APRESENTAÇÃO DA UFCD

ENGENHARIA DE SOFTWARE

PROGRAMA

- Introduction
- Software processes
- Agile software development
- Requirements engineering
- System modeling
- Architectural design
- Design and implementation
- Software testing
- Software evolution

CALENDERIZAÇÃO

- Aula 1
  - Introduction
  - Software processes
- Aula 2
  - Agile software development
  - Requirements engineering

CALENDERIZAÇÃO

- Aula 3
  - System modeling
  - Architectural design (Design Patterns)
- Aula 4
  - Design and implementation
  - Software testing
  - Software evolution

REQUISITOS PREVIOS

- UML

BIBLIOGRAFIA

- Sommerville, Ian. *Software Engineering*. 10 Harlow, England: Addison-Wesley, 2016.

AVALIAÇÃO

- Trabalho pratico
- Teste teórico (teste de escolha múltipla na ultima aula)

ENGENHARIA DE SOFTWARE

INTRODUCTION

- **Programa vs Software**
  - Programa - Aplicação
  - Software com documentação associada

O QUE LEVA UM SOFTWARE A FALHAR

- **Aumento da Complexidade so software**
- **Não usar técnicas de engenharia de software.**

DESENVOLVIMENTO PROFISSIONAL VS NÃO PORFISSIONAL

- Sofware utilizado por publico em geral
- Sofware sofre atualizações e modificações
- Técnicas de engenharia de software (analise, desenho e evolução)
- O Sofware tem documentação associada
- Sistema

DESENVOLVIMENTO PROFISSIONAL VS NÃO PORFISSIONAL

- **Um Sistema consistem em:**
  - **Vários programas distintos**
  - **Documentação**
  - **Documentação para o utilizador**
  - **Onde encontrar o sistema**

TIPOS DE SOFTWARE

- **Software genérico**
- **Software a medida**

ATRIBUTOS DE UM BOM SOFTWARE

- Deve fazer o que e suposto
- Deve ter o desempenho requerido
- Passível de manutenção
- Confiável
- Utilizável

ENGENHARIA DE SOFTWARE

"Software engineering is an engineering discipline that is concerned with all aspects of software production from the early stages of system specification through to maintaining the system after it has gone into use"

ENGENHARIA DE SOFTWARE

- **Engineering discipline**
  - Seleção e aplicação das tecnologias e metodologias apropriadas para o software que esta a ser desenvolvido
- **All aspects of software production**
  - Inclui não so a parte de tecnologia mas também tópicos como gestão de projectos de software entre outras atividades

ENGENHARIA DE SOFTWARE

- **Um Engenheiro não se limita a obter resultados, mas sim os resultados pretendidos dentro o orçamento e prazo estipulados.**
  - Selecionar os métodos e técnicas para cada cenário

ENGENHARIA DE SOFTWARE

- **Factores que levam a tornam importante existência dos engenheiros de software**
  - A nossa sociedade depende cada vezes de softwares
  - A longo prazo fica mais barato

ENGENHARIA DE SOFTWARE

- Abordagem sistemática
- Software process

ENGENHARIA DE SOFTWARE – SOFTWARE PROCESS

- Sequência de atividades que tornam possível a criação de um software
  - Especificação
  - Desenvolvimento
  - Validação
  - Evolução

ENGENHARIA DE SOFTWARE

- A engenharia de software vai beber conceitos tanto a computer science como systems engineering.

ENGENHARIA DE SOFTWARE

Não Existe uma conjunto de técnicas universais que permitem resolver todos os problemas

ENGENHARIA DE SOFTWARE

No Entanto ....

ENGENHARIA DE SOFTWARE

- **Existem quatro questões que são comuns a maioria dos sistemas a serem desenvolvidos:**
  - Heterogeneidade
  - Mudanças na sociedade
  - Questões de segurança
  - Escala

DIVERSIDADE DE SOFTWARE

A engenharia de software é uma abordagem sistemática a produção de software.

DIVERSIDADE DE SOFTWARE

- **Não Existe uma formula universal para a resolução de problemas em engenharia de software**
  - Conjunto de técnicas que tem vindo a ser desenvolvidas nos últimos 50 anos
  - SEMAT (Software Engineering Method and Theory)

DIVERSIDADE DE SOFTWARE

- **O que mais influencia nas técnicas utilizadas é o tipo de software:**
- **Alguns tipos de Software:**
  - Stand-alone applications
  - Interactive transaction-based applications
  - Embedded control systems
  - Batch processing systems
  - Entertainment systems

DIVERSIDADE DE SOFTWARE

- **Tipos de software**
  - Systems for modeling and simulation
  - Data collection and analysis systems
  - Systems of systems

DIVERSIDADE DE SOFTWARE

- **Fundamentos da engenharia de software que se aplicam a “todos” os tipos de software:**
  - O desenvolvimento deve ser planeado
  - Fiabilidade e desempenho
  - Entender e gerir os requisitos do software
  - Reutilização de código

ENGENHARIA DE SOFTWARE

SOFTWARE PROCESSES

SOFTWARE PROCESS

A software process is a set of related activities that leads to the production of a software system

- Quando se descreve uma atividade
  - Indicar os envolvidos
  - O que é produzido
  - Pré e pos condições

- Atividades existente na maioria dos software processes
  - Software specification
  - Software development
  - Software validation
  - Software evolution

