

College of Humanities Master of Financial Technology- FinTech

Business Analysis

FNT613

9th Lecture

**Prepared By:
Dr. Abeir Moussa**





مجموعة جامعة
الجميع الذكية
للتعليم القابضة



DOCUMENTING AND MANAGING REQUIREMENTS & MODELLING REQUIREMENTS

9th lecture

FNT613

Prepared By:

Dr. Abeir Moussa

DOCUMENTING AND MANAGING REQUIREMENTS & MODELLING REQUIREMENTS



Week 10

Chapters- 11&12

Learning objectives

- Understand the importance of documenting requirements
- Learn the key elements of a requirement document
- Gain knowledge of different requirement elicitation techniques
- Learn how to prioritize and manage requirements
- Explore different requirement documentation formats
- Understand the role of stakeholders in requirement management
- Understand the purpose and benefits of requirement modeling
- Learn different requirement modeling techniques
- Gain knowledge of modeling tools and software
- Understand the process of creating requirement models
- Gain knowledge of best practices for requirement modeling

Chapter outline

1. Documenting and Managing Requirements
2. Modelling Requirements

- **INTRODUCTION**
- **THE REQUIREMENTS DOCUMENT**
- **THE REQUIREMENTS CATALOGUE**
- **MANAGING REQUIREMENTS**



1. Documenting and Managing Requirements

INTRODUCTION

This part of knowledge is concerned with two of the key elements of requirements engineering ;

- documenting the requirements that have been gathered, and
- managing those requirements in such a way that they can be traced through the business change process, from source to delivery.

The importance of Documentation

Documenting requirements clearly is vital for the success of a project. the documentation provides the business managers and staff, who are the sources and owners of the requirements, with a firm basis for validating that the documentation accurately records what they need the solution to provide. For example, in the maintenance of IT systems and during benefits realization.

THE REQUIREMENTS DOCUMENT

➤ **Structure**

The requirements document has to provide the basis for the solutions to be delivered to the organization so it needs to be well formed and clear.

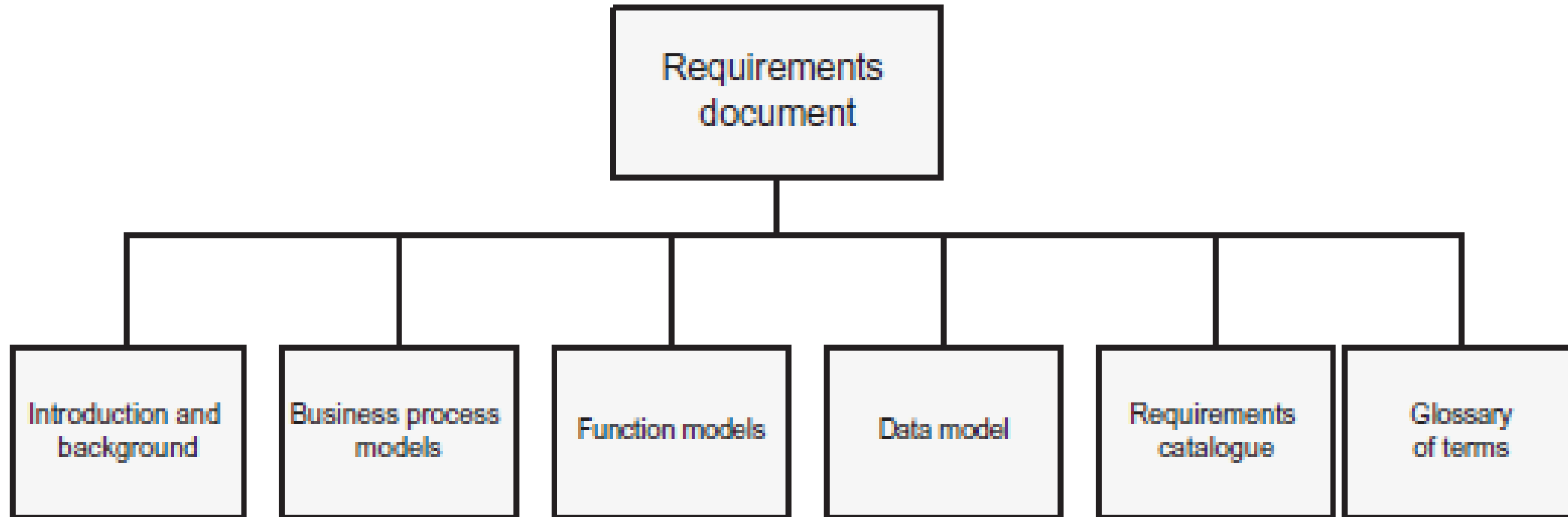
Whatever the basis used for partitioning, an organized requirements document will make reviewing much easier and help to ensure that the agreed requirements are accurate.

