PROJECT

REQUIREMENTS GATHERING AND MANAGING YOUR TIME

Hamilton V. Niculescu

hamilton.niculescu@ncirl.ie

(please add 'Project' in the subject line when emailing me)

Aim and Objectives

Aim

To understand the move from analysis to design

Objectives

To understand

- Requirement analysis process
- Structured Natural Language specification
- Requirements document
- Requirements validation

Requirements engineering

The process of establishing the services that the customer requires from a system and the constraints under which it operates and is developed

Requirements engineering processes

Today we will look at a number of generic activities common to all Software Engineering processes:

1. Requirements elicitation and analysis
2. Requirements specification
3. Requirements validation

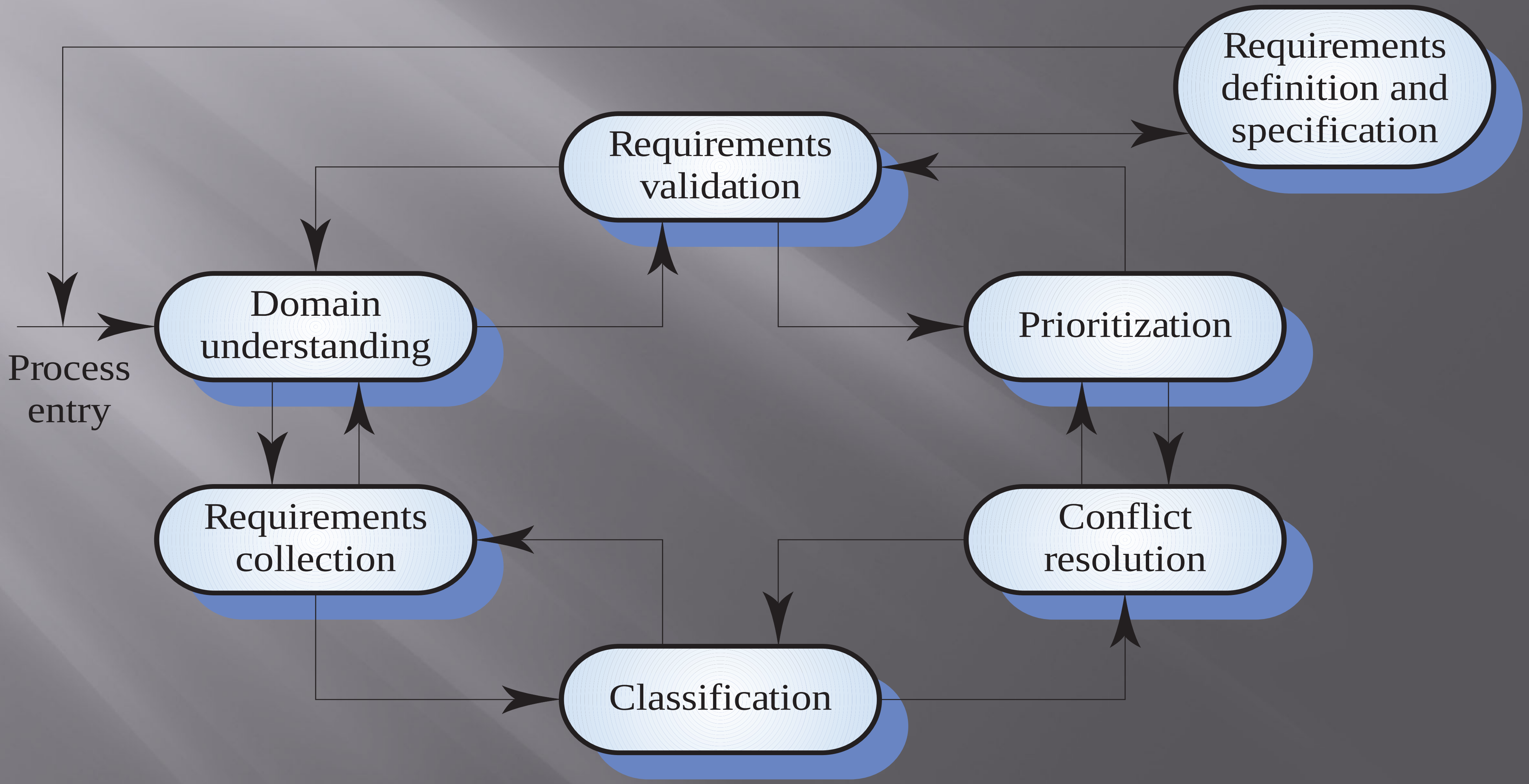
1. Requirements analysis

- Involves technical staff working with customers to find out about the application domain, the services that the system should provide and the system's operational constraints
- May involve end-users, managers, engineers involved in maintenance, domain experts, trade unions, etc. These are called stakeholders

