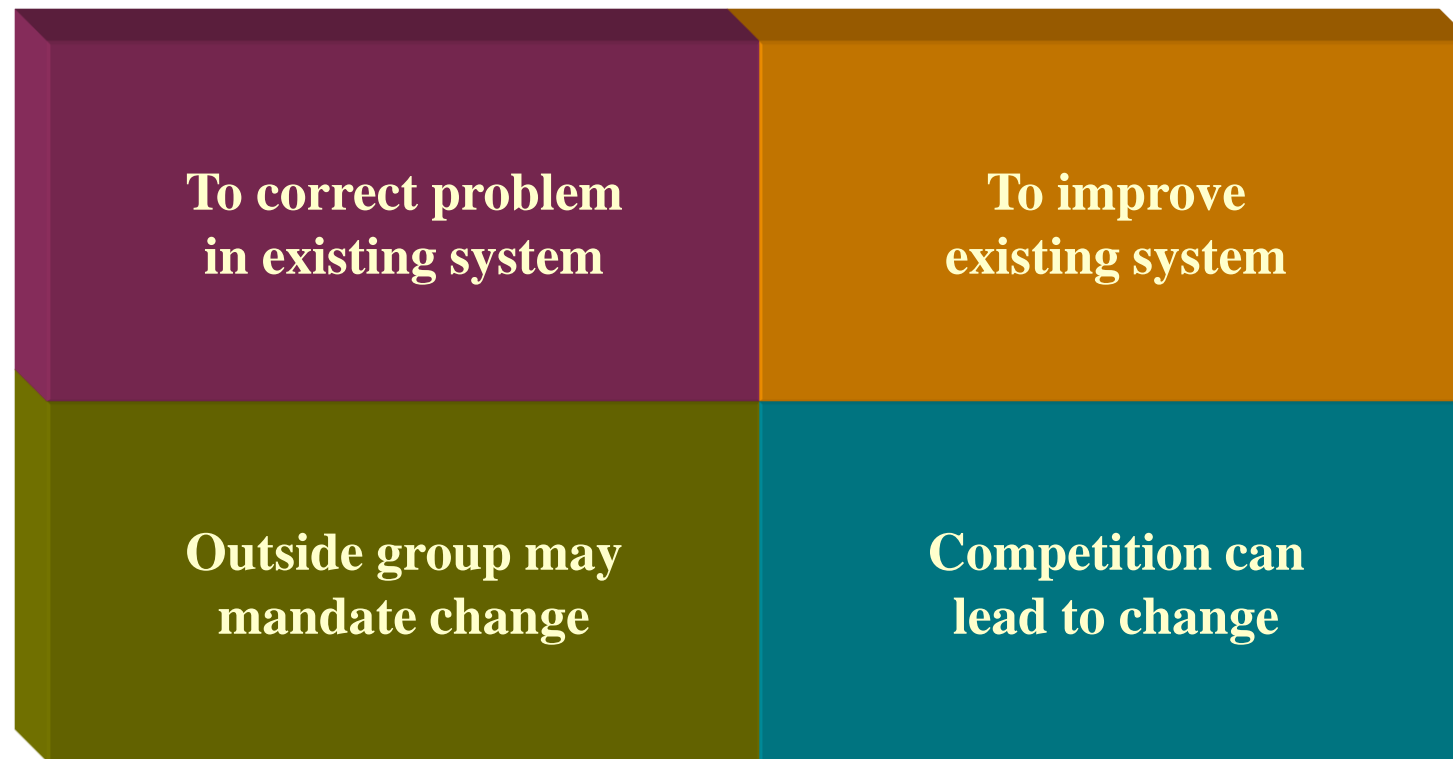
What are six data and information gathering techniques?

جمع

- **Review documentation**
- **Observe** الملاحظة
- **Questionnaire** استبيان
- **Interview**
- **Joint-application design (JAD) session**
- **Research**



What are some reasons to create or modify an information system?

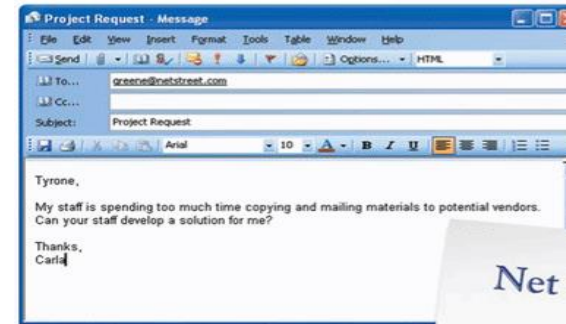


for changing rules
or requ of
customer



What is a request for **system services**?

- **Formal request for new or modified information system**
- Also called **project request**



Net Street Café Form IT-102A
REQUEST FOR SYSTEM SERVICES

SUBMITTED BY: Carla Rosatti
DEPARTMENT: Marketing
DATE: 11-12-2005

TYPE OF REQUEST:
☐ New System
☒ Existing System Enhancement
☐ Existing System Modification

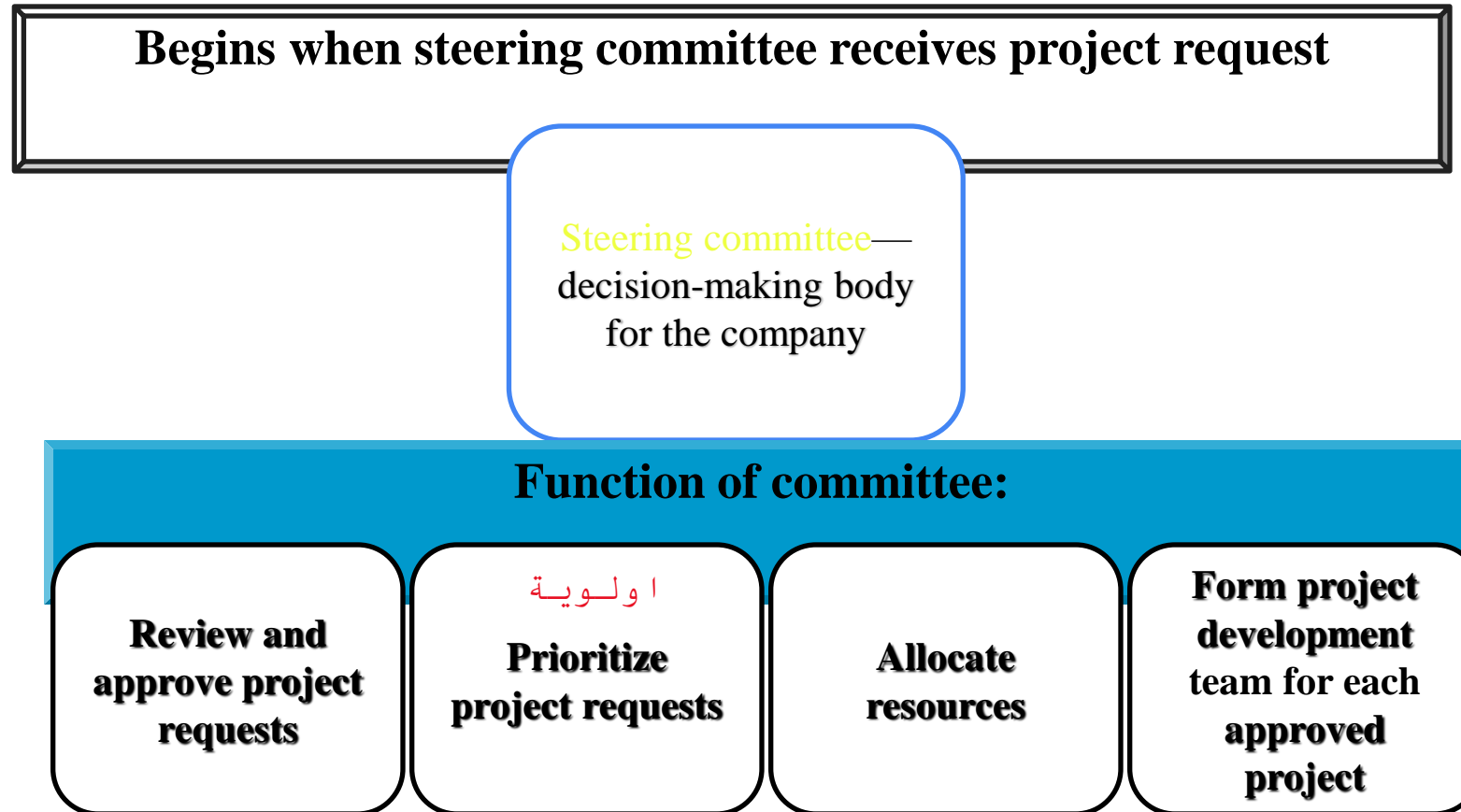
BRIEF STATEMENT OF PROBLEM
The staff in the marketing department is spending too much time copying and mailing materials to potential vendors.

BRIEF STATEMENT OF EXPECTED SOLUTION:
Modify our current Web site so it contains a Vendor Information Web link that displays a Vendor Information Web page. From this page, vendors could download all necessary vendor information.

