Contents

What are the problems of Systems Development?

- Two categories:-
  - Those that are concerned with the management of the system
  - Those that relate to poor quality in the delivered system

System Development Problems (Cont.)

- Projects generally fail due to one of the following:-
  - Unacceptable quality, or
  - Poor productivity

Quality Problems

- Quality is defined in terms of its “fitness for purpose”
- For system quality it is necessary to know:-
  - For what purpose the system is intended
  - How to measure its fitness
- Generally projects that are started with no clear idea about exactly what are the nature and goals of the client organization will be fail

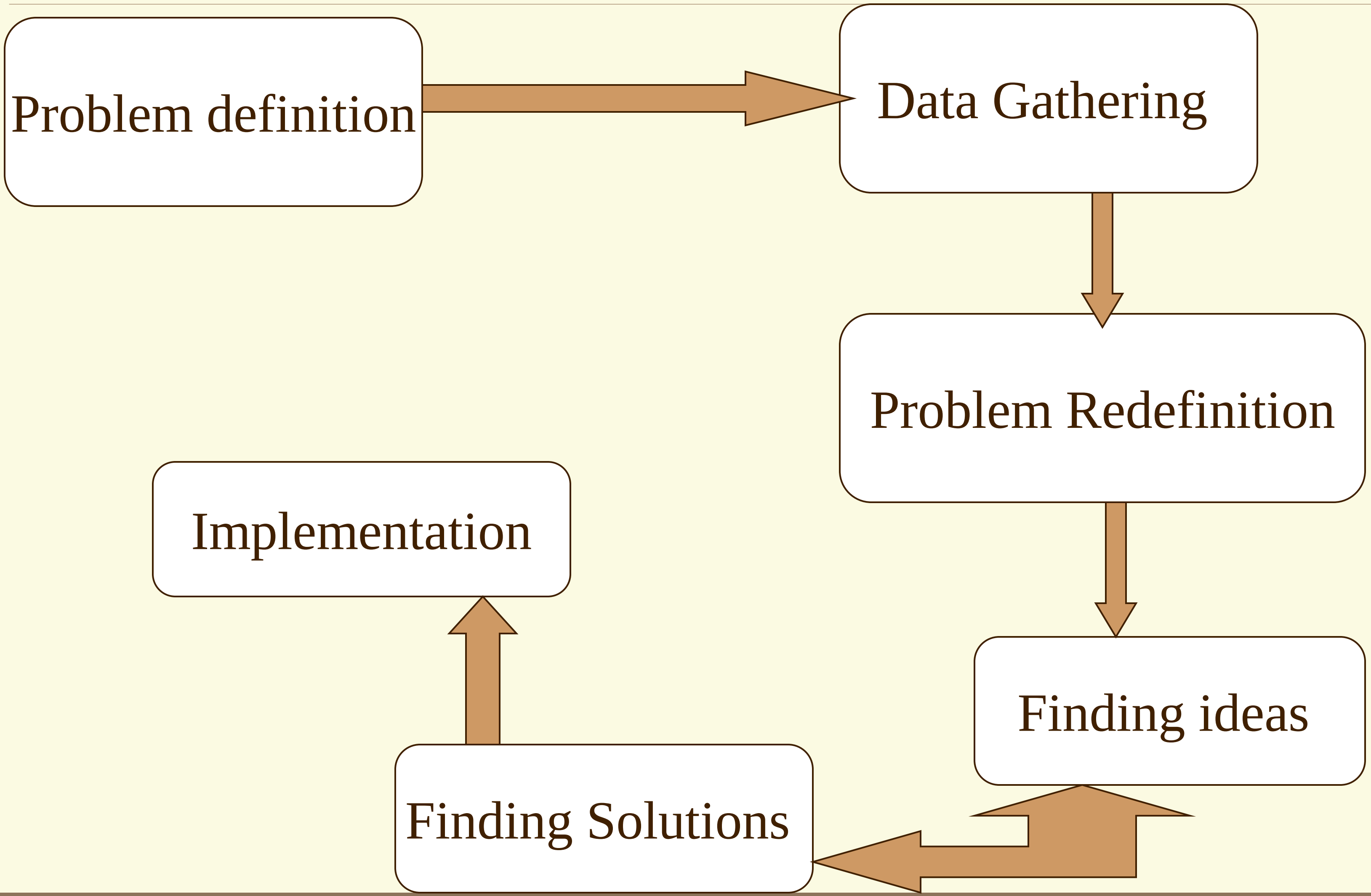
Productivity Problems

- Productivity problems relate to the rate of progress of a project and the resources (including time and money) that it consumes along the way.
- Related Questions:-
  - Will the product be delivered?
  - Will it be delivered in time to be useful?
  - Will it will be affordable (Within Budget)?

Self-test question

- From your point of view, is there is a difference between producing a function and producing a quality function?, if your answer is yes, what is the characteristics of the quality function?

General Problem Solving Model



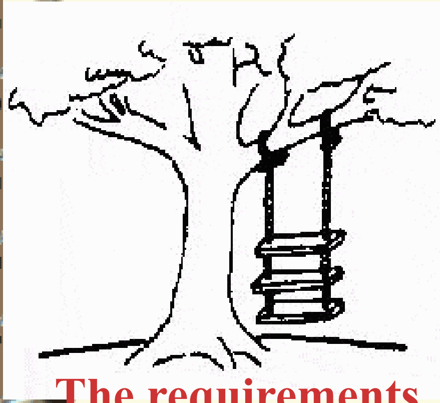
General Problem Solving Model (Cont.)