➤ **Content of the requirements document**

Figure 11.1 Contents of a requirements document

THE REQUIREMENTS DOCUMENT

Figure 11.1 Contents of a requirements document



THE REQUIREMENTS DOCUMENT

Introduction and background- Typically, business requirements involve changes to the business processes and any new or enhanced software solution must support the business process changes. If the requirements catalogue contains lists of data within individual requirements, these lists are extremely difficult to review and evaluate for correctness.

Business process models- The 'as is' processes that are to be revised may also be included here

Function models.- These diagrams provide an excellent overview of the solution and provide a means for structuring the requirements catalogue.

Data model- provides a far superior view of the data than textual descriptions and enables the analyst to consider the exact business rules and requirements to be met by the solution. For example, a list of assumptions or decisions made can often be found in a requirements document. Requirements catalogue. The information about each individual requirement should be documented in the requirements catalogue.

Glossary of terms. One of the key quality characteristics for the requirements document is to ensure that it provides a clear definition of the requirements

THE REQUIREMENTS CATALOGUE

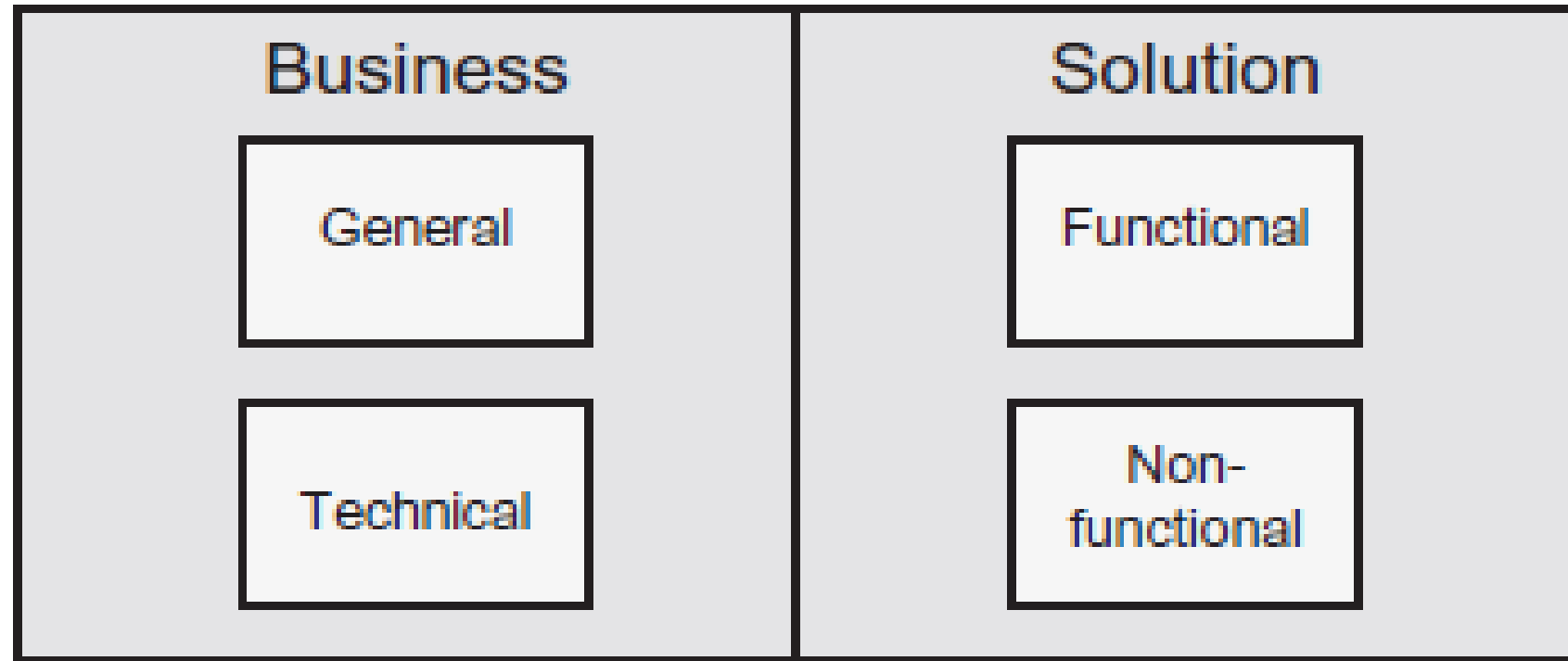
When requirements are initially elicited, they are not organized and it is only once the requirements analysis activity takes place that they are structured and formed into an organized set.