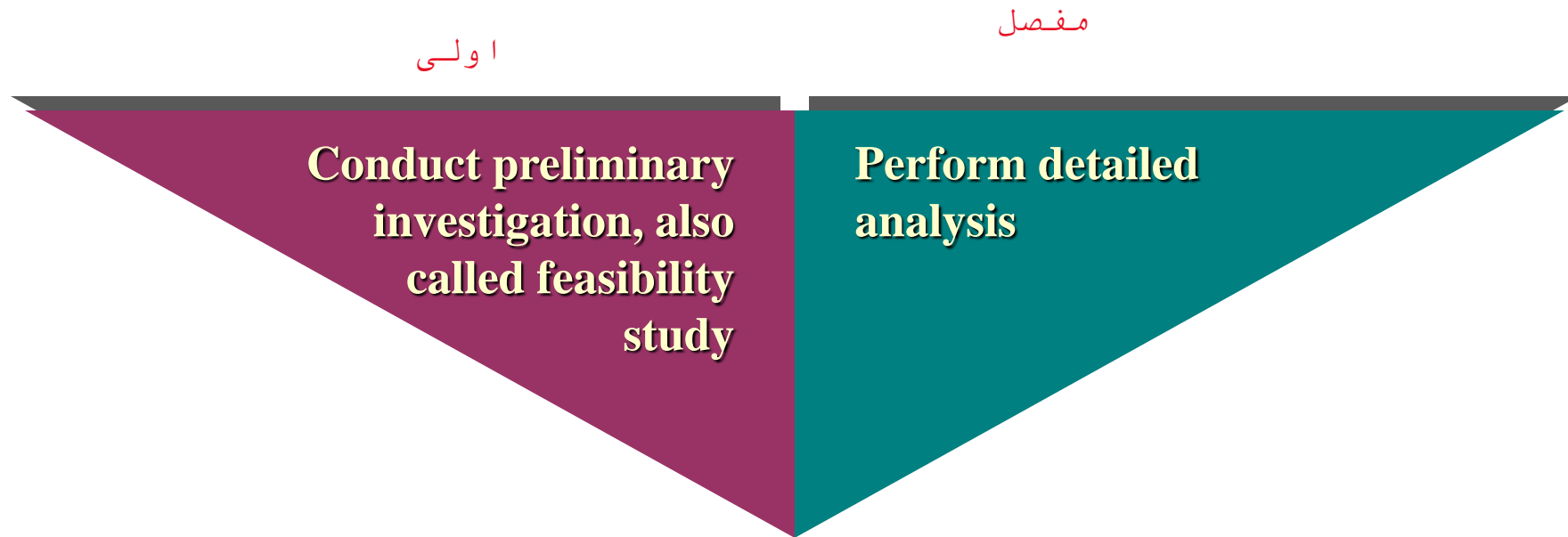
ACTION (To be completed by steering committee member):
☐ Request Approved
☐ Request Delayed
☐ Request Rejected

Analyst Assigned: _____
Start Date: _____
Until: _____
Reason: _____
Signature: _____
Date: _____

What is the planning phase?

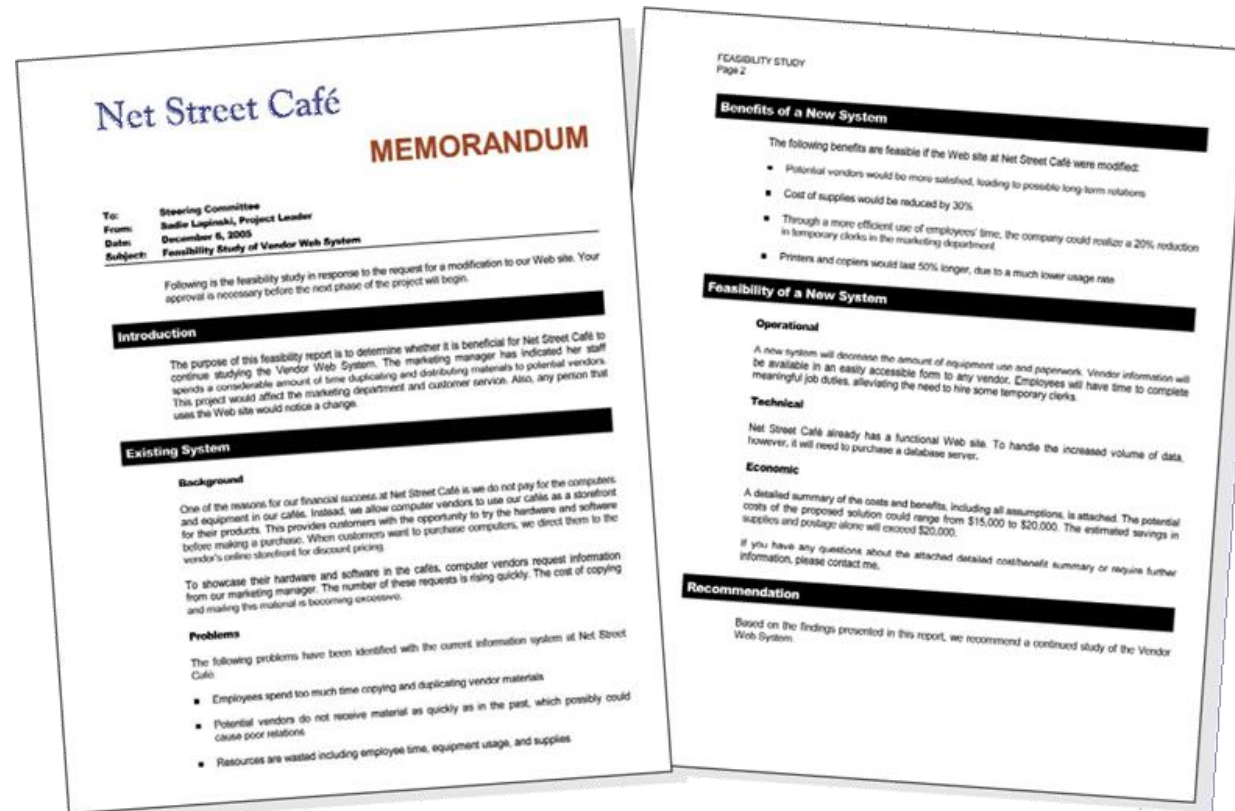


What is the analysis phase?

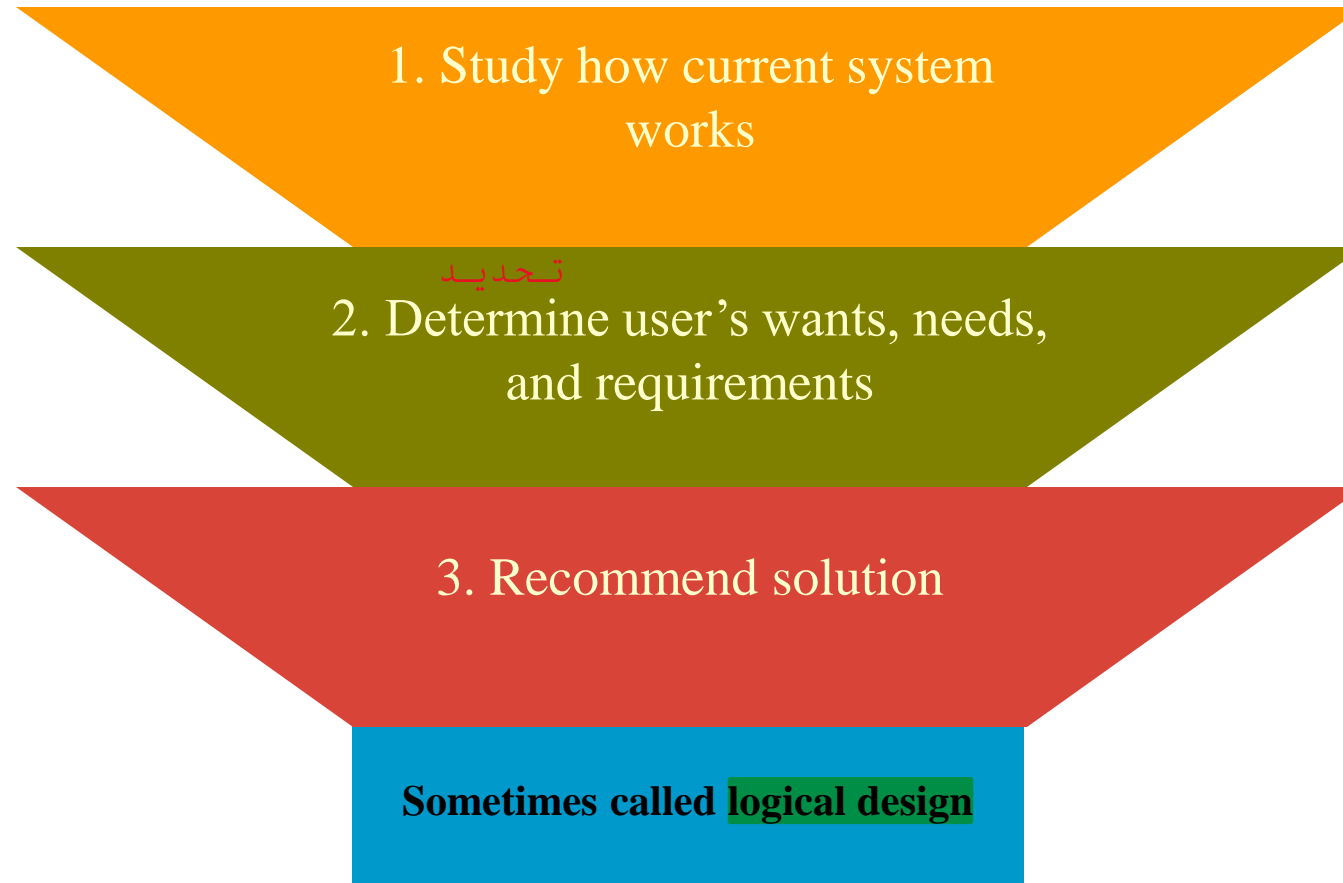


What is the preliminary investigation?

- Determine exact nature of problem or improvement and whether it is **worth pursuing**
 - Findings are presented in feasibility report, also known as a feasibility study



What is detailed analysis?



What is the system proposal?



What are possible solutions?

Buy **packaged software**—prewritten software available for purchase

Horizontal market software—meets needs of many companies

Vertical market software—designed for particular industry

Write own **custom software**—software developed at user's request

Outsource—have outside source develop software



What is the design phase?