Problems of requirements analysis

- Stakeholders don’t know what they really want
- Stakeholders express requirements in their own terms
- Different stakeholders may have conflicting requirements
- Organisational and political factors may influence the system requirements
- The requirements change during the analysis process. New stakeholders may emerge and the business environment change

The requirements analysis process



The requirements analysis process

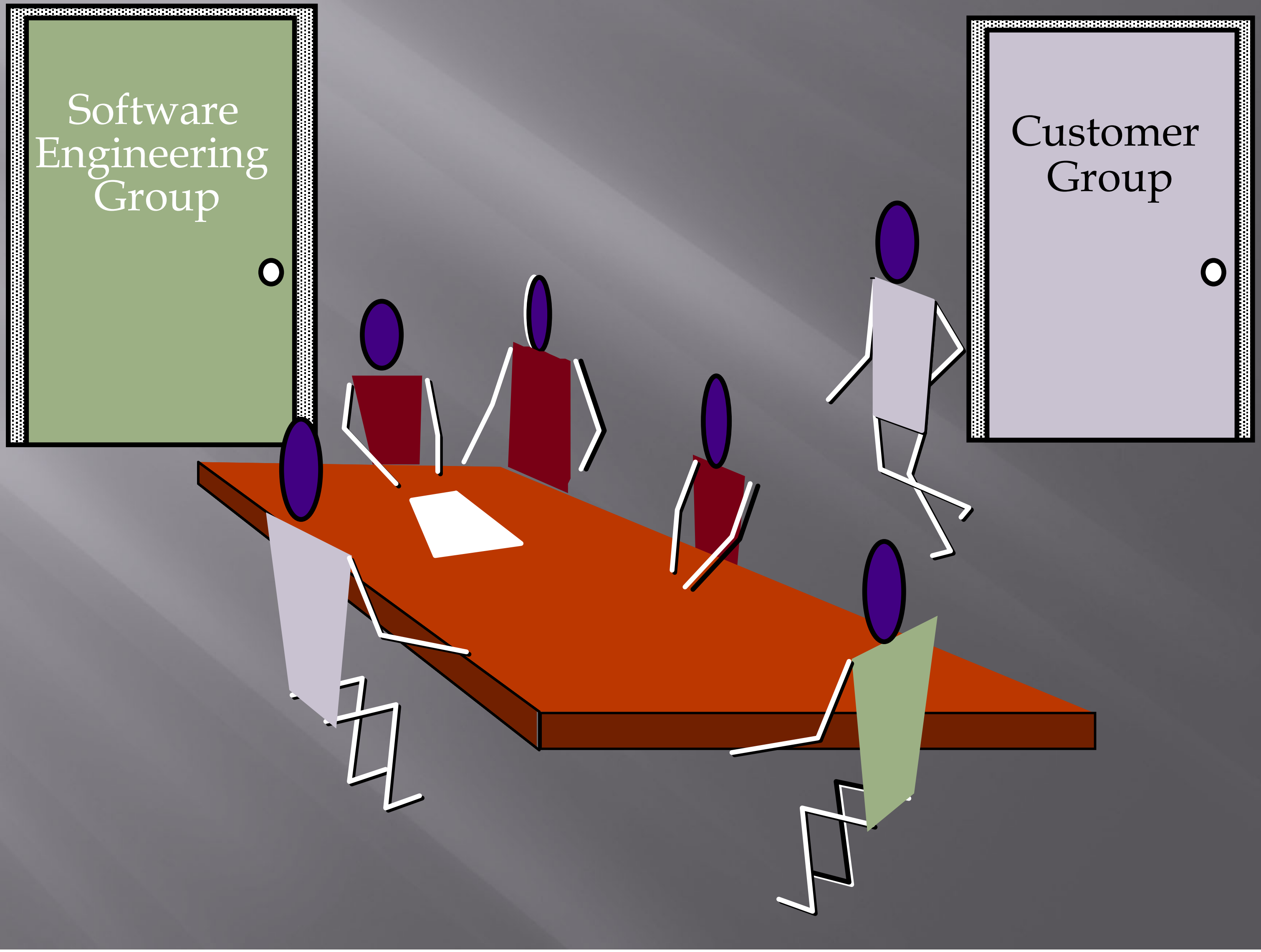
- Domain understanding
  - Analysts must develop their understanding of the application domain
- Requirements collection
  - This is the process of interacting with stakeholders in the system to discover their requirements
- Classification
  - This activity takes the unstructured collection of requirements and organises them into coherent clusters