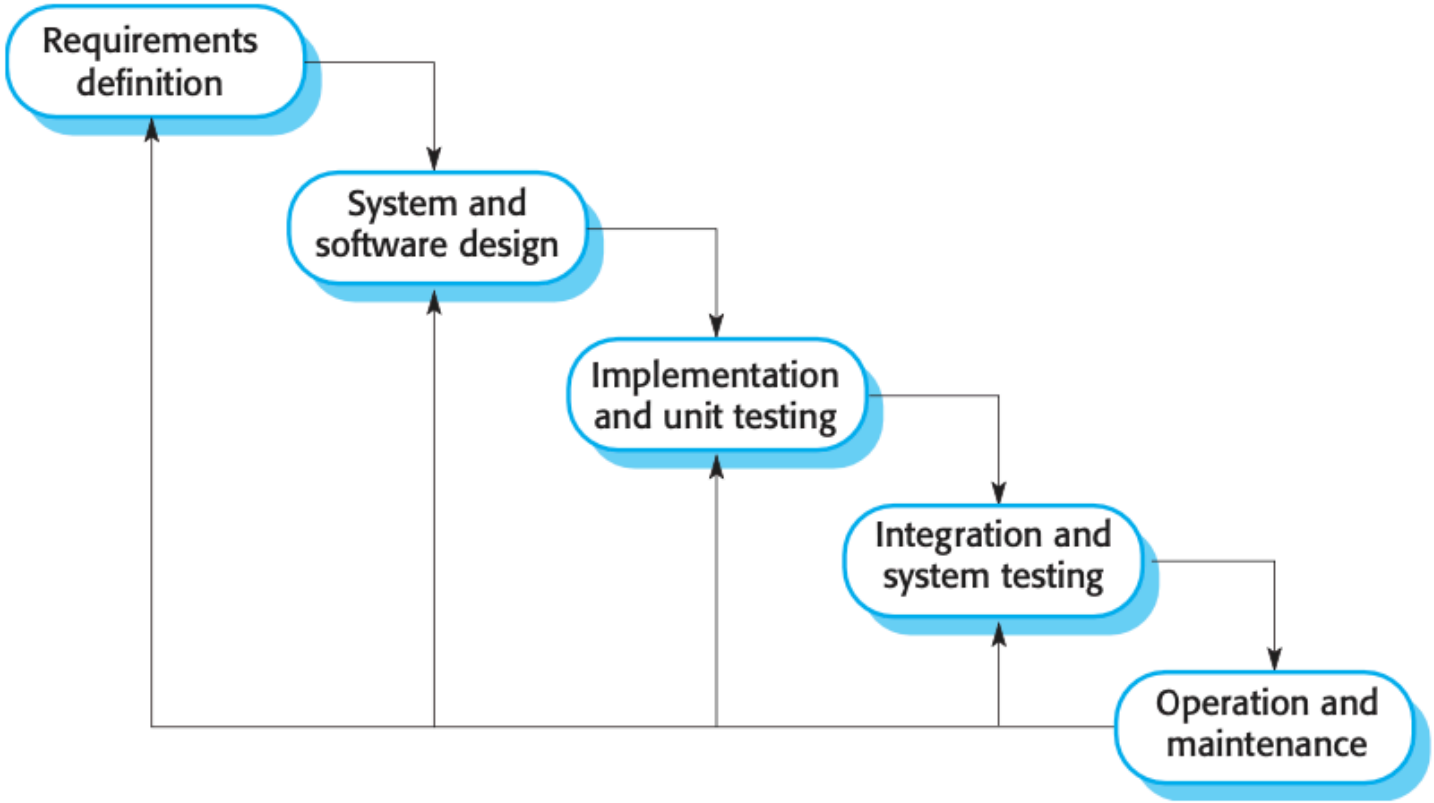
SOFTWARE PROCESS MODELS

- Versão simplificada de um Software process
- Software process models genéricos
  - Waterfall model
  - Incremental development
  - Integration and configuration

SOFTWARE PROCESS MODELS – WATERFALL MODEL

- **plan-driven process**
- **Etapas**
  - Requirements analysis and definition
  - System and software design
  - Implementation and unit testing
  - Integration and system testing
  - Operation and maintenance