Acquire hardware and software

Develop all details of new or
modified information system



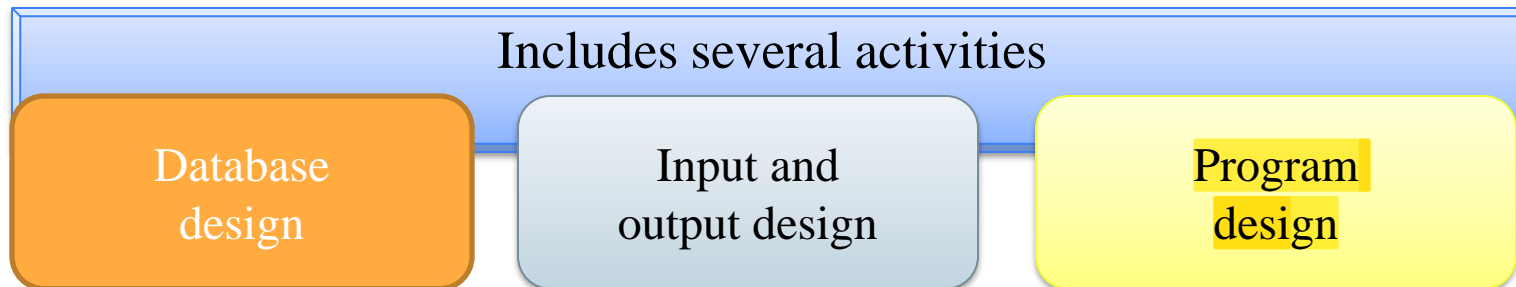
What is needed to acquire new hardware and software?

- **Identify all hardware and software requirements of new or modified system**



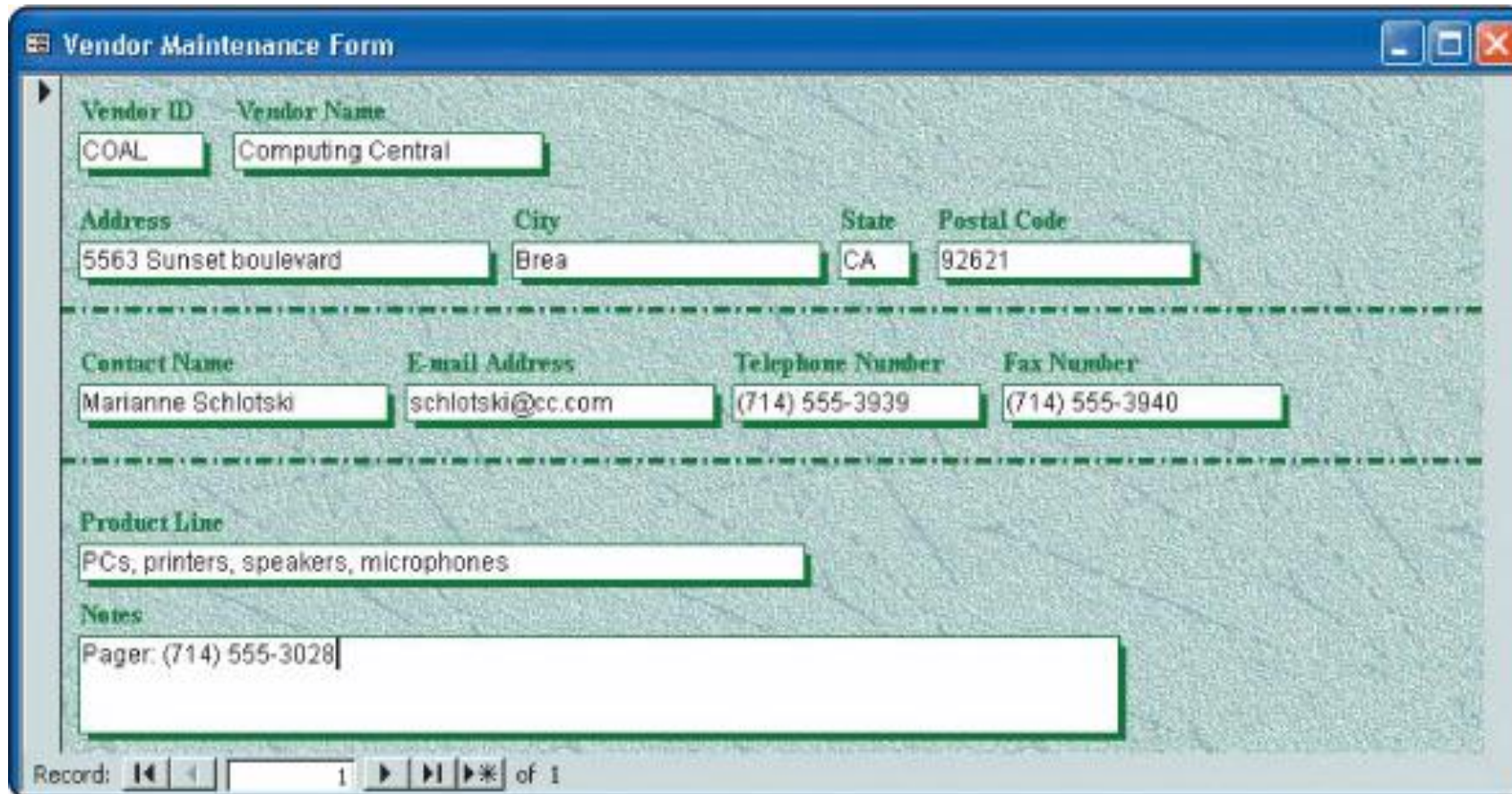
What is a detailed design?

Detailed design specifications for components in proposed solution



What is a **mockup**?

- Sample of input or output that contains **actual data**



The image shows a screenshot of a 'Vendor Maintenance Form' window. The form contains several input fields with the following data:

Vendor ID	Vendor Name	Address	City	State	Postal Code	Contact Name	E-mail Address	Telephone Number	Fax Number	Product Line	Notes
COAL	Computing Central	5563 Sunset boulevard	Brea	CA	92621	Marianne Schlotski	schlotski@cc.com	(714) 555-3939	(714) 555-3940	PCs, printers, speakers, microphones	Pager: (714) 555-3028

At the bottom of the form, there is a record navigation bar showing 'Record: 1 of 1'.



What is a prototype?