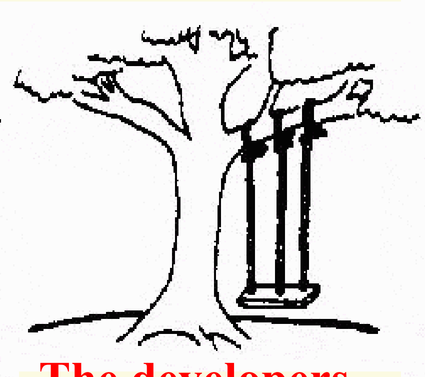
- Data Gathering and problem redefinition are concerned with understanding the nature of the problem (What is the problem is about)
- Finding ideas attempts to identify ideas that help us for understanding more about the nature of the problem and possible solutions
- Finding solutions is concerned with providing a solution to the problem
- Implementation: puts the solution into practice

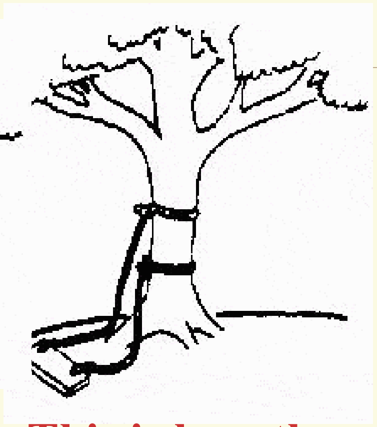
Software System Development Process

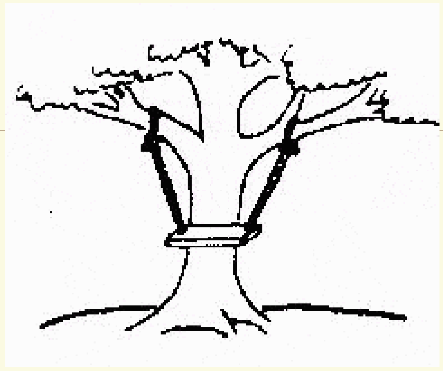
- How to deal with the problem?
- Three main tasks
  - Identifying what is required
  - Planning how to deliver what is required
  - Delivering what is required

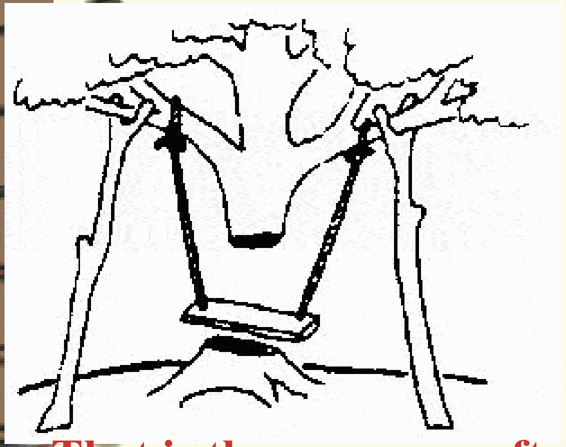
How Programs Are Usually Written …

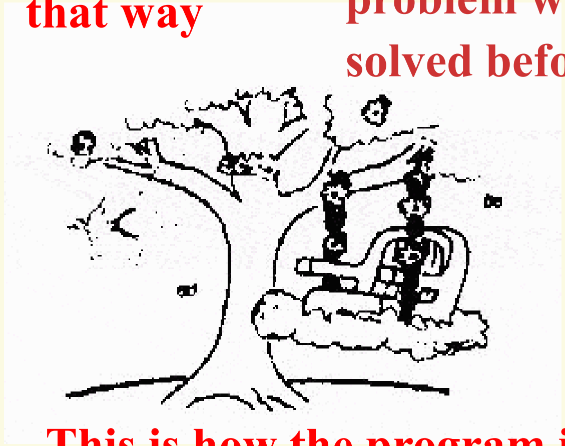

The requirements specification was defined like this

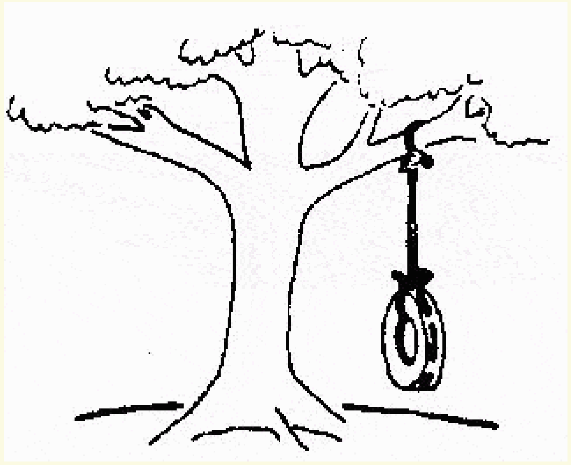

The developers understood it in that way


This is how the problem was solved before.


This is how the problem is solved now


That is the program after debugging


This is how the program is described by marketing department


This, in fact, is what the customer wanted … ;-)

Question ?

- What you need when you set about a software project
  - A clear statement of what is required
  - A package of methods and tools
  - A plan
- The plan of action is known as a process model (lifecycle model or software process)

Lifecycle Model (software process Model)

- Subdividing the process of software development produces what is known as a lifecycle model (software process model)

What is a software process model?

- **A software process model is an abstract representation of a process**
  - It presents a description of a process from some particular perspective

  - **Examples of process perspectives:**

Workflow perspective: represents inputs, outputs and dependencies

Data-flow perspective : represents data transformation activities

Role/action perspective: represents the roles/activities of the people involved in software process

- A **set of activities** whose goal is the development or evolution of software
  -

Generic Software Process Models

- **The waterfall model**
  - Separate and distinct phases of specification and development. Each phase produces a product which is the input into the next one. Widely used in **military** and **aerospace** industries
- **Evolutionary development**
  - Specification and development are interleaved
- **Formal systems development (example - ASML)**
  - A mathematical system model is formally transformed to an implementation
- **Reuse-based development**
  - The system is assembled from existing components