SOFTWARE PROCESS MODELS – WATERFALL MODEL



SOFTWARE PROCESS MODELS – INCREMENTAL DEVELOPMENT

- plan-driven
- agile
- Mix das duas

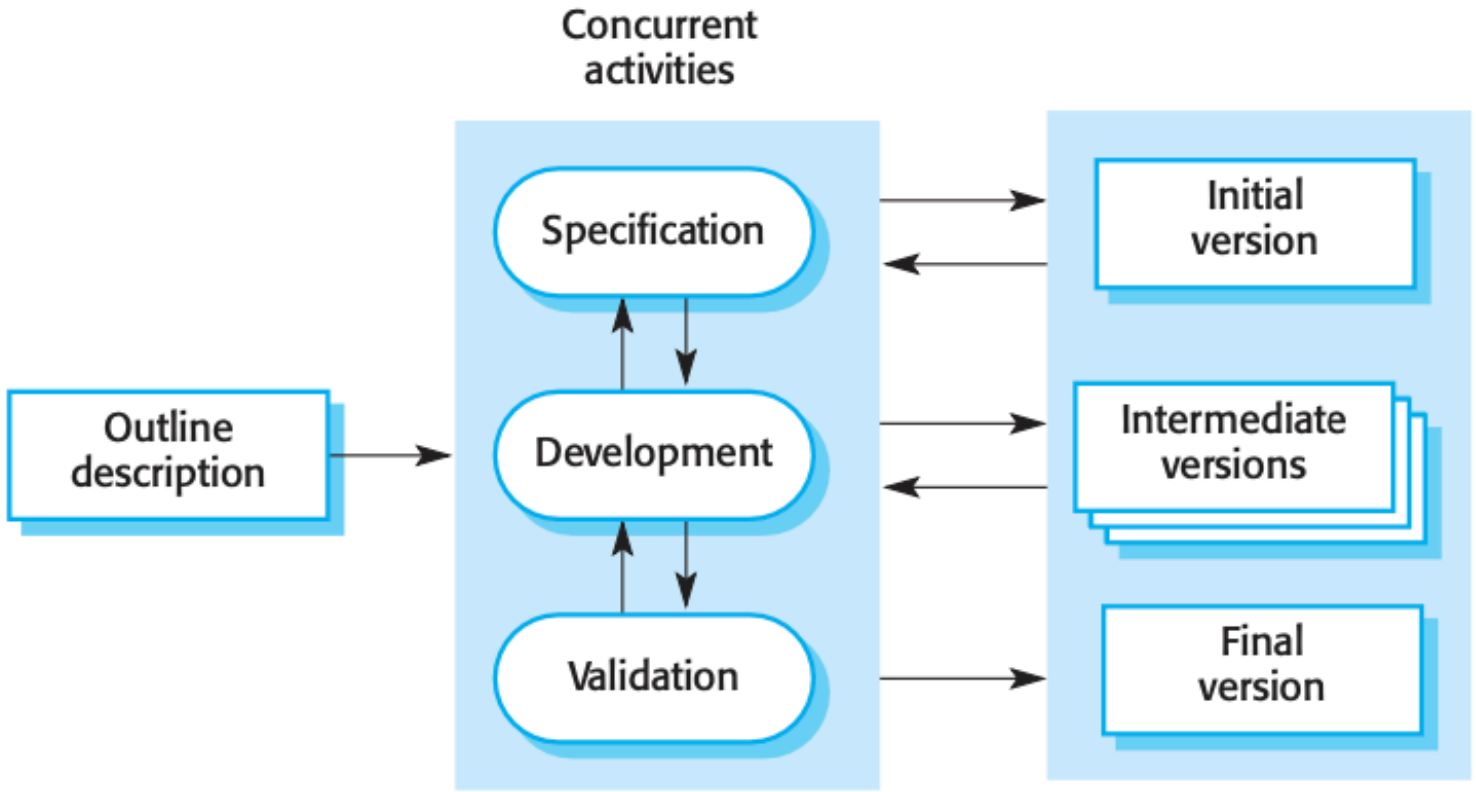
SOFTWARE PROCESS MODELS – INCREMENTAL DEVELOPMENT

Incremental development is based on the idea of developing an initial implementation, getting feedback from users and others, and evolving the software through several versions until the required system has been developed

INCREMENTAL DEVELOPMENT

- Cada versão deve
  - Ter feedback (do cliente?)
  - Implementar novas funcionalidades
    - 1° são implementadas as funcionalidades mais urgentes

SOFTWARE PROCESS MODELS – INCREMENTAL DEVELOPMENT



INCREMENTAL DEVELOPMENT VS WATERFALL MODEL

- Alteração de requisitos simplificada
- Mais feedback
- Disponibiliza versões mais rapidamente

INCREMENTAL DEVELOPMENT VS WATERFALL MODEL

Desvantagens?

INCREMENTAL DEVELOPMENT VS WATERFALL MODEL

Util para todos os projectos? Porquê?

SOFTWARE PROCESS MODELS – INTEGRATION AND CONFIGURATION

Foca-se na reutilização de software

SOFTWARE PROCESS MODELS – INTEGRATION AND CONFIGURATION

- **Componentes reutilizados com frequência:**
  - Stand-alone application systems that are configured for use in a particular environment
  - Collections of objects that are developed as a component or as a package to be integrated with a component framework
  - Web services

SOFTWARE PROCESS MODELS – INTEGRATION AND CONFIGURATION

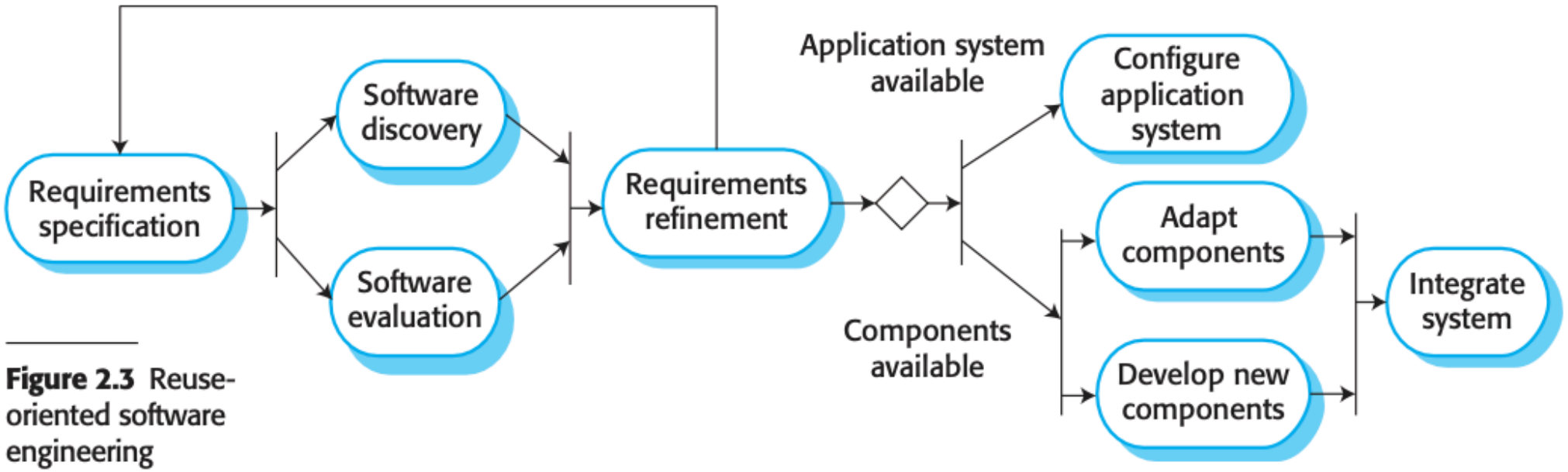


Figure 2.3 Reuse-oriented software engineering

SOFTWARE PROCESS MODELS – INTEGRATION AND CONFIGURATION

- **Etapas:**
  - Requirements specification
  - Software discovery and evaluation
  - Requirements refinement
  - Application system configuration
  - Component adaptation and integration