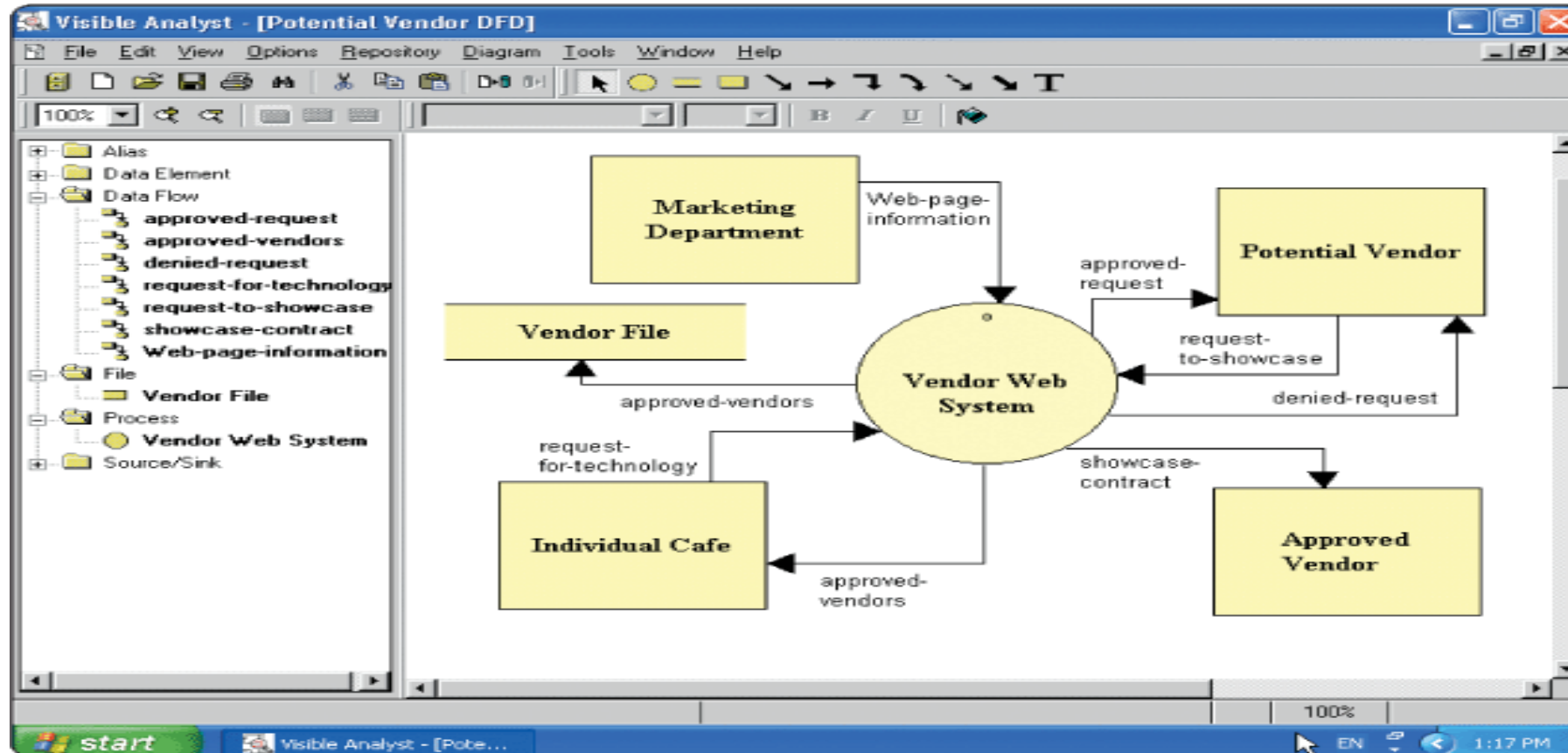
Working model of
proposed system

Beginning a prototype too early
may lead to problems



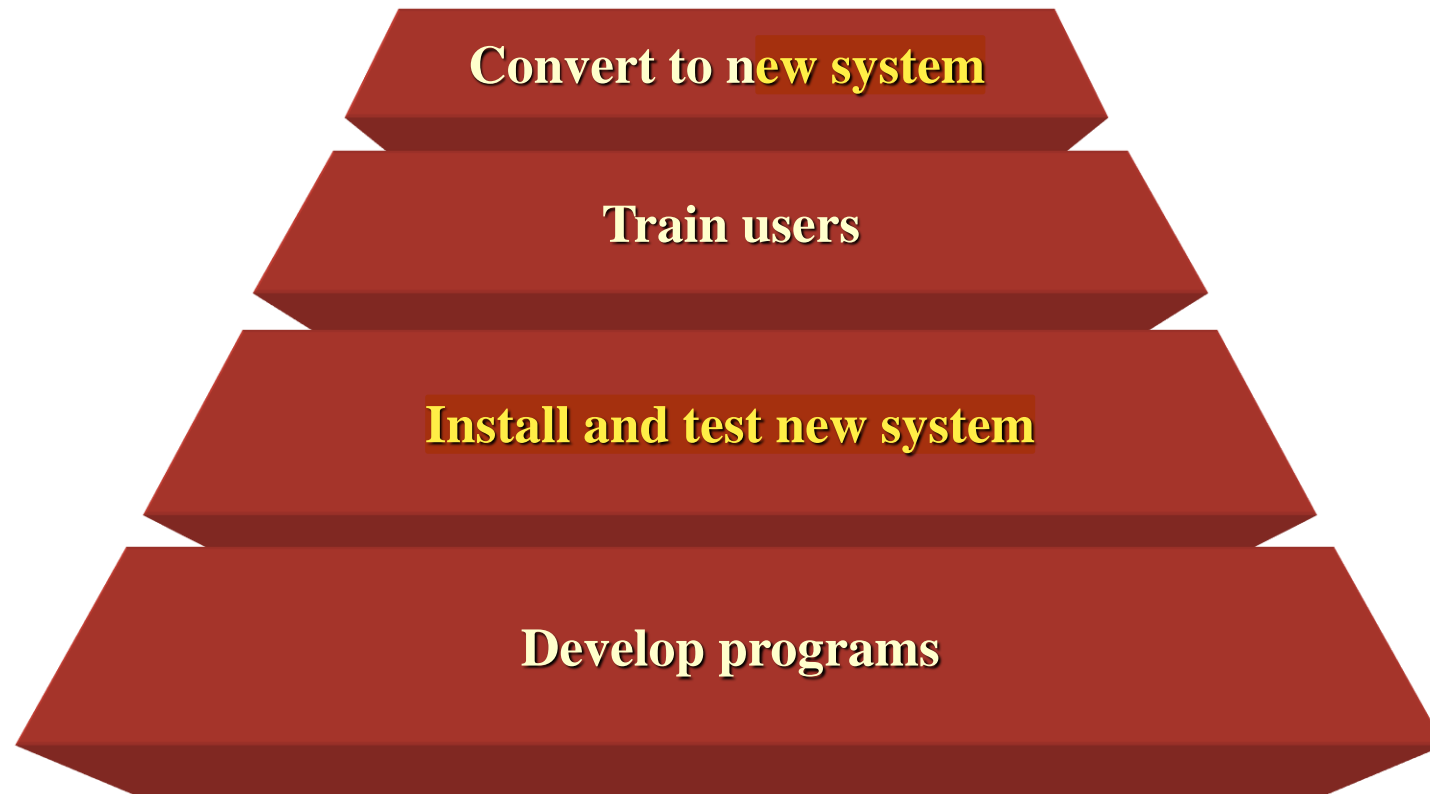
What is computer-aided software engineering (CASE)?

- **Software tools designed to support activities of system development cycle**

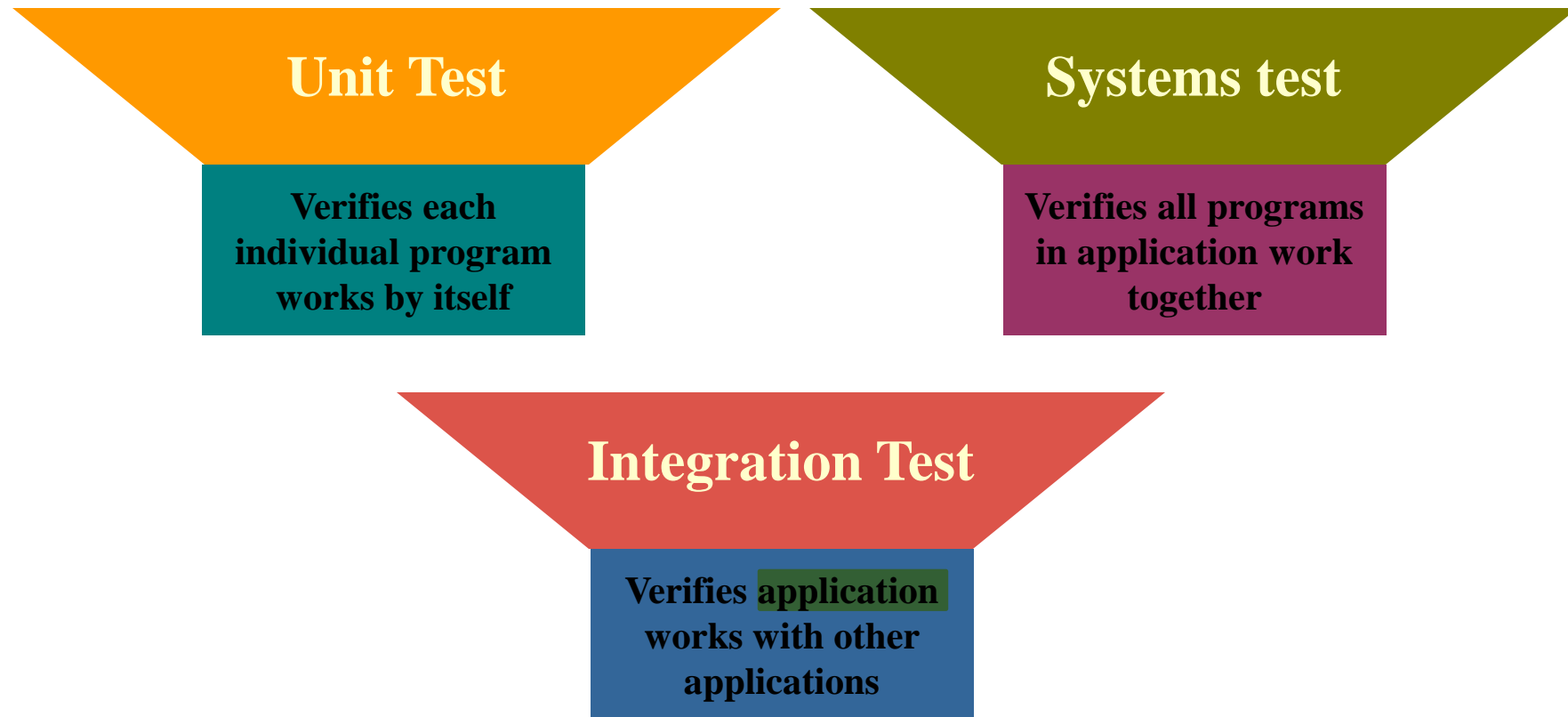


What is the implementation phase?

- **Purpose is to construct, or build, new or modified system and then deliver it to users**



What are the three types of tests performed by system developers?



What is training?

- **Showing users exactly how they will use new hardware and software in system**



What is the support phase?

➤ Provides ongoing assistance after system is implemented

Conduct post-implementation system review—meeting to find out if information system is performing according to expectations