Common Generic Activities of Software Process Model

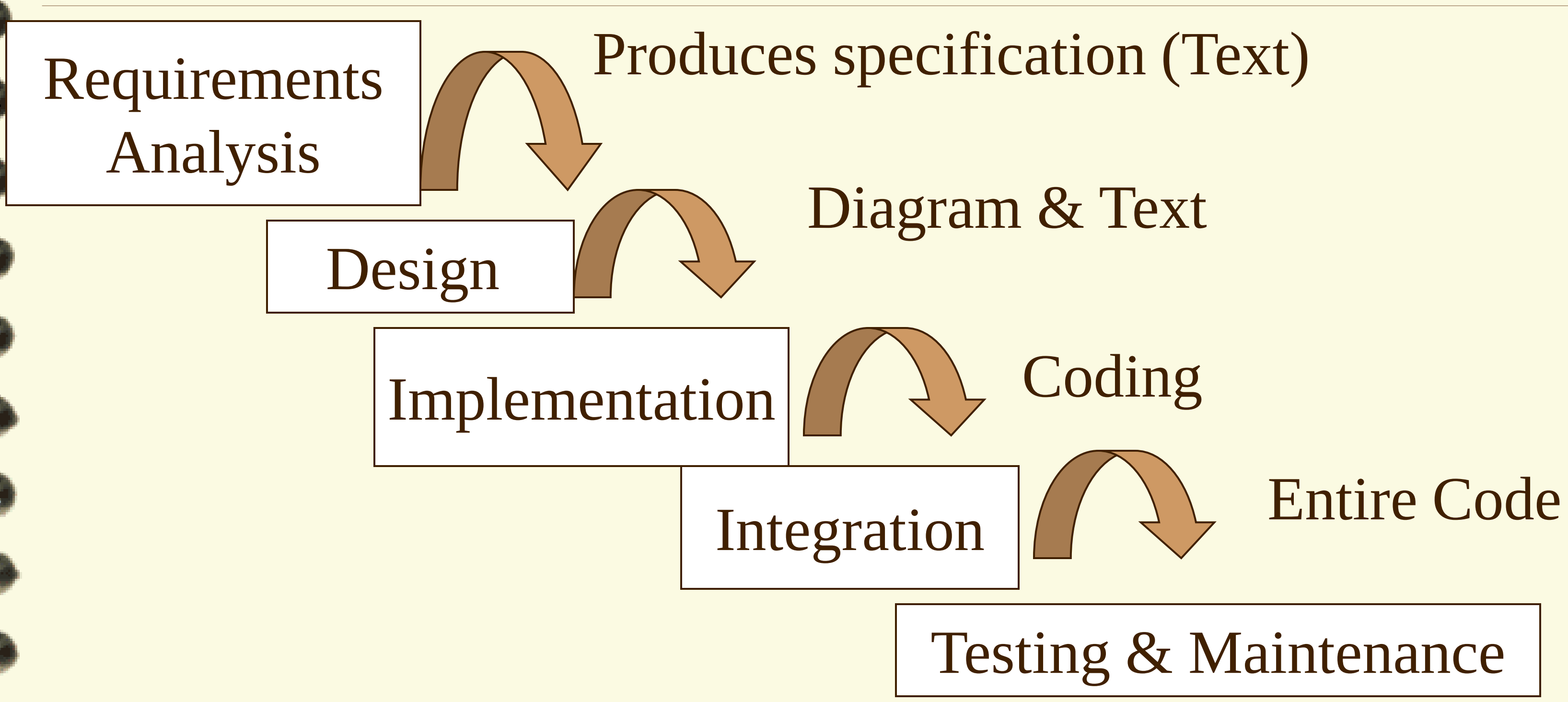
- Common generic activities to all software processes are:
  - Specification - what the system should do and its development constraints
  - Development (Design) - production of the software system
  - Validation - checking that the software is what the customer wants
  - Evolution - changing the software in response to changing demands

Classical Waterfall Model

- Software development is divided into a number of independent steps that are carried out in sequence one after the other. Each stage produces a product which is the input into the next stage

- Waterfall model is a sequence of activities (or phases) consisting of
  - Requirements analysis and definition (consists of gathering the requirements for the product. Its output is a text (specification)
  - System and software design (describes how the product will be structured internally. Output is diagrams and text)
  - Implementation (Programming, output is code) and unit testing
  - Integration and system testing (Integration is the process of assembling the parts to complete the product)
  - Operation and maintenance (The system is repaired and modified so that it continue to be useful)