PROCESS ACTIVITIES

- **Existem quatro processos base:**
  - Specification
  - Development
  - Validation
  - Evolution

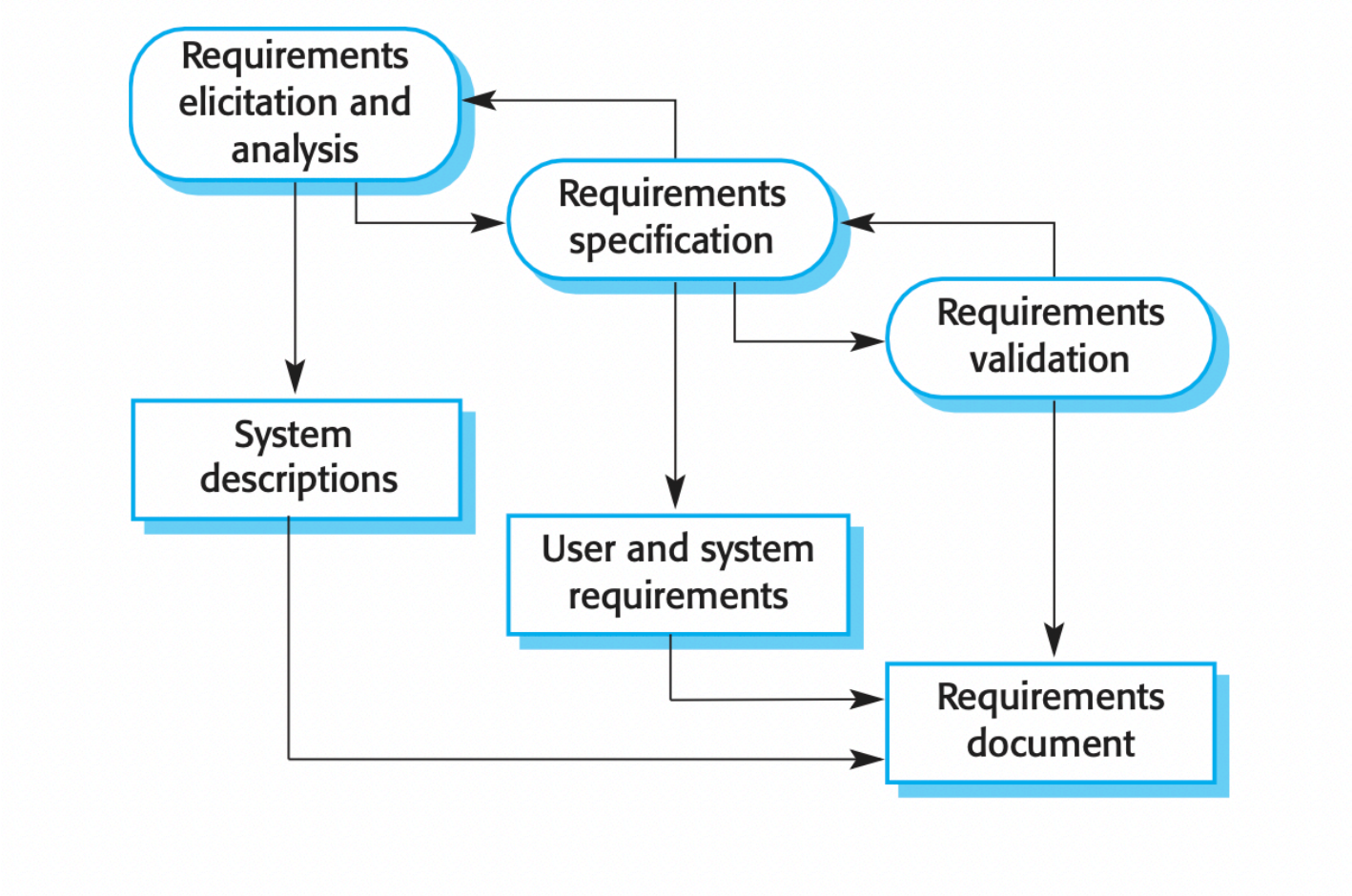
SOFTWARE SPECIFICATION

- **Entender e definir os requisitos**
- **Identificar restrições**
  - Etapa de extrema importancia
- **Produção de um documento com as especificações do software**
  - **Para o utilizador**
  - **Tecnicos**

SOFTWARE SPECIFICATION

- Composto por três etapas
  - Requirements elicitation and analysis
  - Requirements specification
  - Requirements validation