There are a number of ways of organizing requirements but fundamentally, a hierarchical approach will provide the easiest structure for navigating and reviewing the requirements.

Figure 11.2 shows the four types of requirements: general, technical, functional and non-functional.

THE REQUIREMENTS CATALOGUE

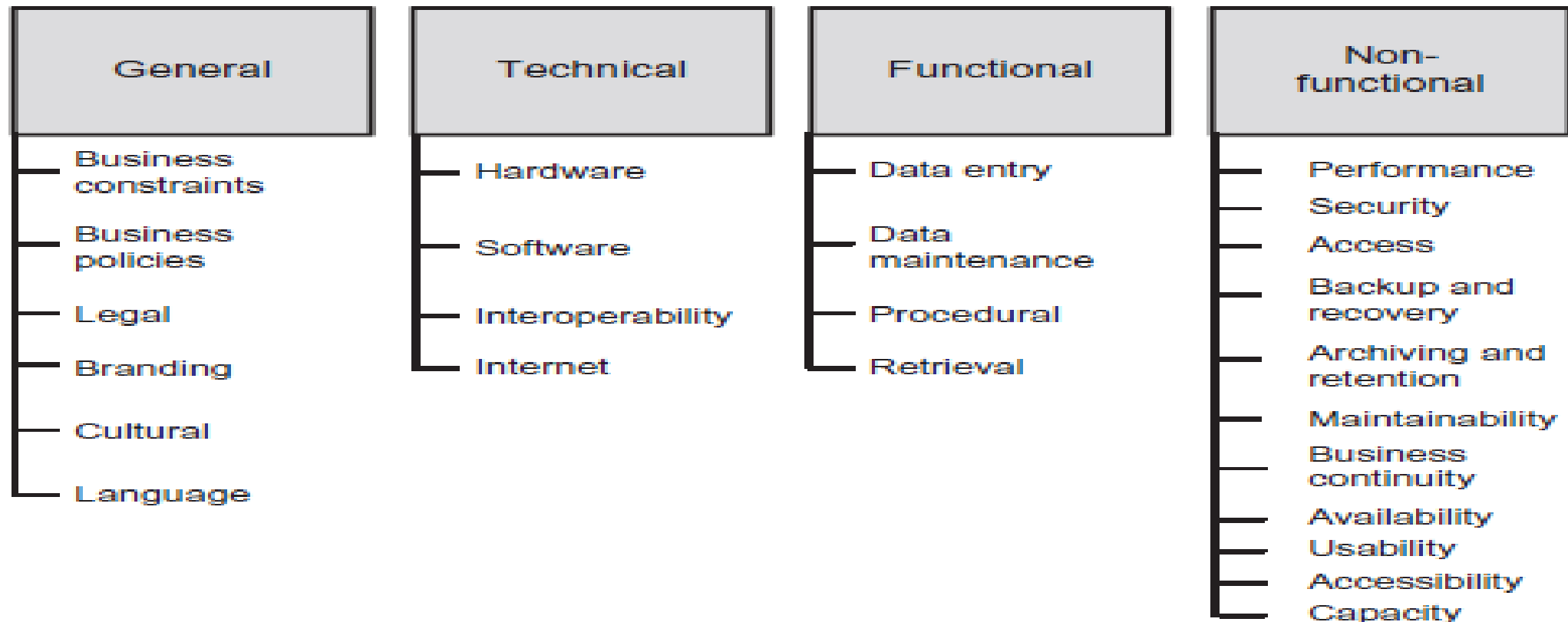
Figure 11.2 Types of requirements



THE REQUIREMENTS CATALOGUE

➤Types of Requirements

Figure 11.3 Categories of requirements



THE REQUIREMENTS CATALOGUE

❑ General requirements

There are specific sub-categories of general requirement:

- **Business constraints** – these cover aspects such as budget, timescale and resources.
- **Business policies** – these cover aspects such as standards and business policy decisions (often known as business rules).

It may define specific areas such as the environmental policy or the strategy to be adopted to ensure business continuity

THE REQUIREMENTS CATALOGUE

- **Legal** – these are the requirements that state relevant legal and regulatory constraints. These requirements may relate to organizations in many business sectors, for example the Data Protection Act applies to any organization that holds personal data, or may be specific to a business sector or industry, for example, the Sarbanes–Oxley Act of 2002 and Basel III Accord which apply in the financial sector.
- **Branding** – these requirements are concerned with the image and style to be promoted for the organization.
- **Cultural**- This may mean that the requirements set out the vision for the organization, the approach taken to dealing with customers and the management style.
- **Language** – many organizations have specific language requirements as they are operating across international boundaries and languages. These requirements set out the languages to be used in the organization and the ways of communicating with customers and other organizations.

THE REQUIREMENTS CATALOGUE

❑ Technical requirements