Classical Waterfall Model



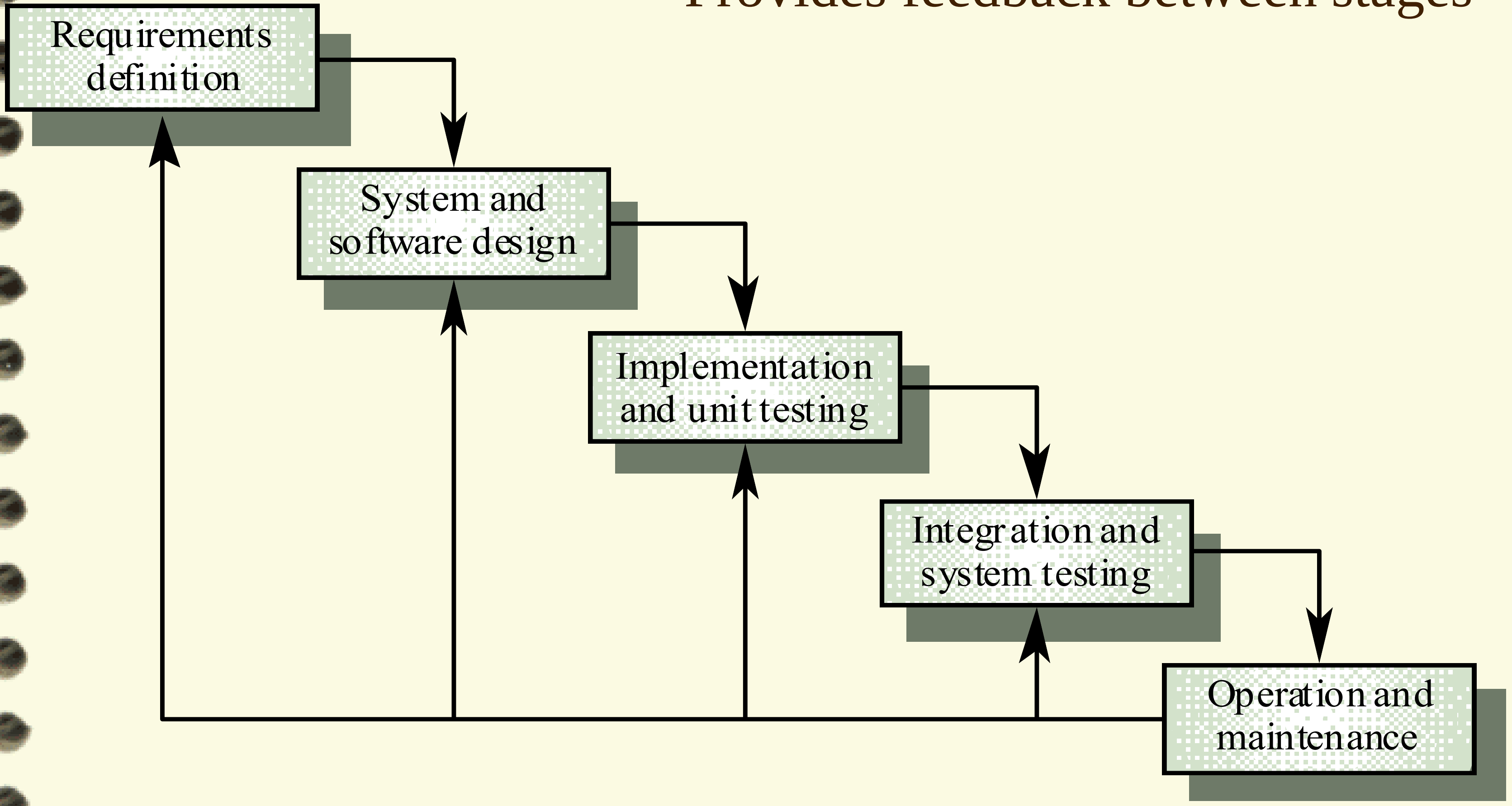
Classical Waterfall Model Drawbacks

- The drawback of the classical Waterfall model is the difficulty of accommodating change after the process is underway

  (The Waterfall model can't flow upwards)
- Examples are:
  - During coding an error in the design (the previous stage) is discovered
  - During unit testing a detailed design fault is revealed

Modified Waterfall Model

** Provides feedback between stages


Requirements definition
System and software design
Implementation and unit testing
Integration and system testing
Operation and maintenance

Modified Waterfall Model Drawbacks

- This modified Waterfall model only provides for feedback to the immediately preceding step and in reality any step may necessitate changes in any of the preceding stages
- For example
  - During user acceptance an error in the design (the previous stage) is discovered

Advantages and Disadvantages of Waterfall model

- **Advantages**
  - **It divides a complex task into smaller, more manageable tasks**
  - **Each task produces a well-defined deliverable**
- **Disadvantages**
  - **A major problem with the Waterfall model is that there is no opportunity to validate the user requirements at an early stage in development (The client only gets to see the product at the very end of the development)**
  - Waterfall model includes frequent iterations which makes it difficult to identify definite management checkpoints for planning and reporting
  - Unrealistic separation of specification from the design
  - This model makes it difficult to respond to changing customer requirements, therefore **this model is only appropriate when the requirements are well understood**

Prototyping

- Prototyping presents one solution to the problem of ensuring that the customer gets what they want. In prototyping the customer is presented at a very early stage with a working version of the system (The part of the system working well or need modification)
- **Prototyping is usually ensuring that the users needs are satisfied** (e.g. Car design)

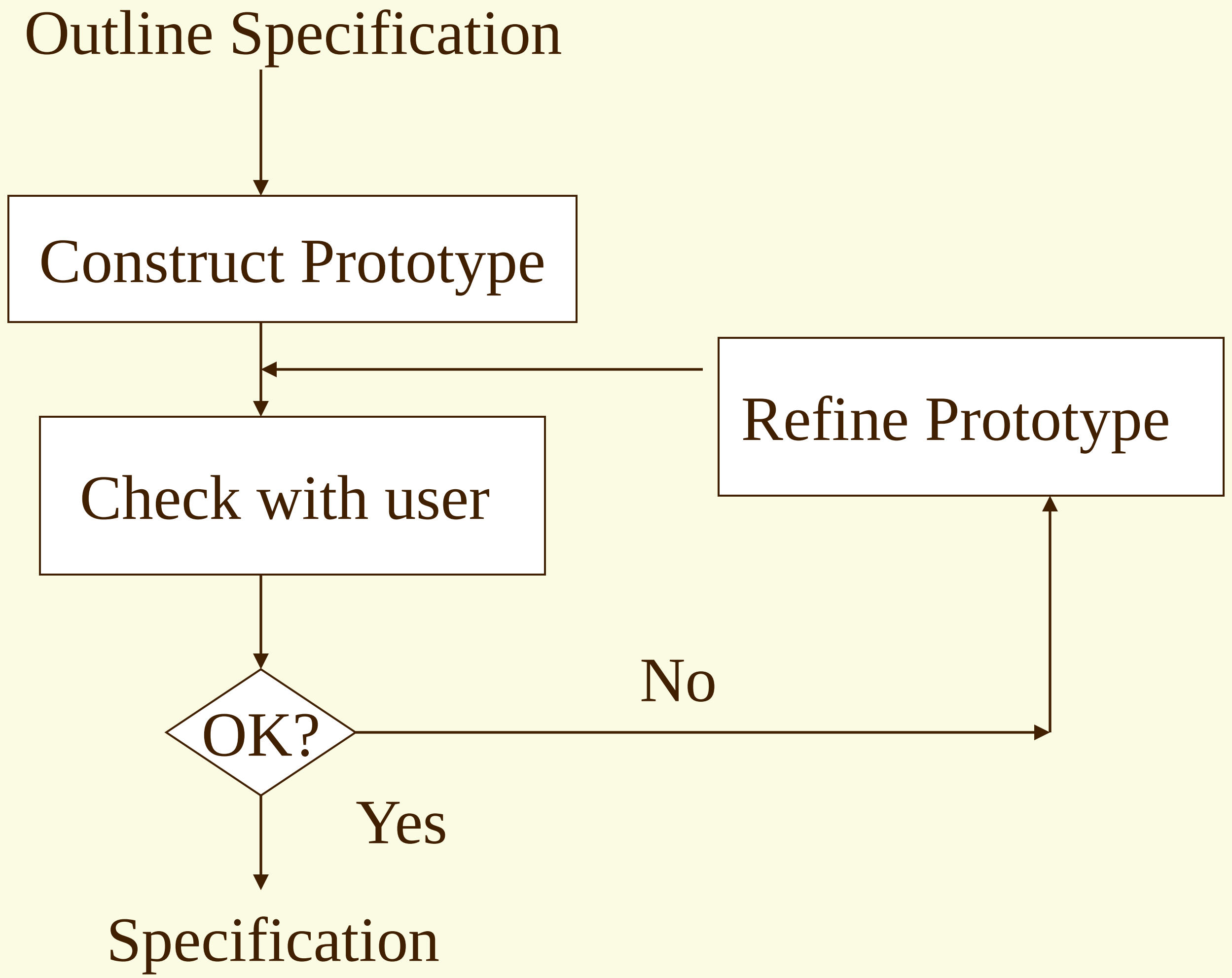
Prototyping Types

- Throwaway (the final system is implemented in some different way)
  - The product of throwaway prototype is a specification that meets the user’s requirements.
- Evolutionary (the prototype becomes the final system)
  - The product of an evolutionary prototype is a system