SOFTWARE SPECIFICATION



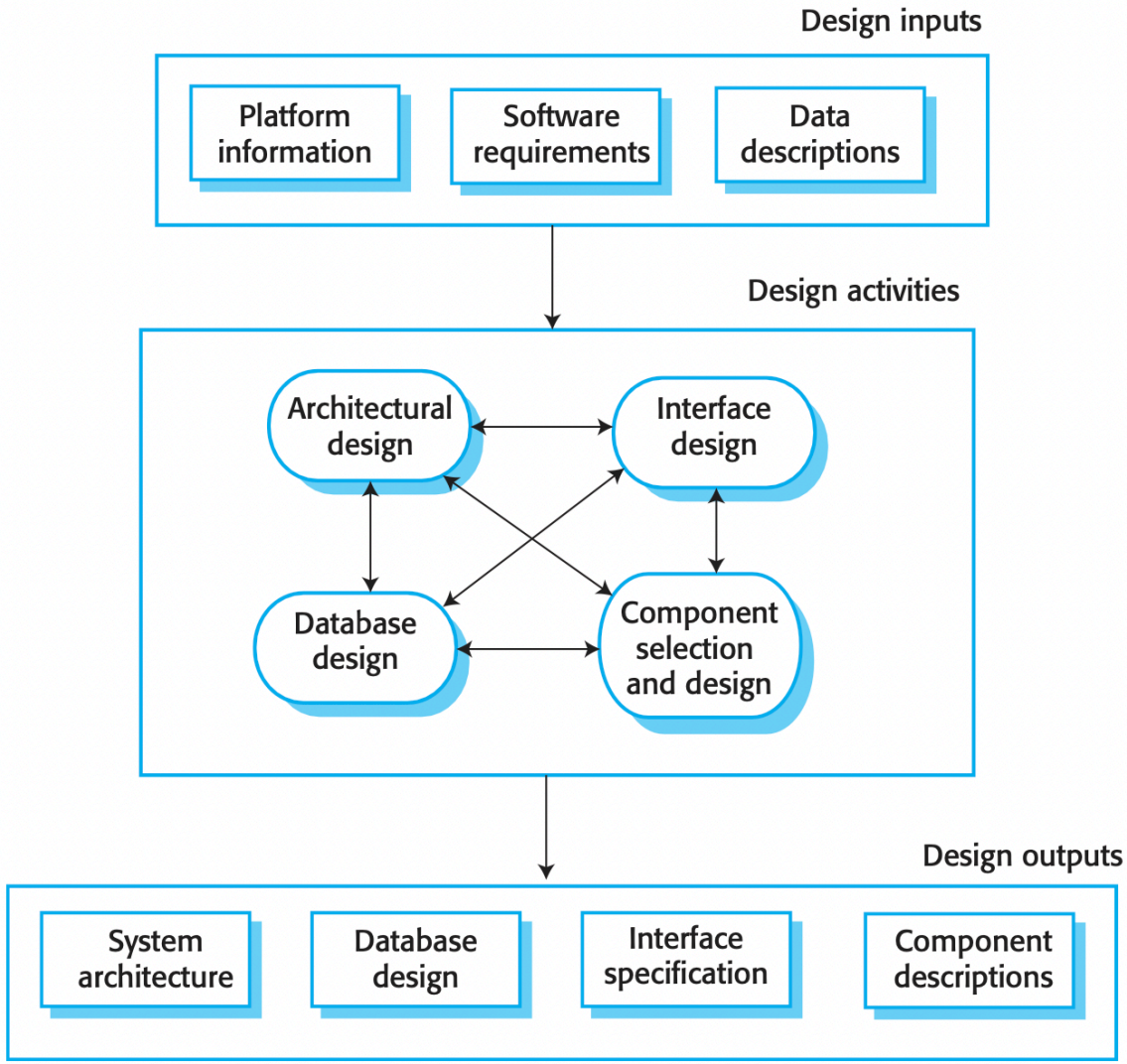
SOFTWARE DESIGN AND IMPLEMENTATION

A software design is a description of the structure of the software to be implemented, the data models and structures used by the system, the interfaces between system components and, sometimes, the algorithms used

- **Não se obtém logo o resultado final**
- **Se são processados dados pré existentes:**
  - Incluir descrição dos dados
- Se os dados não são “nativos” do software devem ser indicados com **input**

TEXTO

SOFTWARE DESIG



SOFTWARE DESIGN AND IMPLEMENTATION

- **Atividades que podem fazer parte do design process:**
  - Architectural design
  - Database design
  - Interface design
  - Component selection and design

SOFTWARE DESIGN AND IMPLEMENTATION

For critical systems, the outputs of the design process are detailed design documents setting out precise and accurate descriptions of the system

SOFTWARE DESIGN AND IMPLEMENTATION

Por onde começar o design?

Componentes que se conhece?
Componente que não se conhece?
Modelo de dados

SOFTWARE VALIDATION

- Verificação e validação (V & V)
- Verificar se o software:
  - Está conforme a especificação
  - Vai de encontro ao pretendido pelo utilizador

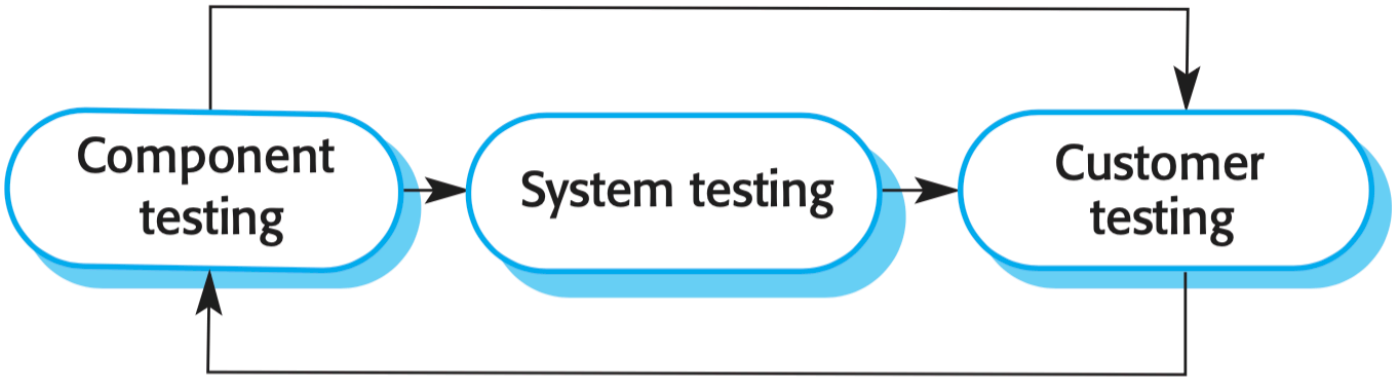
SOFTWARE VALIDATION

- **Como validar um Software**
  - Testes com dados simulados
  - reviews por parte do Cliente