These are the requirements that state the technical policies and constraints to be adopted across the organization and apply to a range of change projects.

The specific sub-categories of technical requirements are:

- **Hardware requirements-** covering aspects such as the make and model of hardware equipment to be used in the organization.
The requirements may cover the IT hardware but can also include equipment relating to the work of the organization such as production or general office equipment.
 - **Software requirements-** covering areas such as operating systems, software applications, networking and communications software.
 - **Interoperability requirements-** that cover the standards for communicating between systems where they are required to exchange data.
- Internet requirement** - These requirements relate to the technical policies governing the organization's use of the internet and web-enabled services.

THE REQUIREMENTS CATALOGUE

❑ Functional requirements

The functional requirements set out the features that any solution should provide.

Some requirements may be better delivered by non-IT means, for example, a personal communication may be more effective. The functional requirements form a key element of the requirements catalogue because they define the specific features to be delivered by a solution.

Also, whereas the **general** and **technical** requirements tend to have a certain longevity, the functional requirements can be subject to frequent change particularly as they are elaborated to a more detailed level.

Also, it may be the case that while some of the requirements may evolve during the iterative development, there may be a part of the system that requires the application of complex business rules; in this case, it would be very sensible to define these clearly up front. If the requirements are to be managed closely, certain characteristics – such as the identifier and source – need to be documented so that each requirement can be tracked through the development and delivery of the business solution.