Throwaway Prototyping Concept and Purpose

- Throwaway Prototyping Concept
  - The starting point for throwaway prototyping is an outline specification for the software. A throwaway prototype implements only those requirements that are poorly understood. It is discarded after the desired information is learned
- Throwaway Prototyping Purpose
  - The purpose of it is the formulation of a validated specification

Throwaway Prototyping



Waterfall Model with Prototyping

- After using the prototype and when the requirements are clearly established, the prototype is thrown away and at this stage a different software process model such as the Waterfall model is undertaken

Evolutionary Prototyping

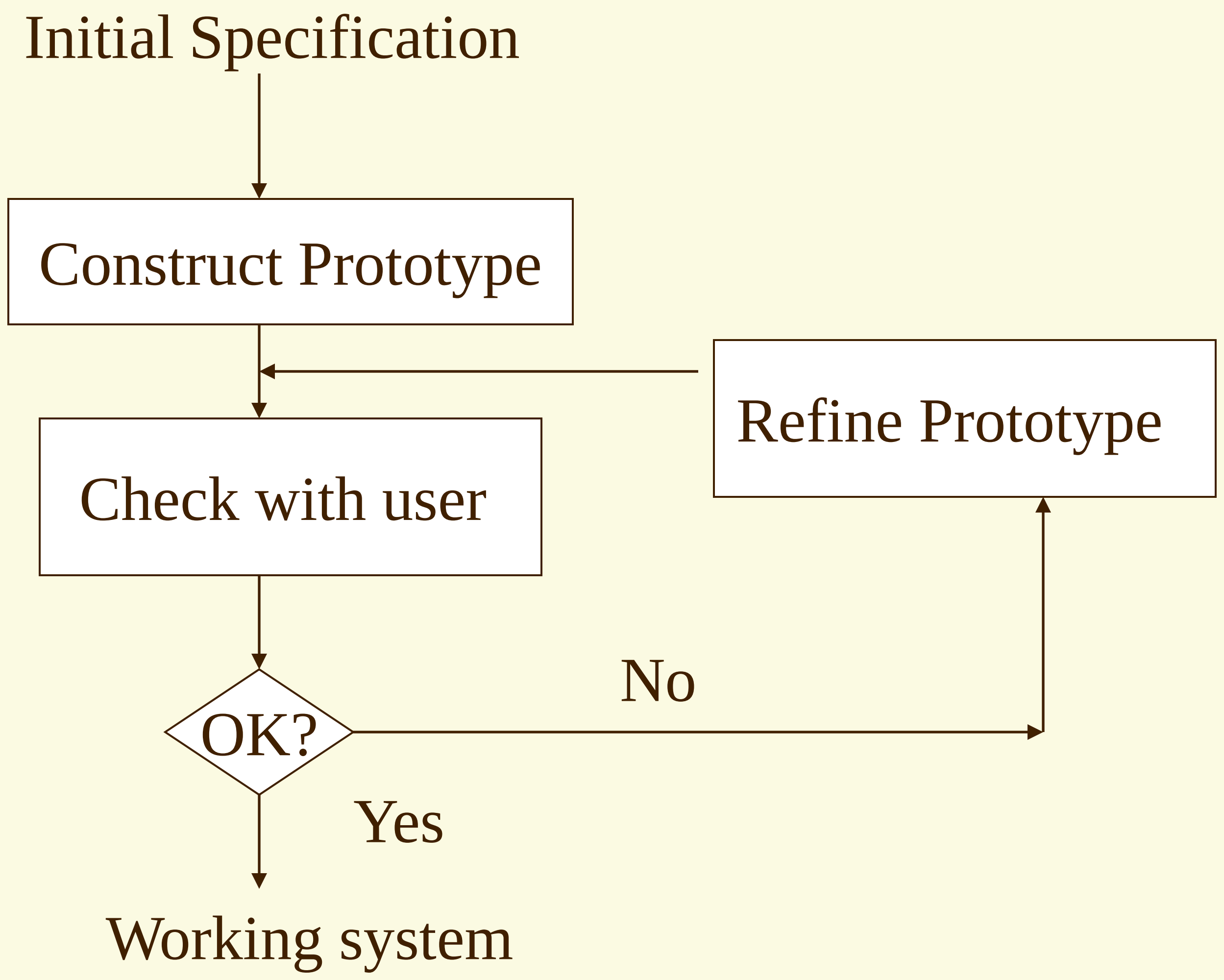
- Evolutionary prototyping is based on the idea of developing an initial exposing this to user comment and refine this through repeated stages until an adequate system has been developed

Evolutionary Prototyping Stages

1. Requirements definition (Initial specification): create an initial specification for the software
2. Prototype construction: A prototype is built in a quality manner including design, documentation and through verification
3. Evaluation (check with the user): problems in the developer's perception of the customer requirements are uncovered. The prototype are the communication medium which enables the developer and consumer to communicate with each other
4. Iteration (refine prototype): evaluation is carried out repeatedly until the prototype meets the objectives. The specification is updated with every iteration

- The product is a fully working system

Evolutionary Prototyping



Prototyping Advantages

1. A prototype gives the user a very clear picture of what the system will look like and what it will do. Thus prototyping eliminates many of the design errors in the very early stages of a project
2. It enables developers to cope with lack of clarity in requirements
3. It gives the user the opportunity to change his/her mind before commitment to the final system

Prototyping Advantages (Cont.)