SOFTWARE VALIDATION – TESTES

- **Os testes são feitos em três etapas:**
  - Component testing
  - System testing
  - Customer testing

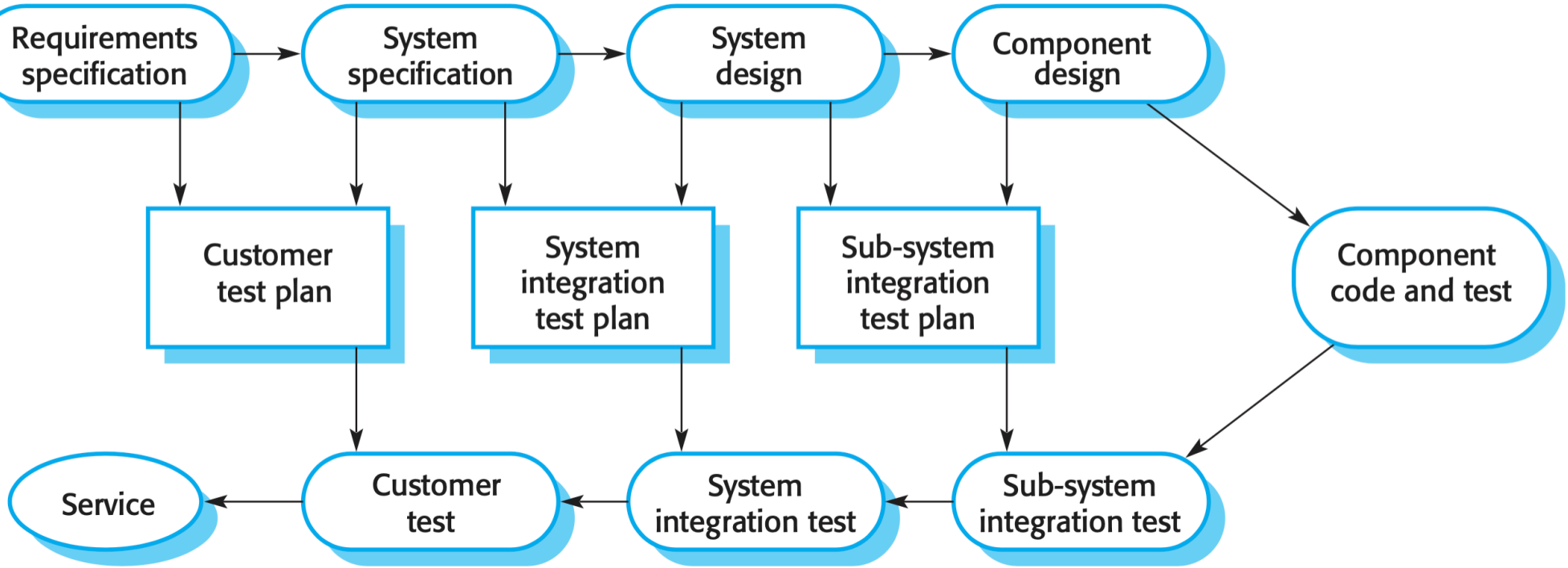
SOFTWARE VALIDATION



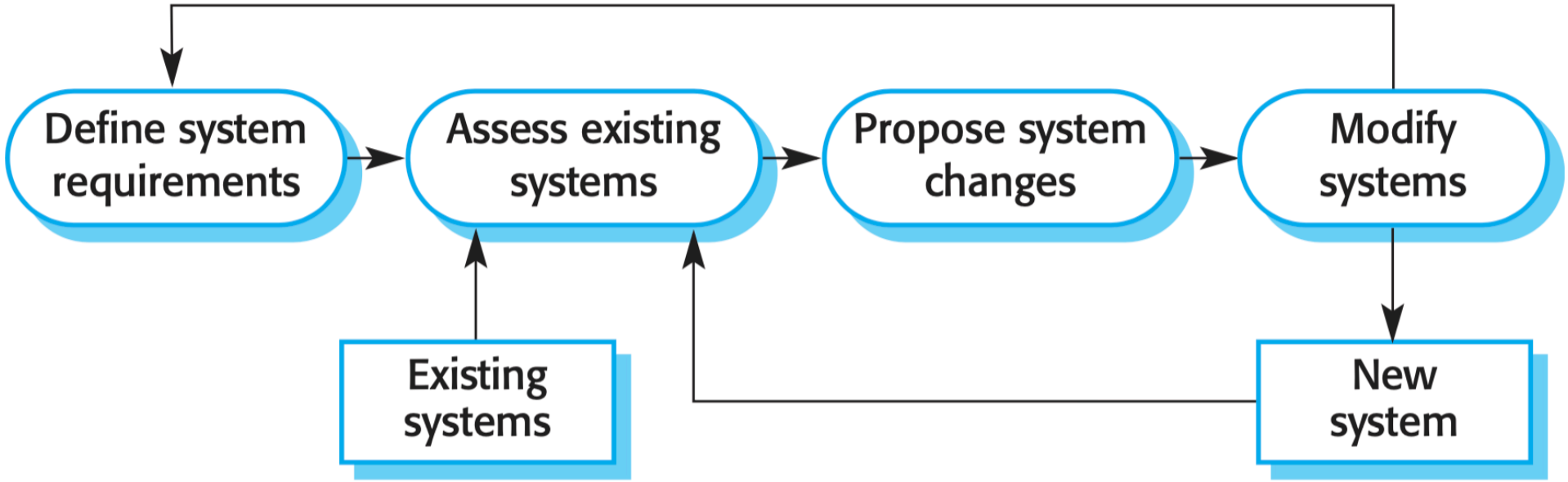
SOFTWARE VALIDATION

Para sistemas Críticos

SOFTWARE VALIDATION



SOFTWARE EVOLUTION



ENGENHARIA DE SOFTWARE

REQUIREMENTS ENGINEERING

CONTEÚDOS

- Requisitos Funcionais vs Requisitos não funcionais
- Requirements engineering processes
  - Requirements elicitation
  - Requirements specification
  - Requirements validation

- Requisitos do utilizador - normalmente definidos em linguagem natural + diagramas, deve indicar o que o sistema deve fazer assim como as suas restrições.