THE REQUIREMENTS CATALOGUE

When eliciting and investigating functional requirements, it can be useful to consider several categories:

- **Data entry requirements-** are concerned with gathering and recording the data that is required in the solution. solution.
- **Data maintenance requirements-** handle changes to the data used within the Procedure requirements.
- **Procedure requirements.** These requirements need to be defined clearly so that they can be adopted accurately within the solution.
- **Retrieval requirements-** These requirements are concerned with requests about the data and include the provision of specified reports and responses to enquiries. Management and operational information requirements can be numerous and wide-ranging.

THE REQUIREMENTS CATALOGUE

❑ Non-functional requirements

The non-functional requirements are concerned with how well the solution will operate and answers questions such as: 'how quickly will it respond?' and 'how easy will it be to use?' **There are several areas of non-functional requirement to consider:**

- **Speed of performance-** These requirements are usually related to the security requirements as they define the stakeholder groups and their levels of access to the data.
- **Level of security-** The majority of organizations handle information and data of varying levels of confidentiality.
- **Backup and recovery-** The requirements for secure storage and access of information are also linked to these requirements.
- **Archiving and retention-** These requirements define aspects such as the length of time of the retention, the nature of the archiving approach and the approaches to be taken to the disposal of information and data.
- **Maintainability requirements-** These will include aspects such as servicing, problem investigation and correction.

THE REQUIREMENTS CATALOGUE

- **Business continuity requirements** may include those to prevent a disaster affecting the organization to any great extent – for example, the need for duplicate installations –and also contingency requirements should a disaster occur.
- **Availability-** These requirements cover areas such as the volumes of data or images to be stored, the volumes of transactions to be processed and the number of stakeholders to be supported.
- **Usability.** This area concerns the ease with which a stakeholder can learn, apply and use new processes and systems.
- **Accessibility.** The accessibility requirements are related to the usability requirements but they focus specifically on enabling access for all, regardless of infirmity or disability.
- **Capacity.** These requirements cover areas such as the volumes of data or images to be stored, the volumes of transactions to be processed and the number of stakeholders to be supported.

THE REQUIREMENTS CATALOGUE

The non-functional requirements are areas that are often left until later or even dismissed at an overview level without clear thought and analysis.

Some of the general and technical requirements are elaborated and expanded in the non-functional and functional requirements;

some functional requirements are related to non-functional requirements. If a requirement is not aligned with these then it needs to be explored further as it is questionable that it is really a requirement. Understanding the hierarchy of requirements helps ensure that the requirements are consistent and coherent.

When eliciting non-functional requirements, the business and technical policies help give insights into why these requirements are necessary at the level of performance stated. For example, data protection legislation defines the principles to be adopted by any organization that stores personal data.

Further, the functional requirements that define the data requirements will be linked to the security and legal requirements for this area.

THE REQUIREMENTS CATALOGUE

❑ Hierarchy of requirements

The hierarchy of requirements, linking functional and non-functional requirements back to the general and technical business requirements, provides a means of tracing the original business need for the requirements.

Initially, the description may be at an outline level and elaborated in more detailed versions of the requirement documentation.

When describing requirements, it is good practice to adopt the following structure:

• **actor** (or user role) • verb phrase • object (noun or noun phrase).

An example functional requirement is: 'the receptionist shall be able to view the customer name, address and telephone number'.

THE REQUIREMENTS CATALOGUE

An example general requirement is: ‘the solution shall comply with the provisions of the Data Protection Act’. The originating person or information source for the requirement.

For example, a stakeholder may have identified the requirement during an interview or other discussion, or there may be an earlier document – such as a project brief or feasibility study – that includes some of the business requirements. The business stakeholder who can make decisions regarding the requirement.

The type of requirement may be defined at sub-category level, for example, a requirement may be general, legal. The level of priority of the requirement.