do meeting

Identify errors

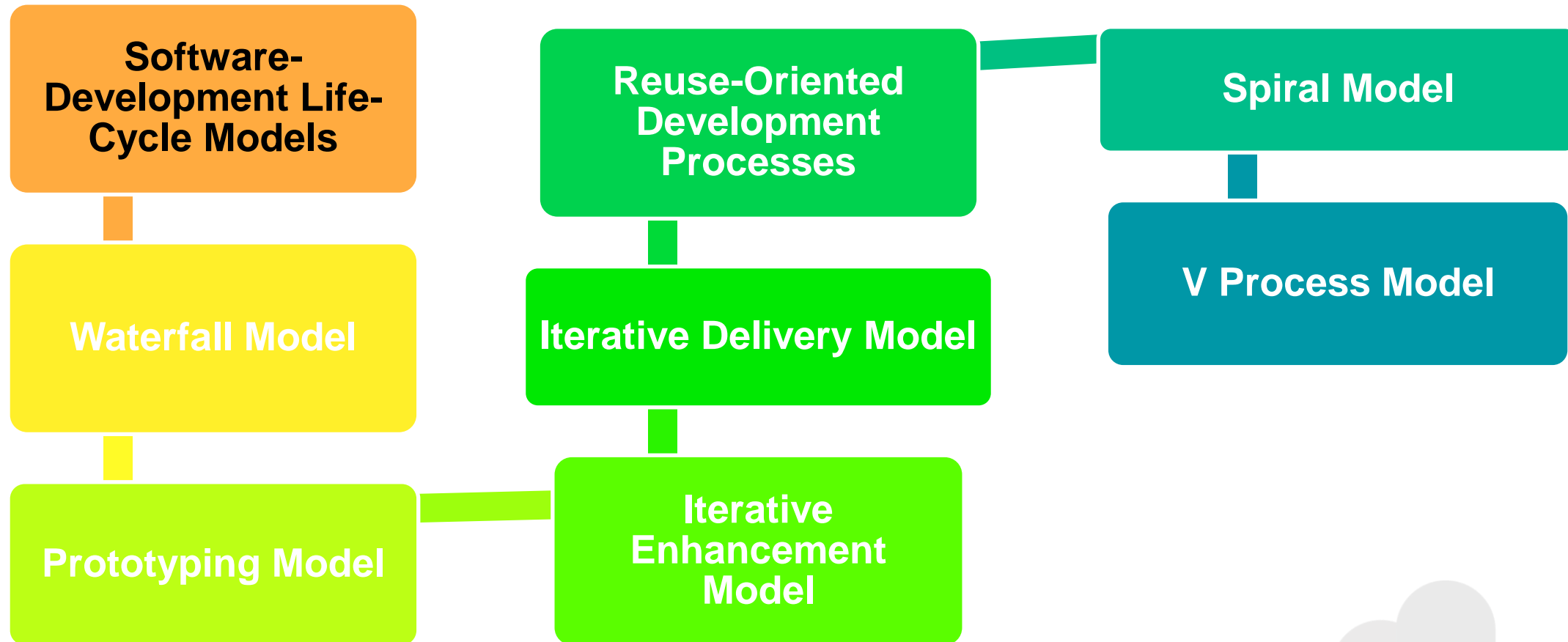
Identify enhancements

Monitor system performance

watch



The Software Development Life Cycle



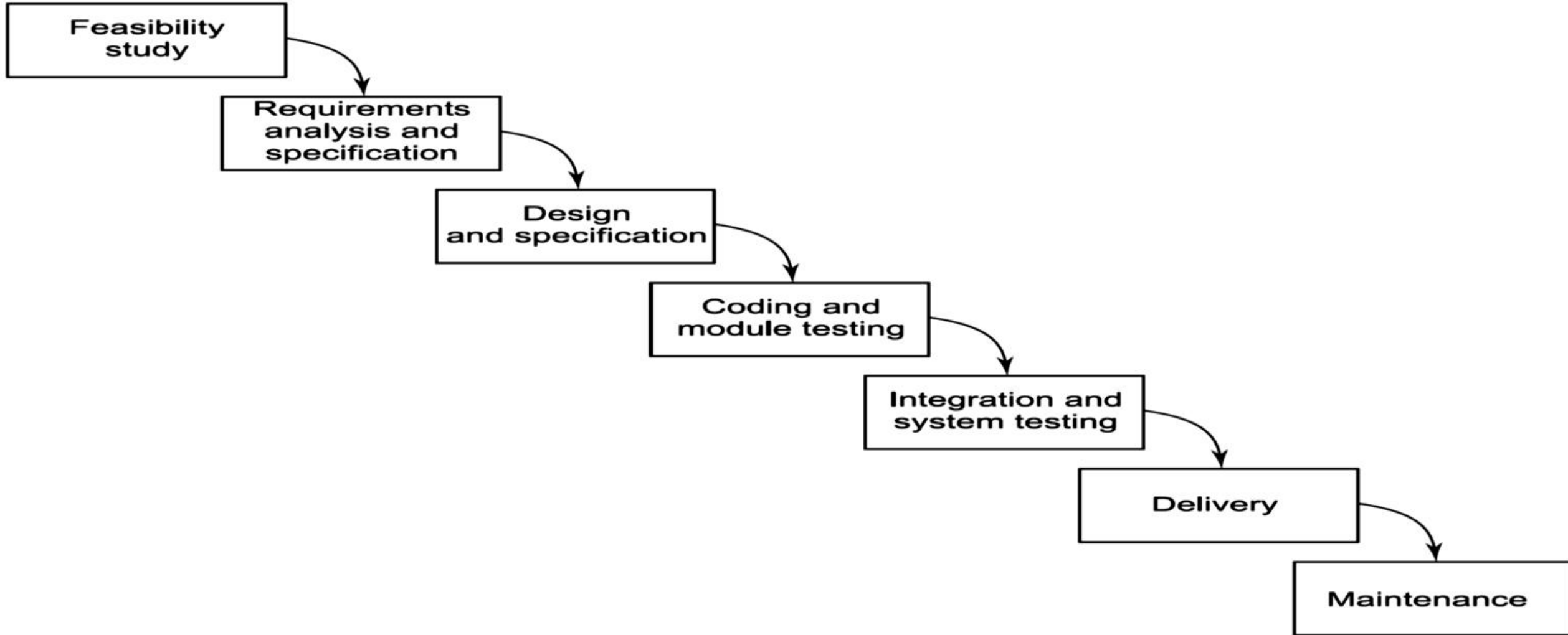
1- Waterfall Process Model

- The waterfall model is a very common software development process model.
- Because of the cascade from one phase to another, this model is known as the waterfall model.
- Each phase, in turn, is structured as a set of activities that might be executed by different people **concurrently**.
- The waterfall model is illustrated in the next Figure

شلال



Waterfall Process Model



Waterfall Process Model

1. Feasibility Study:

- The purpose of this phase is to produce a feasibility study document that evaluates the costs and benefits of the proposed application.
- The feasibility study is usually done within **limited time bounds and under pressure.**
- Often, **its result is an offer to the potential customer.** هي العرض للعميل
- **At the end** of this phase, **a report called a feasibility study** is prepared by a group of software engineers.
- This report determines whether the **project is feasible or not.**

Waterfall Process Model

2. Requirement Analysis and Specification:

- This phase **exactly tells the requirements and needs** of the project.
- The **purpose of a requirements** analysis is to identify the qualities required of the application, in terms of functionality, performance, ease of use, portability, and so on.
- The requirements describe the “**what**” of a system, not the “**how.**”
- The resultant document is known as the **software requirement specification (SRS)** document.



Waterfall Process Model

3. Design and Specification:

- The goal is to transform the requirements specified in the SRS document into a structure that is suitable for implementation in some programming language.

4. Coding and Module Testing:

- Is the phase in which we actually write programs using a programming language.
- The output of this phase is an implemented and tested collection of modules.



Waterfall Process Model

5. Integration and System Testing:

- The modules are integrated in a **planned manner**.
- The different modules making up a **software product**.
- Integration is normally carried out incrementally over a **number of steps**.
- **During each integration step, the partially integrated system is tested and a set of previously planned modules are added to it.**
- Finally, when all the modules have been successfully integrated and tested, system testing is carried out.



Waterfall Process Model

- The objective of system testing is to determine whether the software system performs per the **requirements mentioned in the SRS document.**
- This testing is known as **system testing.**
- A fully developed software product is system tested. The system testing is done in three phases: **Alpha, Beta, and Acceptance Testing.**
- **Alpha Testing** is conducted by the **software-development team** at the developer's site.



Waterfall Process Model

- **Beta** Testing is performed by **a group of friendly customers** in the presence of the software-development team.
- **Acceptance** Testing is performed by the customers themselves. If the **software is successful in acceptance testing**, the product is installed at the customer's site.



Waterfall Process Model

6. Delivery and Maintenance:

- The delivery of software is often done **in two stages**.
 - In the **first stage**, the application **is distributed among a selected group of customers** prior to its official release.
 - The **purpose** of this procedure is to perform a kind of controlled experiment to determine, on the basis of **feedback from** users, whether any changes are necessary prior to the official release.
 - In the **second stage**, the product is **distributed to the customers**.



Waterfall Process Model

Advantages of Waterfall Model

- The main advantage is its **simplicity**
- It is **straightforward and divides the large** task of building a software system into a series of cleanly divided phases, each phase dealing with a separate logical concern
- Waterfall model is also **easy to administer** in a contractual setup.
- As **each phase** is completed and its work **product produced**, **some amount of money** is given by the customer to the developing organization



Waterfall Process Model

Disadvantages of Waterfall Model

- It assumes that the requirements of a system can be frozen before the design begins (having unchanging requirements is unrealistic for some projects)
- Freezing the requirements usually requires choosing the hardware (due to the speed at which hardware using old hardware technology is changing, it is likely that the final software will use a hardware technology on the verge of becoming obsolete).
- This is clearly not desirable for such expensive software systems.



Waterfall Process Model

- It follows the “**big bang**” approach. The entire software is delivered in **one shot at the end**.
- If the project runs out of money in the middle, then there will be no software. That is, it has the “all or nothing” value proposition.
- It encourages “**requirements bloating**”
- Since all requirements must be specified at the start, it encourages the users and other stakeholders to add even those features which they think might be needed.

Waterfall Process Model

- Despite these limitations, the waterfall model has been the **most widely used process model**
- If the **developing organization is quite familiar with the problem domain and the requirements** for the software are quite clear, the waterfall model works well, and may be the most efficient process