- Requisitos do Sistema - descrição mais detalhada das funções, serviços e restrições de um software.

User requirements definition

1. The Mentcare system shall generate monthly management reports showing the cost of drugs prescribed by each clinic during that month.

System requirements specification

1.1 On the last working day of each month, a summary of the drugs prescribed, their cost and the prescribing clinics shall be generated.
1.2 The system shall generate the report for printing after 17.30 on the last working day of the month.
1.3 A report shall be created for each clinic and shall list the individual drug names, the total number of prescriptions, the number of doses prescribed and the total cost of the prescribed drugs.
1.4 If drugs are available in different dose units (e.g. 10mg, 20mg, etc.) separate reports shall be created for each dose unit.
1.5 Access to drug cost reports shall be restricted to authorized users as listed on a management access control list.

REQUISITOS FUNCIONAIS VS REQUISITOS NÃO FUNCIONAIS

- Requisitos Funcionais - o que o software deve fazer, como reage a determinados inputs, como o sistema se comporta,

- Requisitos não funcionais - são as restrições que o software tem.

REQUISITOS FUNCIONAIS

- Descreve o que o software deve fazer
- Escritos em linguagem natural
- Requisitos do utilizador são feitos para serem entendidos pelos gestores
- Requisitos funcionais do sistema são escritos para o developer

REQUISITOS FUNCIONAIS – EXEMPLO

- A user shall be able to search the appointments lists for all clinics.
- The system shall generate each day, for each clinic, a list of patients who are expected to attend appointments that day.
- Each staff member using the system shall be uniquely identified by his or her eight-digit employee number.

REQUISITOS FUNCIONAIS

Imprecision in the requirements specification can lead to disputes between customers and software developers

REQUISITOS NÃO FUNCIONAIS

Property	Measure
Speed	Processed transactions/second User/event response time Screen refresh time
Size	Megabytes/Number of ROM chips
Ease of use	Training time Number of help frames
Reliability	Mean time to failure Probability of unavailability Rate of failure occurrence Availability
Robustness	Time to restart after failure Percentage of events causing failure Probability of data corruption on failure
Portability	Percentage of target dependent statements Number of target systems

REQUIREMENTS ENGINEERING PROCESSES

- The process
- Requirements elicitation
- Requirements specification
- Requirements validation