The requirements analysis process

- Conflict Resolution
  - Inevitability, where multiple stakeholders are involved, requirements will conflict. This activity is concerned with finding and resolving these conflicts
- Prioritisation
  - In any set of requirements some will be more important than others
- Requirements checking
  - The requirements are checked to discover if they are complete, consistent and in accordance with what the stakeholders really want from the system

Requirements Gathering

Facilitated Application Specification Techniques



2. Requirements specification

User requirements specification

- User requirements are defined using natural language, tables and diagrams

Problems with natural language

- Lack of clarity - precision is difficult without making the document difficult to read
- Requirements confusion - functional and non-functional requirements tend to be mixed-up
- Requirements amalgamation - several different requirements may be expressed together

2. Requirements specification

Example of requirements confusion:

"A grid facility to be provided to assist with the positioning of entities on a diagram. The user may turn on the grid in either centimetres or inches, via an option on the control panel. Initially, the grid is off. The grid may be turned on and off at any time during an editing session and can be toggled between inches and centimetres at any time. A grid option will be provided on the reduce-to-fit view but the number of grid lines shown will be reduced to avoid filling the smaller diagram with grid lines"

2. Requirements specification

Guidelines for specifying user requirements

- Invent / adopt a standard format and use it for all requirements
- Use language in a consistent way.
  E.g. use "shall" for mandatory requirements,
  and "should" for desirable requirements
- Use text highlighting to identify key parts of the requirement
- Avoid the use of computer jargon

2. Requirements specification

System requirements

- System requirements are more detailed descriptions of the user requirements
- They are used by software engineers as the starting point for system design
- Natural language (NL) is often used to write system requirements specifications - as we know this isn't ideal

Alternative to NL specification

Structured Natural Language (SNL)

- A limited form of natural language may be used to express requirements
- This approach depends on defining standard forms or templates to express the requirements specification
- This removes some of the problems resulting from ambiguity and flexibility and imposes a degree of uniformity on a specification

PDL-based specification

- Requirements may be specified operationally using a Program Design Language (PDL), like a programming language, but with more flexibility of expression

- Most appropriate in two situations
  - Where a requirement is specified as a sequence of actions and the order is important
  - When hardware and software interfaces have to be specified