When to use the Waterfall Model

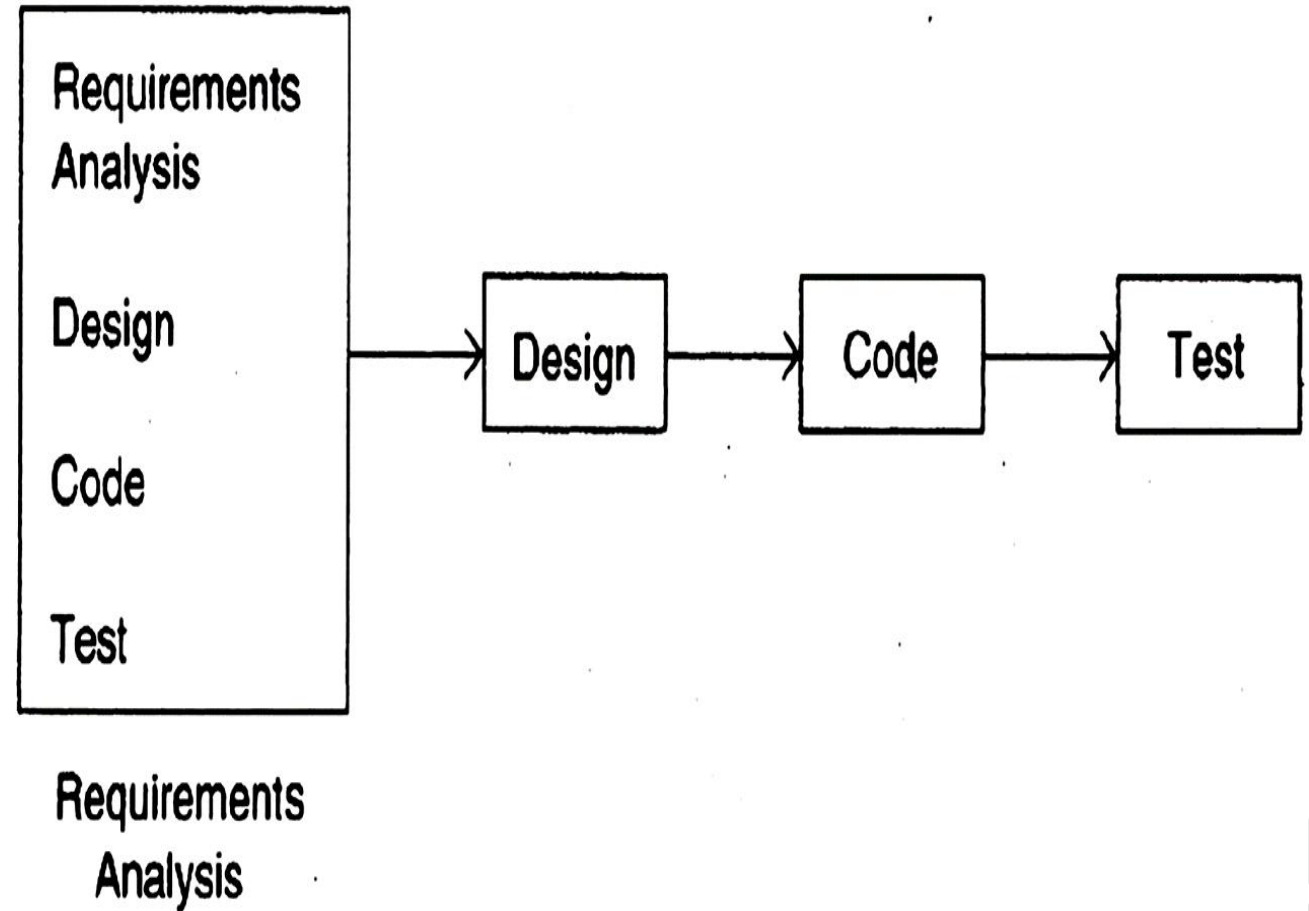
- Requirements are very **well known**
- Product definition is **stable**
- Technology is **understood**
- New **version of an existing product**
- **Porting an existing product to a new platform.**

2- Prototyping Process Model

- The goal of a prototyping-based development process is to **counter the first limitation of the waterfall model**
- The basic idea here is that instead of freezing the requirements before any design or coding can proceed, **a throwaway prototype is built to help understand the requirements**
- Prototyping is an attractive idea for complicated and **large systems.**
- This might be needed for novel systems, where it is not clear that constraints can be met or that algorithms can be developed to implement the requirements

Prototyping Process Model

- The Prototyping model is illustrated in the Figure below



Prototyping Process Model

- After the prototype has been developed, the users are given an opportunity to use and explore the prototype
- They provide feedback to the developers regarding the prototype: What is correct, what needs to be **modified**, what is missing, what is not needed, etc.
- Based on the feedback, the prototype is modified to incorporate some of the suggested changes that can be done easily, and then the users and the clients are again allowed to use the system
- **This cycle repeats until we reach the requirements needed**

Advantages of Prototyping Model

- Suitable for large systems for which there is no manual process to define the requirements
- User training to use the system
- Quality of software is good
- Requirements are not freezed

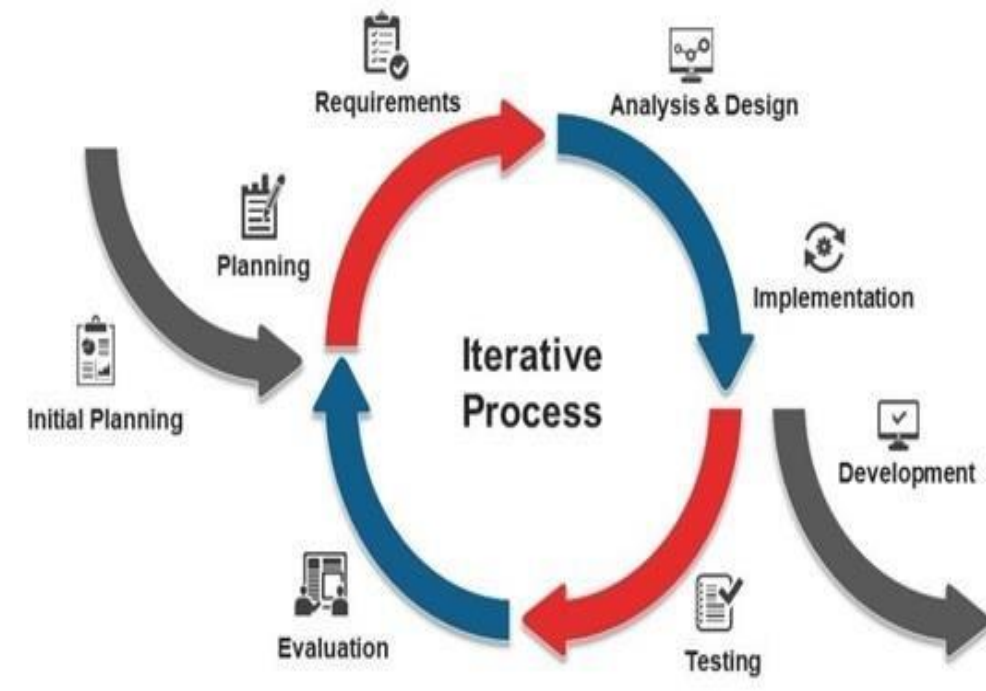


3- Iterative Model

- The iterative development process model tries to combine the benefits of both prototyping and the waterfall model.
- Two common approaches for iterative development model:

A. **Iterative-Enhancement model**

B. **Iterative Delivery model**



A. Iterative Enhancement Model

- The iterative-enhancement model combines elements of the linear sequential model (Waterfall) with the iterative philosophy of prototyping
- The software is broken down into several modules, which are incrementally developed and delivered
 1. The development team develops the core module of the system
 2. It is later refined into increasing levels of capability of adding new functionalities in successive versions

Iterative Enhancement Model

- Each linear sequence produces a **deliverable increment of the software**
- For example, word-processing software developed using the iterative paradigm might deliver basis file management, editing, and document production functions in the first increment. More sophisticated editing and document production capabilities in the second increment; and spelling and grammar checking in the third increment

Advantages of Iterative Enhancement Model

- The feedback from early increments improve the later stages
- The possibility of changes in requirements is reduced
- Users get benefits earlier than with a conventional approach
- Early delivery of some useful components improves cash flow
- Smaller sub-projects are easier to control and manage
- The project can be temporarily abandoned if more urgent work crops up
- Job satisfaction is increased for developers

Disadvantages of Iterative Enhancement Model

- **Later** increments may require modifications to **earlier** increments
- Programmers may be more productive working on one large system than on a series of smaller ones
- Some problems are difficult to divide into **functional units** (modules)