REQUIREMENTS ENGINEERING PROCESSES

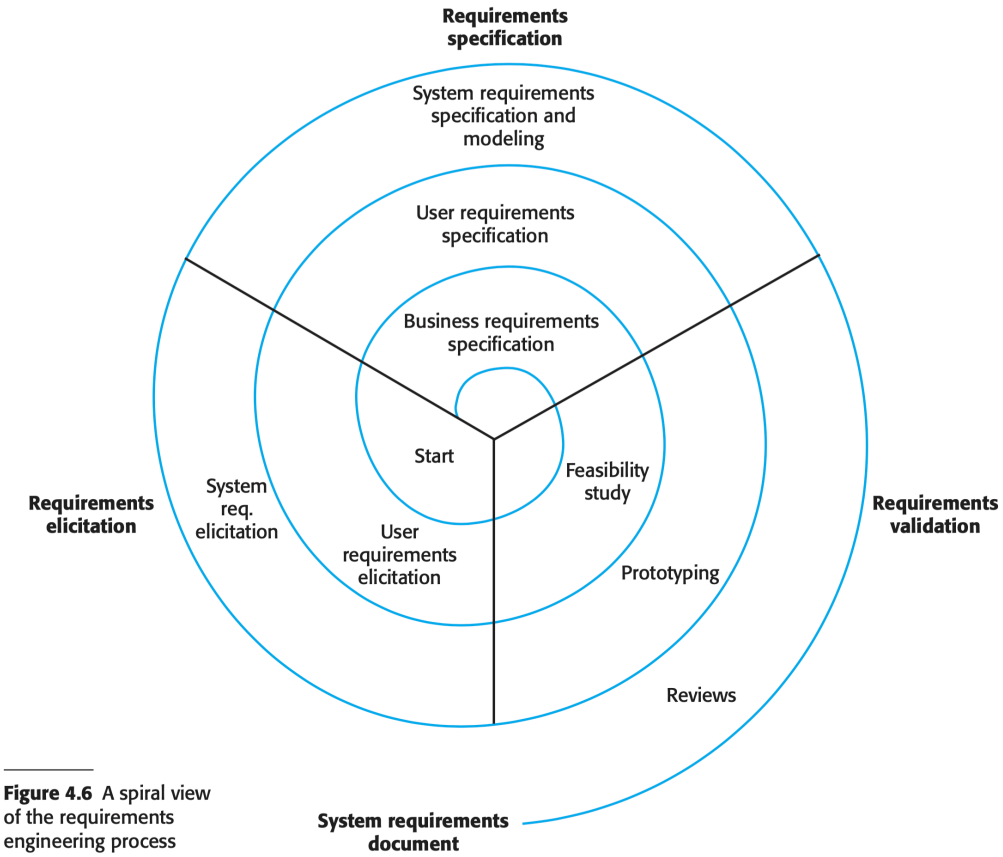
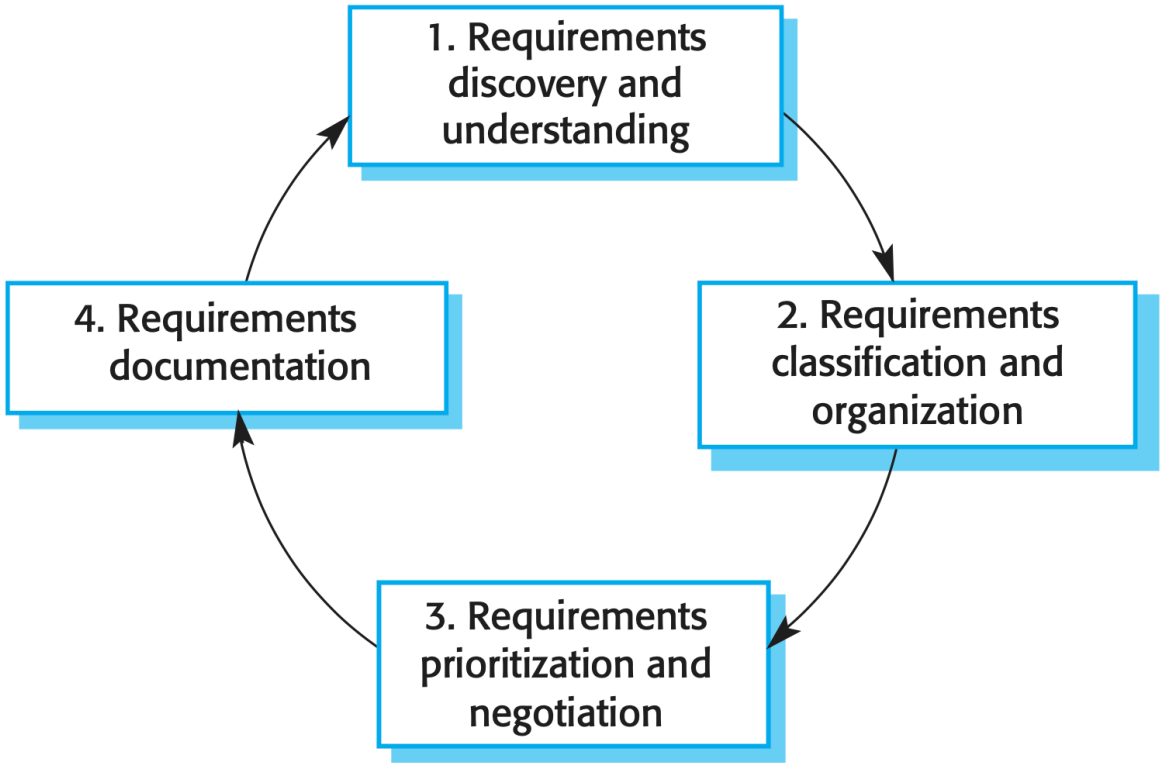


Figure 4.6 A spiral view of the requirements engineering process

ENGINEERING ELICITATION



ENGINEERING ELICITATION – COMO?

- Entrevistas
- Etnografia

ENGINEERING ELICITATION – STORIES AND SCENARIOS

- Descrição de alto nível do sistema

ENGINEERING ELICITATION – STORIES

Photo sharing in the classroom

Jack is a primary school teacher in Ullapool (a village in northern Scotland). He has decided that a class project should be focused on the fishing industry in the area, looking at the history, development, and economic impact of fishing. As part of this project, pupils are asked to gather and share reminiscences from relatives, use newspaper archives, and collect old photographs related to fishing and fishing communities in the area. Pupils use an iLearn wiki to gather together fishing stories and SCRAN (a history resources site) to access newspaper archives and photographs. However, Jack also needs a photo-sharing site because he wants pupils to take and comment on each other's photos and to upload scans of old photographs that they may have in their families.

Jack sends an email to a primary school teachers' group, which he is a member of, to see if anyone can recommend an appropriate system. Two teachers reply, and both suggest that he use KidsTakePics, a photo-sharing site that allows teachers to check and moderate content. As KidsTakePics is not integrated with the iLearn authentication service, he sets up a teacher and a class account. He uses the iLearn setup service to add KidsTakePics to the services seen by the pupils in his class so that when they log in, they can immediately use the system to upload photos from their mobile devices and class computers.

ENGINEERING ELICITATION – SCENARIO

Uploading photos to KidsTakePics

Initial assumption: A user or a group of users have one or more digital photographs to be uploaded to the picture-sharing site. These photos are saved on either a tablet or a laptop computer. They have successfully logged on to KidsTakePics.

Normal: The user chooses to upload photos and is prompted to select the photos to be uploaded on the computer and to select the project name under which the photos will be stored. Users should also be given the option of inputting keywords that should be associated with each uploaded photo. Uploaded photos are named by creating a conjunction of the user name with the filename of the photo on the local computer.

On completion of the upload, the system automatically sends an email to the project moderator, asking them to check new content, and generates an on-screen message to the user that this checking has been done.

What can go wrong: No moderator is associated with the selected project. An email is automatically generated to the school administrator asking them to nominate a project moderator. Users should be informed of a possible delay in making their photos visible.

Photos with the same name have already been uploaded by the same user. The user should be asked if he or she wishes to re-upload the photos with the same name, rename the photos, or cancel the upload. If users choose to re-upload the photos, the originals are overwritten. If they choose to rename the photos, a new name is automatically generated by adding a number to the existing filename.

Other activities: The moderator may be logged on to the system and may approve photos as they are uploaded.

System state on completion: User is logged on. The selected photos have been uploaded and assigned a status "awaiting moderation." Photos are visible to the moderator and to the user who uploaded them.