THE REQUIREMENTS CATALOGUE

☐ Documenting a requirement

Each requirement is documented to define clearly what is required. The exact set of characteristics used to define requirements will depend upon the organizational standards and development approach to be adopted. (page 214- 217)

Alternatively, a more detailed approach may be useful and the name of the business process or use case may be used. The job roles or names of any stakeholders with a particular interest in the successful resolution of this requirement, and the details of their interest. Identifying stakeholders and their interests for each requirement provides a useful prompt to the business analyst to ensure that all relevant stakeholders' interests have been considered. Some functional requirements are associated with specific non-functional requirement.

THE REQUIREMENTS CATALOGUE

For each requirement, we should consider how we can check or measure if a requirement has been met.

Related requirements, Related documents, Comments, Rationale, Resolution
The identifiers of any requirements that are related to this requirement. They may be related for several reasons: there is a higher-level business requirement that provides further business information or justification for a functional or non-functional requirement; there are non-functional requirements concerning areas such as usability or security that affect functional requirements or vice versa; there are other requirements that concern a similar general, technical, functional or non-functional area.

THE REQUIREMENTS CATALOGUE

Another form of documentation that may be linked to the requirements is the modelling documents that have been created for the business change project. Some of these models may be contained within the requirements document however, it is still useful to show where there are requirements that are related to them. Additional comments that the analyst feels are useful to document for a particular requirement. The business justification for the requirement.

The level of detail of the definition will depend upon several factors:

- The stage of the analysis, is it an initial view of the requirements or a more detailed
- The nature of the solution, for example, a business process change or IT system requirements specification? replacement.
- The level of priority of each requirement. **For example**, if a requirement is allocated a priority, the detailed work should be deferred until the point where it is to be included in the solution.

THE REQUIREMENTS CATALOGUE

- The approach to be adopted to deliver the solution, for example, evolutionary system development or off-the-shelf software purchase. Some aspects of a requirements catalogue definition will emerge earlier than others; initially, we may only document the identifier, name, description, source and author.

After the requirements catalogue has been structured and duplicate or over lapping requirements removed, features such as the related requirements will be stated. Cross-referencing to other documents or models may be done late in requirements analysis.

There solution of a requirement may only be entered once the requirements have been validated and this may be subject to change if a MoSCoW prioritization approach is used.

THE REQUIREMENTS CATALOGUE

The version history will only be required if the requirement changes. The requirements catalogue is a central document throughout a business change project.

In this situation, the requirements catalogue should be clear about what is required and leave the further detail to be explored using approaches such as scenarios and prototyping.

MANAGING REQUIREMENTS

A failure to understand, document and manage requirements often lies at the heart of problems with business and IT system changes projects.

It answers the question ‘what happened to this requirement?’ and should show that each requirement has been resolved satisfactorily. Vertical traceability concerns tracing a requirement up or down the hierarchy so answers the questions about alignment with business values, strategy and objectives.

Requirements management is essential if there is to be traceability of requirements.

There are several elements involved in managing requirements as shown in **Figure 11.4 and described below.**