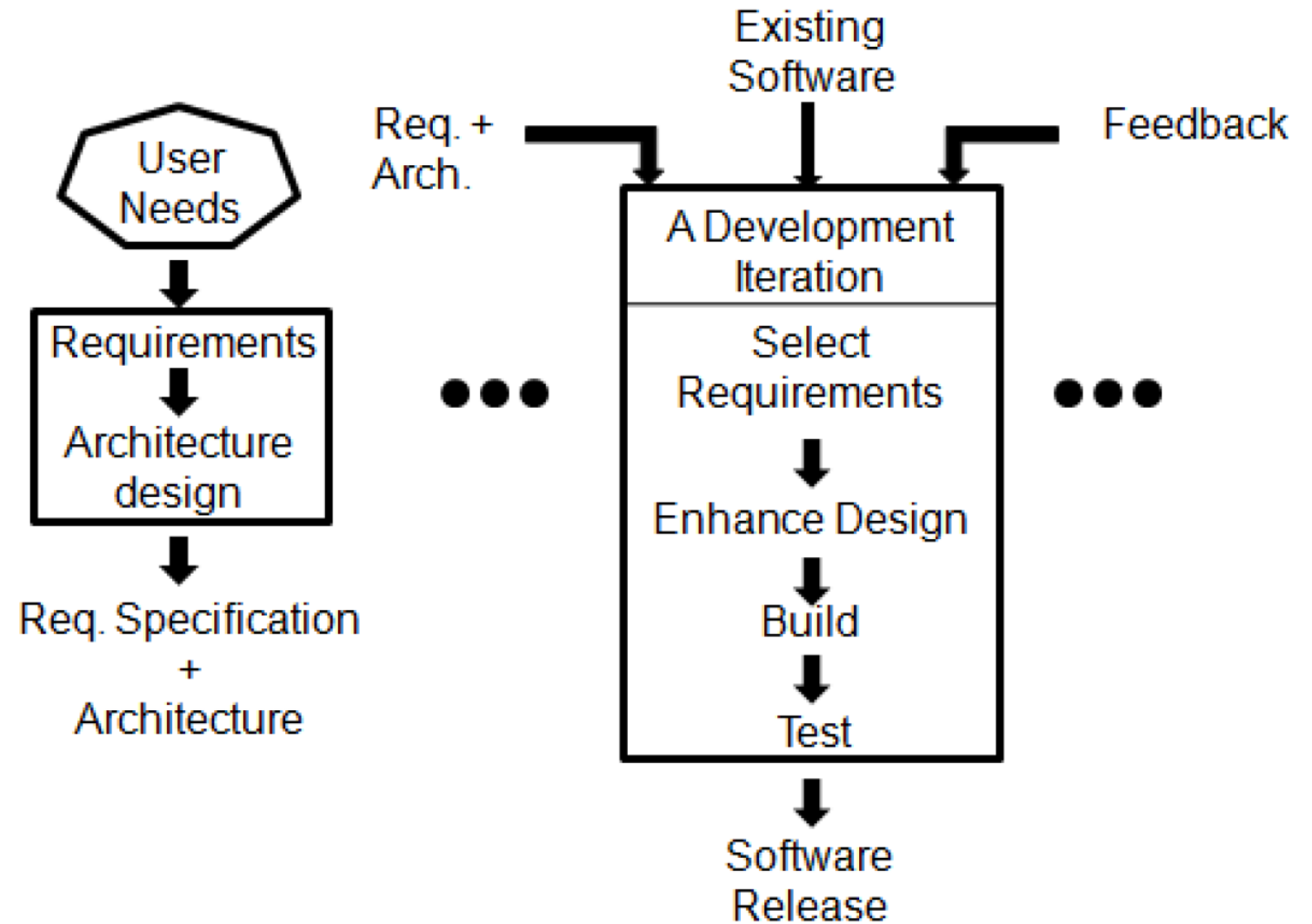
B. Iterative Delivery Model

- In this approach, the requirements and the architecture design is done in a standard waterfall or prototyping approach, but deliver the software iteratively
- That is, the building of the system, is done iteratively, though most of the requirements are specified upfront
- We can view this approach as having one iteration delivering the requirements and the architecture plan, and then further iterations delivering the software in increments.

Iterative Delivery Model

- **At the start of each delivery iteration, requirements which will be implemented in this release are decided, and then the design is enhanced and code developed to implement the requirements**
- **The iteration ends with delivery of a working software system providing some value to the end user**

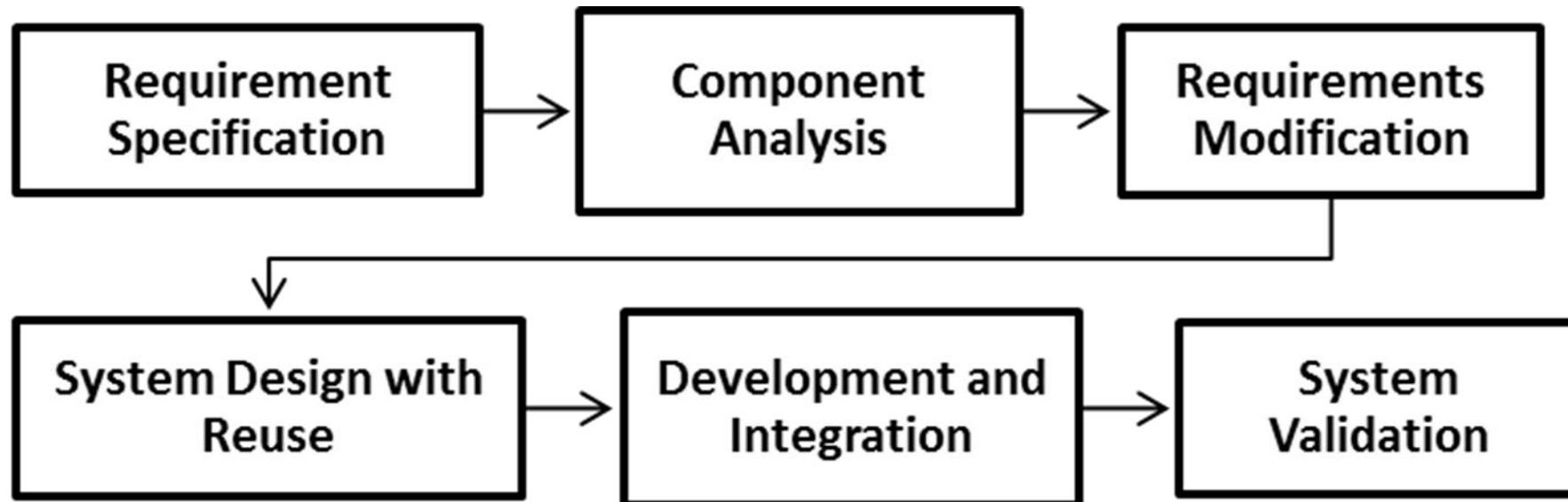
Iterative Delivery Model



4- Component-Based (Reuse-Oriented) Development

- In the majority of software projects, there is some software **reuse**
- This usually happens when people working on the project **know of designs or code which is similar to that required**
- They look for these, **modify them as required** and incorporate them into their system
- This reuse-oriented approach relies on a large base of reusable software components.

- Sometimes, these components are systems in their own right that may provide specific functionality such as text formatting or numeric calculation.
- The generic process model for component-based software engineering (CBSE) is shown in the next Figure



Reuse-Oriented Development Processes

- **While the initial requirements specification stage and the validation stage are comparable with other processes, the intermediate stages in a reuse-oriented process are different.**

those two can compare
with each other processes
but the other can't

- **These stages are as follows:**

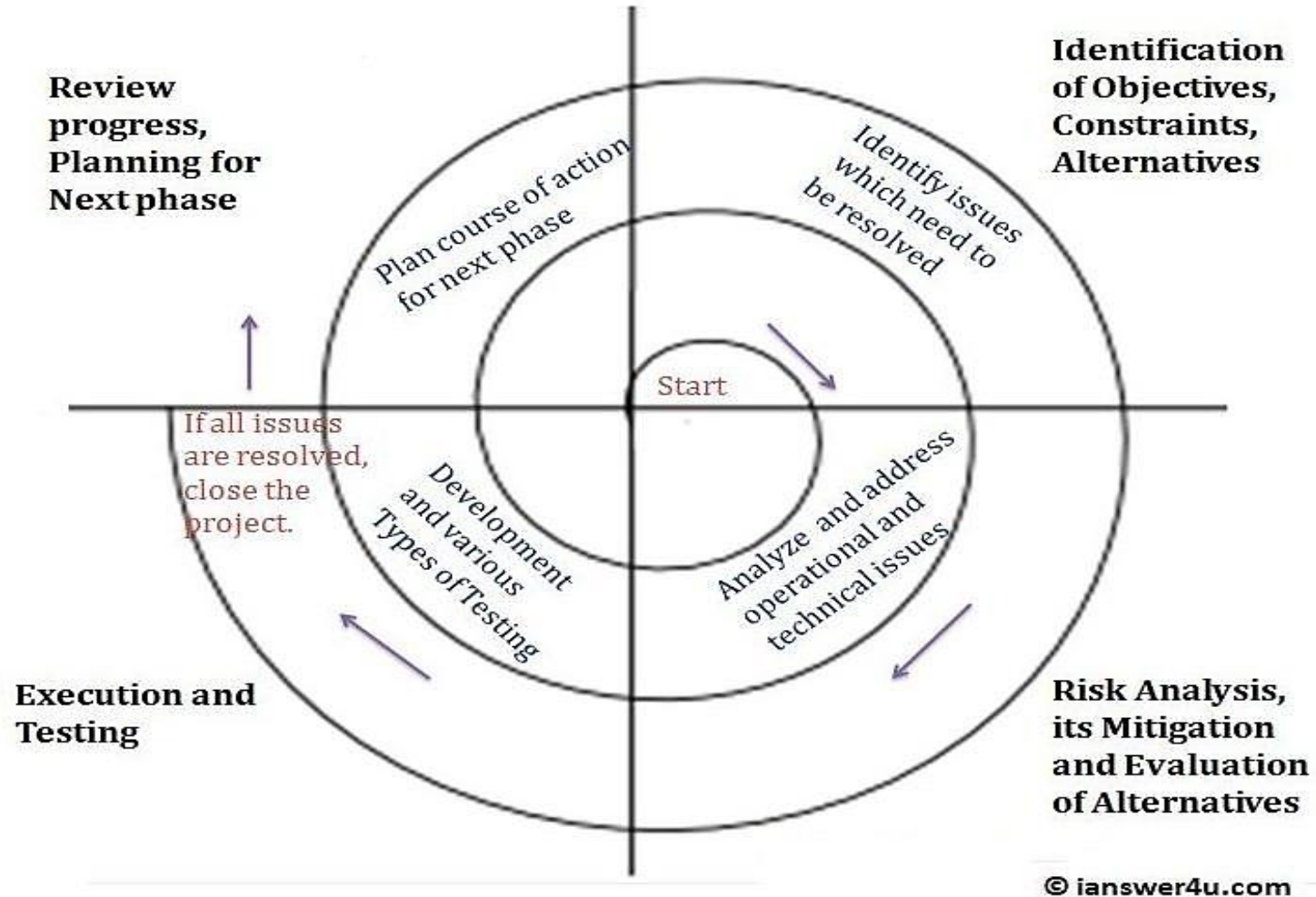
Reuse-Oriented Development Processes

- 1. Component analysis:** Given the requirements specification, a search is made for components to implement that specification.
- 2. Requirements modification:** During this stage, the requirements are analyzed using information about the components that have been discovered. They are then modified to reflect the available components.
- 3. System design with reuse:** During this phase, the framework of the system is designed or an existing framework is reused.
- 4. Development and integration:** Software that cannot be externally procured is developed, and the components and COTS systems are integrated to create the new system.