- User requirements are easier to determine
- Systems are developed faster
- Maintenance effort is reduced because the system meets the user needs
- User-developer communication is enhanced
- It enables a system to be gradually introduced into an organization
- There is increased customer satisfaction with the delivered system

Prototyping Pitfalls (Disadvantages)

- Users Problems
- Software Engineers Problems
- Managerial problems

Prototyping Pitfalls For Users

- Inconsistencies between a prototype and a final system: if the prototype is through away, the end product may not be exactly like the prototype (i.e. <u>What the user sees may not be what is the user gets</u>)
- Users <u>may</u> never be satisfied because they are given too much opportunity to change the development of the system
- Undue user expectations: the ability of the developers to create a prototype so quickly may raise undue expectations that the final system will soon be complete
- Because prototyping is carried out in an artificial environment users may miss some of the shortcomings

Prototyping Pitfalls For Software Engineers

- Non-functional requirements may be omitted since a prototype focuses only on functionality
- Iteration is not easily accepted by some designers because it necessitates discarding their own work
- Incomplete analysis: because prototypes are produced quickly developers may be tempted to plunge into prototyping before sufficient requirements analysis has taken place

Managerial problems

- Estimating, planning and managing a prototyping can be difficult because it can be hard to predict how many iterations of prototyping will take place
- Procedures for change and configuration management may be unsuitable for controlling the rapid change inherent in prototyping

Some Software Systems Prototyping May not always be appropriate for them

- Embedded Software
- Real-time Software
- Scientific Numerical Software

Prototyping Disadvantages

- Users
  - Inconsistencies between a prototype and a final system:

- Software Engineers

2. Evolutionary development

Exploratory development

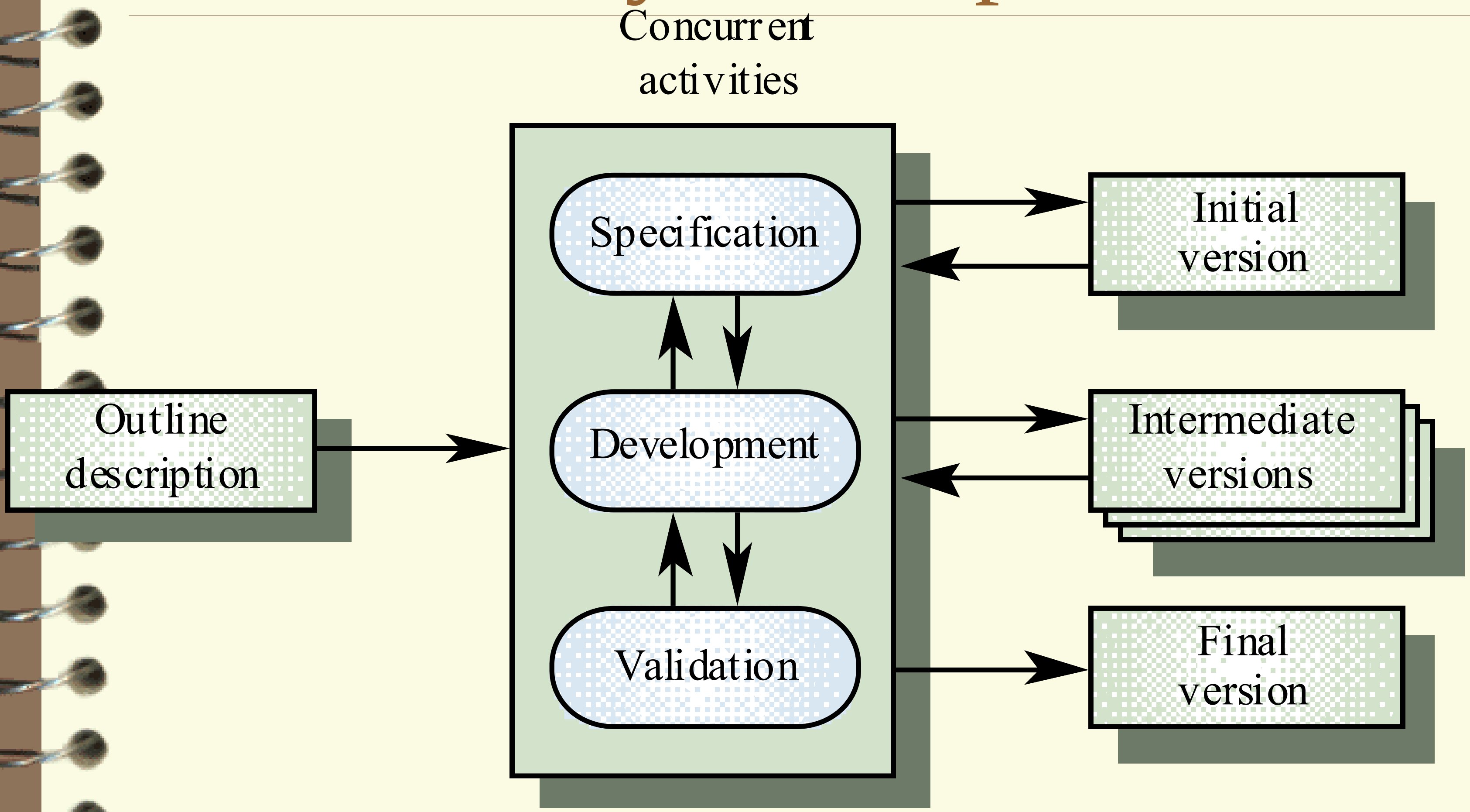
- Objective is to work with customers and to evolve a final system from an initial outline specification.
- Should start with well-understood requirements.
- The system evolves by adding new features as they are proposed by customer.

Throw-away prototyping

- Objective is to understand the system requirements. Should start with poorly understood requirements
  - **Develop "quick and dirty" system quickly;**
  - **Expose to user comment;**
  - **Refine;**

  Until adequate system developed.
- Particularly suitable where:
  - **detailed requirements not possible;**
  - **powerful development tools (e.g. GUI) available**

Evolutionary development



Evolutionary development

- **Problems**
  - Lack of process visibility
  - Systems are often poorly struc[illegible]
  - Special skills (e.g. in languag[illegible] for rapid prototyping) may be required
- **Applicability**
  - For small or medium-size interactive systems
  - For parts of large systems (e.g. the user interface)
  - For short-lifetime systems