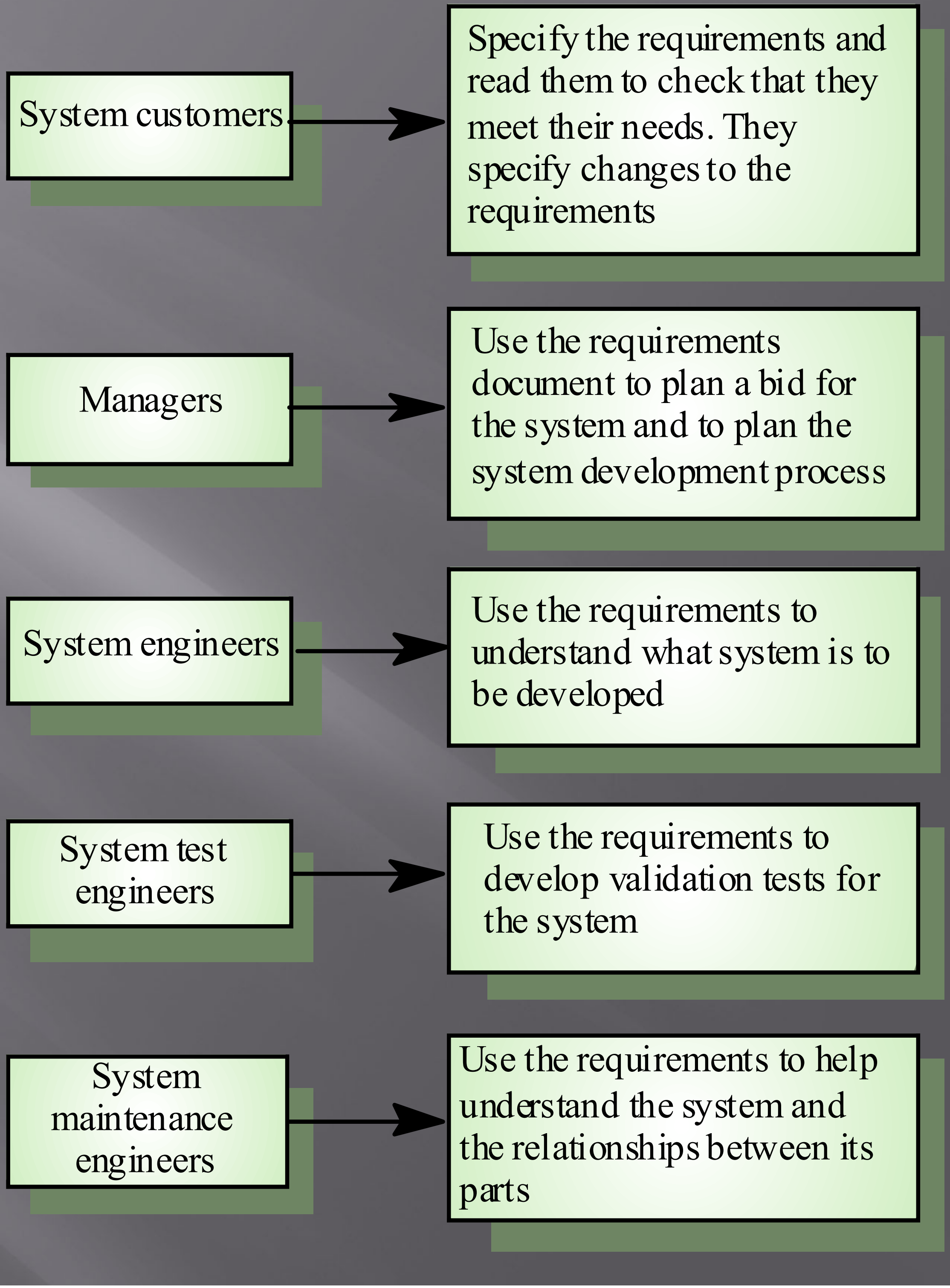
PDL disadvantages

- PDL may not be sufficiently expressive to describe the system functionality in an understandable way
- Notation is only understandable to people with programming language knowledge
- The requirement may be taken as a design specification rather than a model to help understand the system

The requirements document

- The requirements document is the official statement of what is required of the system developers
- Should include both a definition and a specification of requirements
- It is NOT a design document - it focuses on WHAT the system should do rather than HOW it should do it

Users of a requirements document



System customers

Specify the requirements and read them to check that they meet their needs. They specify changes to the requirements

Managers

Use the requirements document to plan a bid for the system and to plan the system development process

System engineers

Use the requirements to understand what system is to be developed

System test engineers

Use the requirements to develop validation tests for the system

System maintenance engineers

Use the requirements to help understand the system and the relationships between its parts

Requirements document

- Specify external system behaviour
- Specify implementation constraints
- Easy to change
- Serve as reference tool for maintenance
- Record forethought about the life cycle of the system i.e., predict changes

3. Requirements validation

- Concerned with demonstrating that the requirements define the system that the customer really wants

- Requirements error costs are high so validation is very important:
  - o Fixing a requirements error after delivery may cost up to 100 times the cost of fixing an implementation error

Requirements checking

During the requirements validation process, different types of checks should be carried out on the requirements document:

- **Validity checks**: Does the system provide the functions which best support the customer's needs?
- **Consistency**: Are there any requirements conflicts?
- **Completeness**: Are all functions required by the customer included?
- **Realism**: Can the requirements be implemented given available budget and technology
- **Verifiability**: Can the requirements be checked?

Requirements validation techniques

- Requirements reviews
  - The requirements are analysed systematically by a team of reviewers

- Prototyping
  - Using an executable model of the system to check requirements

- Test-case generation
  - Developing tests for requirements to check testability

TIME AND PROJECT MANAGEMENT

Identifying Project Goals

- What is the purpose of this project?
- What is the final goal/deliverable?
- What is required to ensure this project will succeed?
- What will happen if I don’t complete the project?

List Project Priorities

- What sections/features of this project are mission critical?
- What happens if a listed priority is not completed?

Time Management

Time Management is a difficult task to control for a project. There are two important factors that must be considered to use you time as efficiently as possible:

1) Have a good estimate on how long a given task should take to complete.
2) How efficiently do you manage your time?