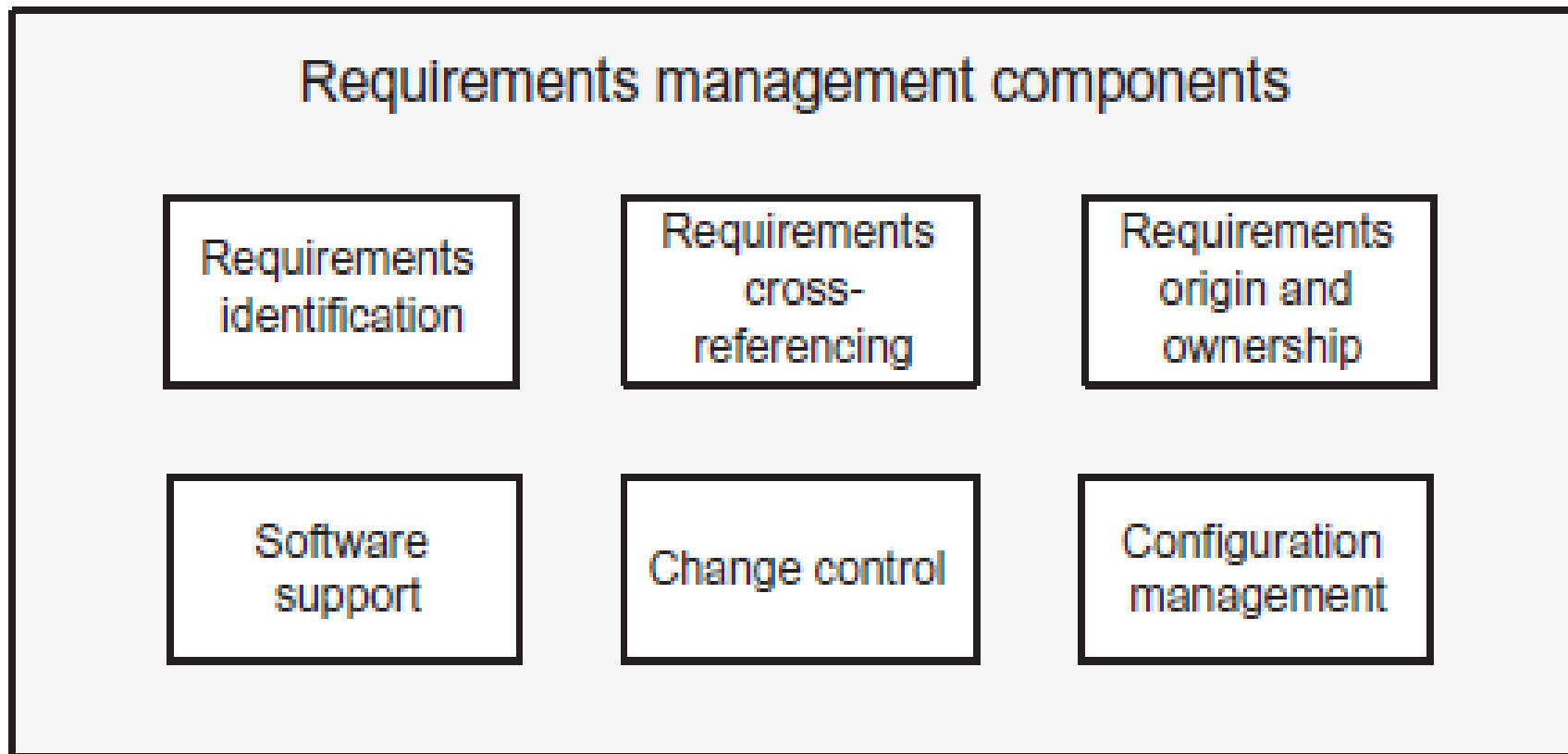
❑ Requirements identification

Each requirement needs to be identified uniquely in order that any reference to that requirement corresponds to only one requirement.

Cross-referencing- All related requirements and documents should be cross-referenced so that further elaboration or information concerning the requirement can be accessed easily.

MANAGING REQUIREMENTS

Figure 11.4 Elements of requirements management



MANAGING REQUIREMENTS

They allow the analyst to identify which requirements are related to the requirement that is the subject of the change, and consider whether the change will affect the other requirements.

- **Origin and ownership-** The source identifies the origin of the requirement. When considering changes to the requirement, the source can help to clarify the impact of the change and as a result help with the decision about the change. The owner typically has responsibility for the business area affected by the requirement and will need to make decisions about the requirement.

In addition, fundamental business changes such as budget reductions or time scale changes may require requirements to be discontinued, reprioritized or delayed; the owner will be the senior person in the decision-making process for the requirement and will have the ultimate authority concerning the requirement.