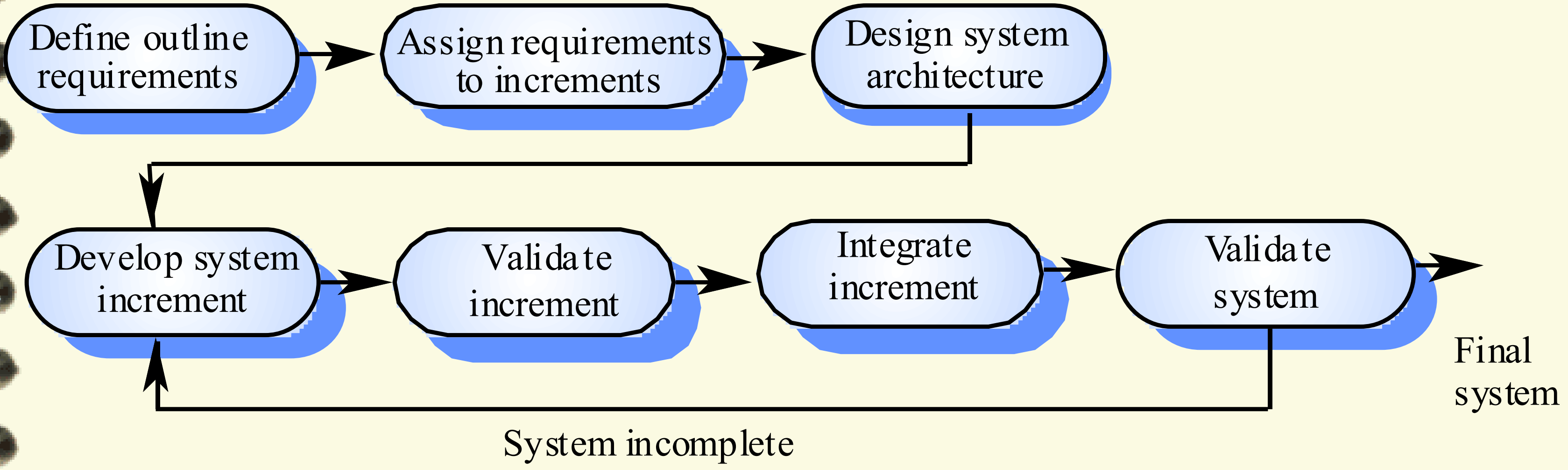
Process iteration

- **Modern development processes take iteration as fundamental,** and try to provide ways of managing, rather than ignoring, the risk. Risks are a consequence of an adequate information.
- **System requirements always evolve in the course of a project** so process iteration where earlier stages are reworked is always part of the process for large systems
- **Two (related) approaches**
  - Incremental development
  - Spiral development

Incremental development

- Rather than deliver the system as a single delivery, **the development and delivery is broken down into increments** with each increment delivering part of the required functionality
- **User requirements are prioritised** and the highest priority requirements are included in early increments
- **Once the development of an increment is started, the requirements are frozen** though requirements for later increments can continue to evolve

Incremental development



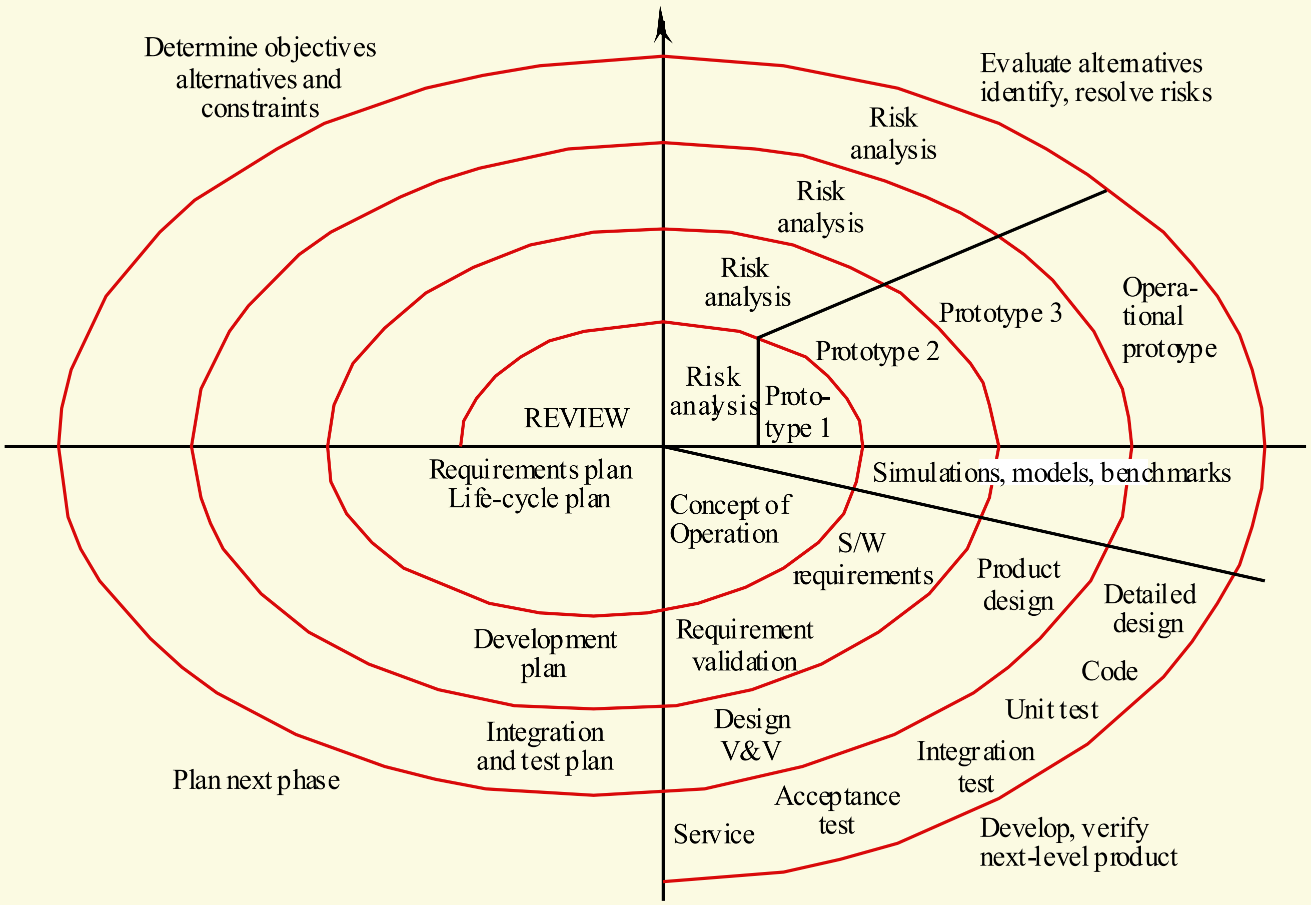
Incremental development advantages

- **Customer value** can be delivered with each increment so system functionality is available earlier
- **Early increments** act as a prototype to help elicit requirements for later increments
- Lower risk of overall project failure
- **The highest priority system services** tend to receive the most testing