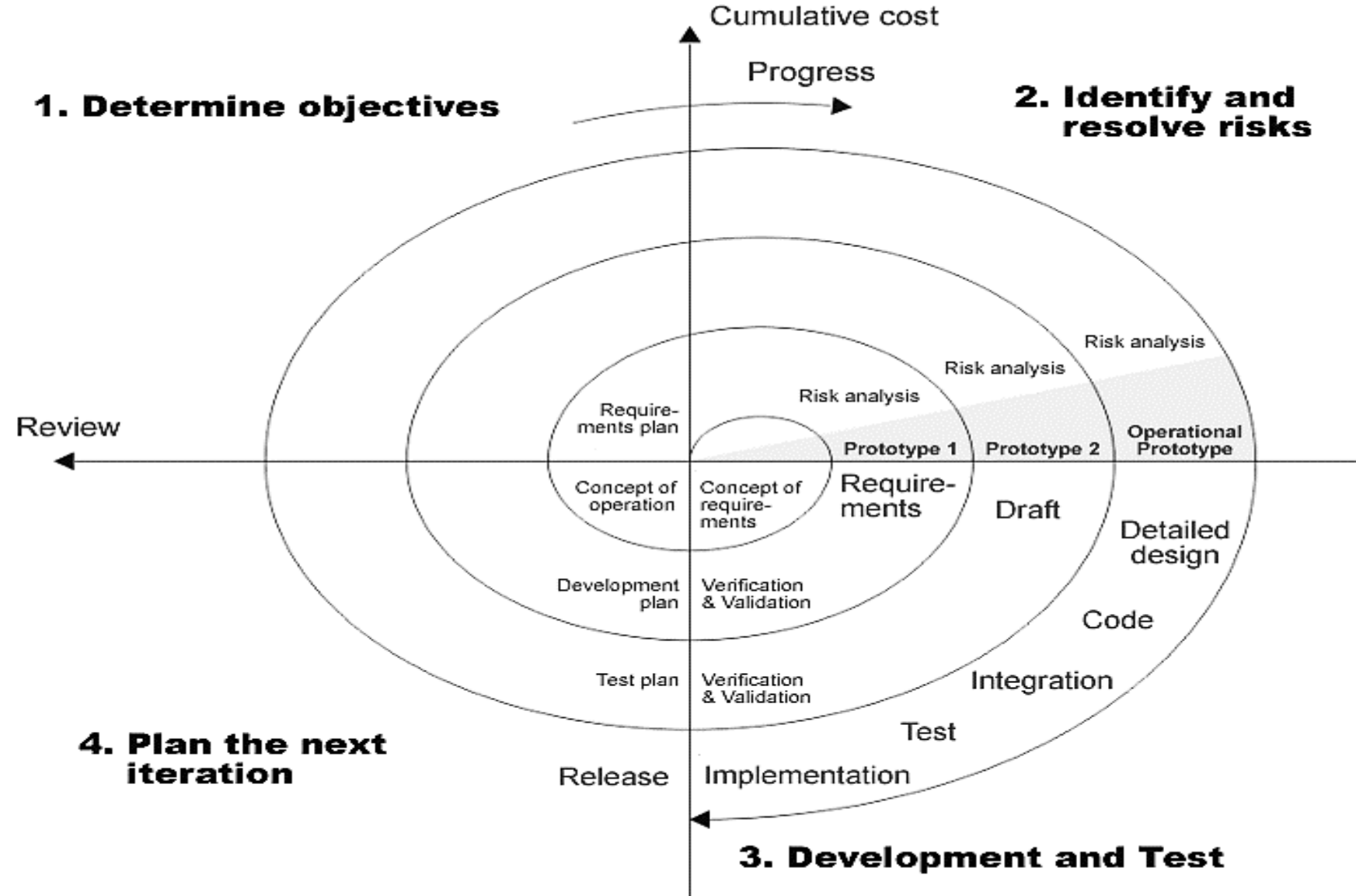
5- Spiral Model

- Rather than represent the software process as a sequence of activities with some backtracking from one activity to another, the process is represented as a spiral.
- **Each loop** in the spiral represents a phase of the software process.
- Thus, the innermost loop might be concerned with system feasibility, the next loop with requirements definition, the next loop with system design and so on.

Spiral Model



Spiral Model



Spiral Model

➤ Each loop in the spiral is split into four sectors:

1. **Objective setting:** Specific objectives for that phase of the project are defined. Constraints on the process and the product are identified and a detailed management plan is drawn up. Project risks are identified. Alternative strategies, depending on these risks, may be planned.
2. **Risk assessment and reduction:** For each of the identified project risks, a detailed analysis is carried out. Steps are taken to reduce the risk. For example, if there is a risk that the requirements are inappropriate, a prototype system may be developed.

Spiral Model

3. **Development and validation:** After risk evaluation, a development model for the system is chosen. For example, if user interface risks are dominant, an appropriate development model might be evolutionary **prototyping**. If safety risks are the main consideration, development based on **formal transformations** may be the most appropriate and so on.
4. **Planning:** The project is reviewed and a decision made whether to continue with a further loop of the spiral. If it is decided to continue, plans are drawn up for the next phase of the project.

يتم مراجعة المشروع واتخاذ قرار ما إذا كان يجب الاستمرار بدورة أخرى من الحلزون

إذا تقرر الاستمرار، يتم وضع خطط للمرحلة التالية من المشروع

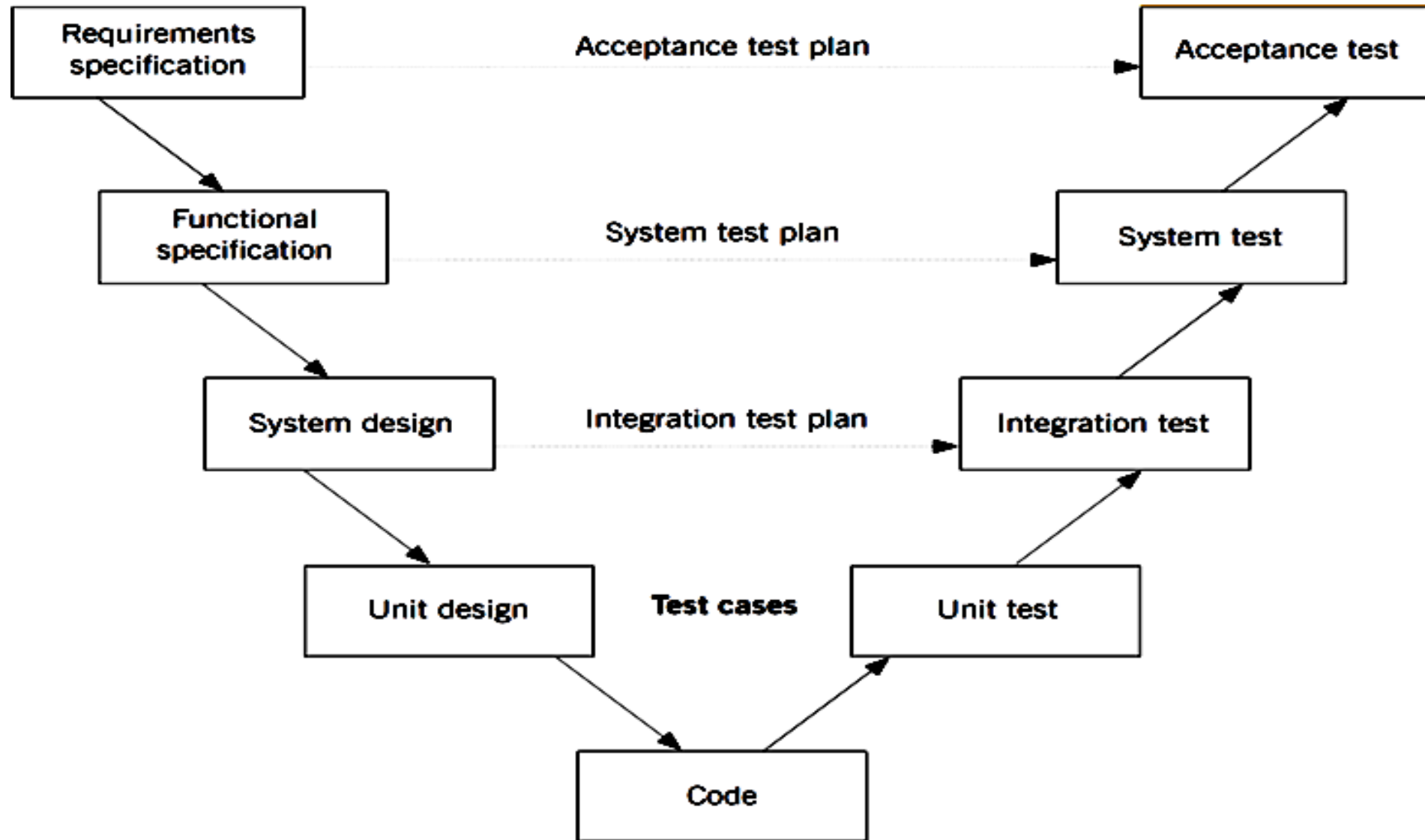
Advantages of Spiral Model

- It is risk-driven model.
- It is very flexible.
- Less documentation is needed.
- It uses prototyping.

Disadvantages of Spiral Model

- No strict standards for software development.
- No particular beginning or end of a particular phase.

6- V Process Model



V Process Model

- The V model is a **variant of the waterfall model**
- It represents a **tacit** recognition that there are testing activities occurring throughout the waterfall software life cycle model and not just during the software testing period
- For example, during requirements specification, the requirements are evaluated for testability and an **STRS** may be written
- This document would describe the strategy necessary for testing the requirements

V Process Model

- **Testing is a full life-cycle activity and that it is important to constantly consider the testability of any software requirement and to design to allow for such testability**

Quality – the degree to which the software satisfies stated and implied requirements

- **Absence of system crashes**
- **Correspondence between the software and the users' expectations**
- **Performance to specified requirements**

توافق

Quality must be controlled because it lowers production speed, increases maintenance costs and can adversely affect business

Quality Assurance Plan

- **Defect tracing** – keeps track of each defect found, its source, when it was detected, when it was resolved, how it was resolved, etc
- **Unit testing** – each individual module is tested
- **Source code tracing** – step through source code line by line
- **Technical reviews** – completed work is reviewed by peers
- **Integration testing** -- exercise new code in combination with code that already has been integrated
- **System testing** – execution of the software for the purpose of finding defects.