Time Estimation and Planning

- The ability to accurately estimate the time required to complete a given task is essential for good project management
- It is very easy to underestimate the time required to complete a given task
- To get a good time estimate you must:

1) Fully understand the task to be completed
2) Estimate the time factoring in any research required, hardware/software setup, other priorities (e.g., CAs), unanticipated events, etc…
3) **Plan for the task going wrong**

How to Manage your Own Time

- Allocate time to your project and stick to this schedule
- Avoid distractions when working on your project
- Don’t continually check email, Facebook, etc. when you are working - this will take your focus from the task you are working on and will drain your time!!
- Document/log your progress - this can be a useful aid when you return to your project again

The 80/20 Rule

- 20% of the tasks are vital to the success of the project
- 80% are trivial

- Identify and focus on what is core to your project
- Make sure this task is completed to a satisfactory level

Planning

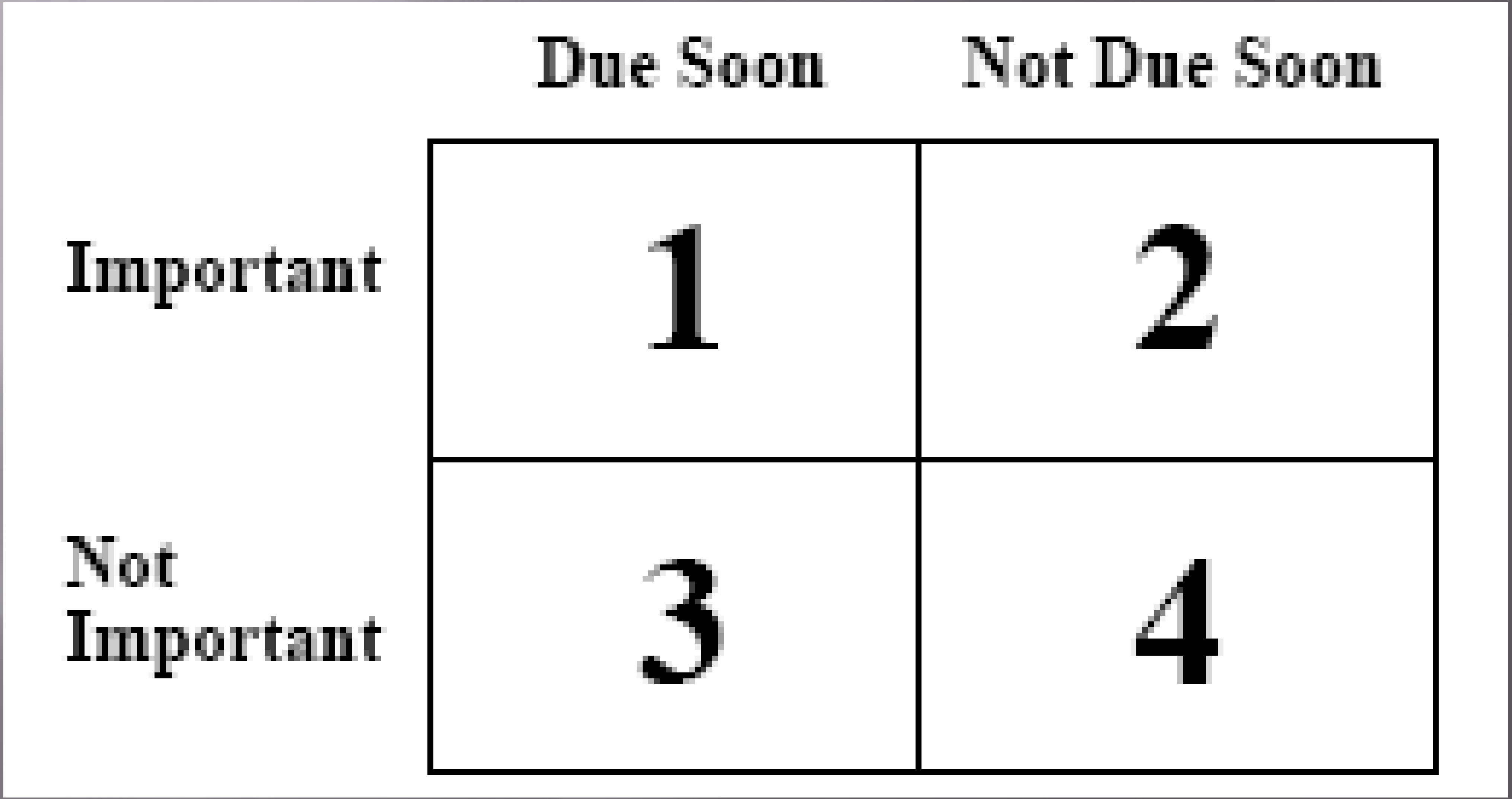
- Map out a schedule of project tasks to be completed and stick to it
- Plan every day
- Plan every week
- Set milestones to track progress

Managing Tasks

- Always break things down into smaller more manageable sub-tasks - this is especially relevant for programming and application development

- Complete the least desirable tasks first - this can be a good motivator to get a difficult task out of the way first!

The four-quadrant TO DO list



	Due Soon	Not Due Soon
Important	1	2
Not Important	3	4