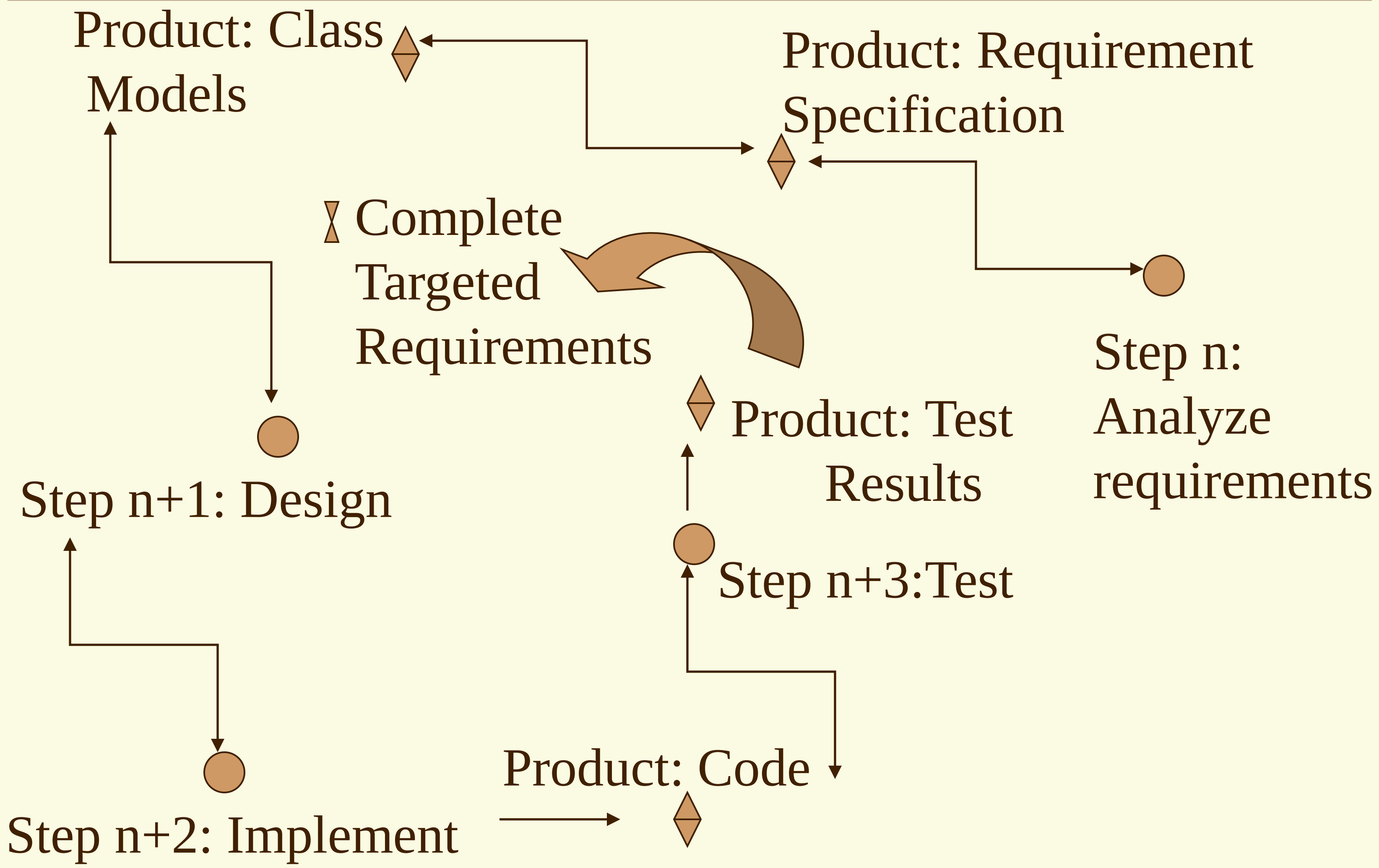
Spiral development

- **Process is represented as a spiral** rather than as a sequence of activities with backtracking
- **Each loop in the spiral represents a phase** in the process.
- **No fixed phases such as specification or design** - loops in the spiral are chosen depending on what is required
- **Risks are explicitly assessed and resolved** throughout the process

Spiral model of the software process



Spiral model of the software process



Defining “Risks”

- A risk is something which may occur in the course of a project and which under the worst outcome would affect it negatively and significantly

Risk Types

- Risks that can be avoided or worked around (retired)
  - Example: "what if the project leader in this 15-peson team leaves the company"

    (retire this by preparing a backup person)
- Risks that can't be avoided
  - Example: "2100 flight data points must be gathered from airport personnel before we can deliver the product"

Risk Management (RM)

- RM consists of the following activities:-
  - Identification (try to continually identified risks)
  - Retirement planning
  - Prioritization
  - Retirement or mitigation

Risk Identification (RI)

- RI consists of writing down all the worries and concerns of those connected with the project then continually pressing all team members to think of even more concerns.
- The following list are the most common Project Risk factors:-
  - Lack of top management commitment
  - Failure to gain user commitment
  - Misunderstanding of requirements
  - Inadequate user involvement
  - Failure to manage end-user expectations
  - Changing scope and/or objectives

Risk Retirement (RR)

- RR is the process whereby risks are reduced or even eliminated
- Two ways to retire the risk:-
  - Make change in the project requirements so that the issue causing the risk is no longer present (avoidance)
  - Develop techniques and designs that solve the problem

Spiral model sectors

- **Objective setting**
  - Specific objectives for the phase are identified
- **Risk assessment and reduction**
  - Risks are assessed and activities put in place to reduce the key risks
- **Development and validation**
  - A development model for the system is chosen which can be any of the generic models
- **Planning**
  - The project is reviewed and the next phase of the spiral is planned