MANAGING REQUIREMENTS

Configuration management- is concerned with controlling any changes made to project deliverables, such as documents, in order to ensure that the changes are made in a disciplined manner and traceability sustained. Without effective configuration management the requirements document can become inaccurate as the following problems can occur:

- inability or difficulty in identifying the latest version of a requirement;
- the reintroduction of out-of-date requirements;
- using the wrong set of requirements for further development or testing work.

As a result, it is important that appropriate mechanisms are developed that will manage the implementation of any changes to the requirements.

MANAGING REQUIREMENTS

During requirements management, the deliverables will include the individual requirements catalogue entries, the composite set of entries that form the requirements catalogue, the models that elaborate and define the requirements, the requirements document.

For example, while a requirement will have a version number, so will the whole requirements catalogue within which the requirement sits.

MANAGING REQUIREMENTS

- **Configuration control**

There will also need to be a process for controlling the **Configuration Items (CIs)** and ensuring that they are not changed without formal approval and version numbering.

For example, a draft requirement catalogue entry with an identifier T-007 could be numbered T-007v0.3 indicating that this is the third draft version of the technical requirement numbered 007. Second, requirements validation is carried out to review and agree the requirements documentation.

At this point, the CIs that describe the requirements are said to be 'base lined'. This means that they are brought under configuration control and cannot be changed by any one without the following the formal configuration management procedure.

Sometimes it can be useful to baseline a requirements document prior to requirements validation, when rigorous requirements analysis has been carried out and a formal approach to change control is felt to be beneficial. Third, once a CI is baselined, no changes can be made to the content without the approval of the configuration manager.

MANAGING REQUIREMENTS

❑ Configuration management in an Agile environment

Having baselined the requirements document reasonably early in the development project, this document will provide an audit trail for the high-level requirements.

Baselining and controlling the versions of the prototypes will also be important if the requirements are to be managed. *There are several possibilities for baselining the prototypes during Agile software development including:*

- **Baselining every prototype before demonstration:** this has the virtue of clarifying the version that has been demonstrated to the business users.
- **Baselining daily:** this is highly disciplined but can prove onerous and unnecessary.
- **Baselining at the end of a time box:** this is fine if the time boxes are reasonably short – a few days perhaps – but less sensible with longer time boxes or, for example, the 21-day ‘sprints’ used within Scrum. Each baselined prototype is placed under configuration control as this is the only way to ensure that the up-to-date version of the prototype is used.

MANAGING REQUIREMENTS

❑ Change control

- **Documenting the proposed change** - The change should be documented as a 'change request', stating who raised the change, a description of the change and a justification for requesting it. •
- **Consulting the stakeholders.** Each change request should be sent to representative stakeholders to assess the impact of the proposed change, including the effort to make the change and the corresponding cost.
- **Deciding on the change-** Each change that has been applied to the requirements is documented and explains why one version was changed to create the new version.

Over time, a complete audit trail will be created explaining what actions were taken, why this was done and when.

MANAGING REQUIREMENTS

❑ Software support

Most requirements documents contain too many requirements to manage the set of documentation, cross-references and versions manually so an automated tool is usually required.

The allocation of identifiers and version numbers to configuration items. Some specialist tool sets provide a range of features, including those above, designed to support rigorous and efficient requirements management.

- **Introduction**
- **MODELLING BUSINESS USE CASES**
- **MODELLING REQUIREMENTS**
- **MODELLING SYSTEM DATA**
- **CLASS MODELS**
- **MODELLING IN AGILE APPROACHES**
- **THE USE OF MODELS IN SYSTEM MAINTENANCE**



2. MODELLING REQUIREMENTS

2. MODELLING REQUIREMENTS

INTRODUCTION

The techniques we describe here have been selected from two distinct approaches to systems modelling: use case diagrams and class modelling from the **Unified Modelling Language(UML)**, and entity relationship modelling from the family of structured, data-driven approaches.

The selected techniques model two distinct views of the IT system: the functions that the system will provide and the data to be stored within the system.

MODELLING BUSINESS USE CASES

A top-level view of an organization or business system can be provided through developing a business use case diagram.

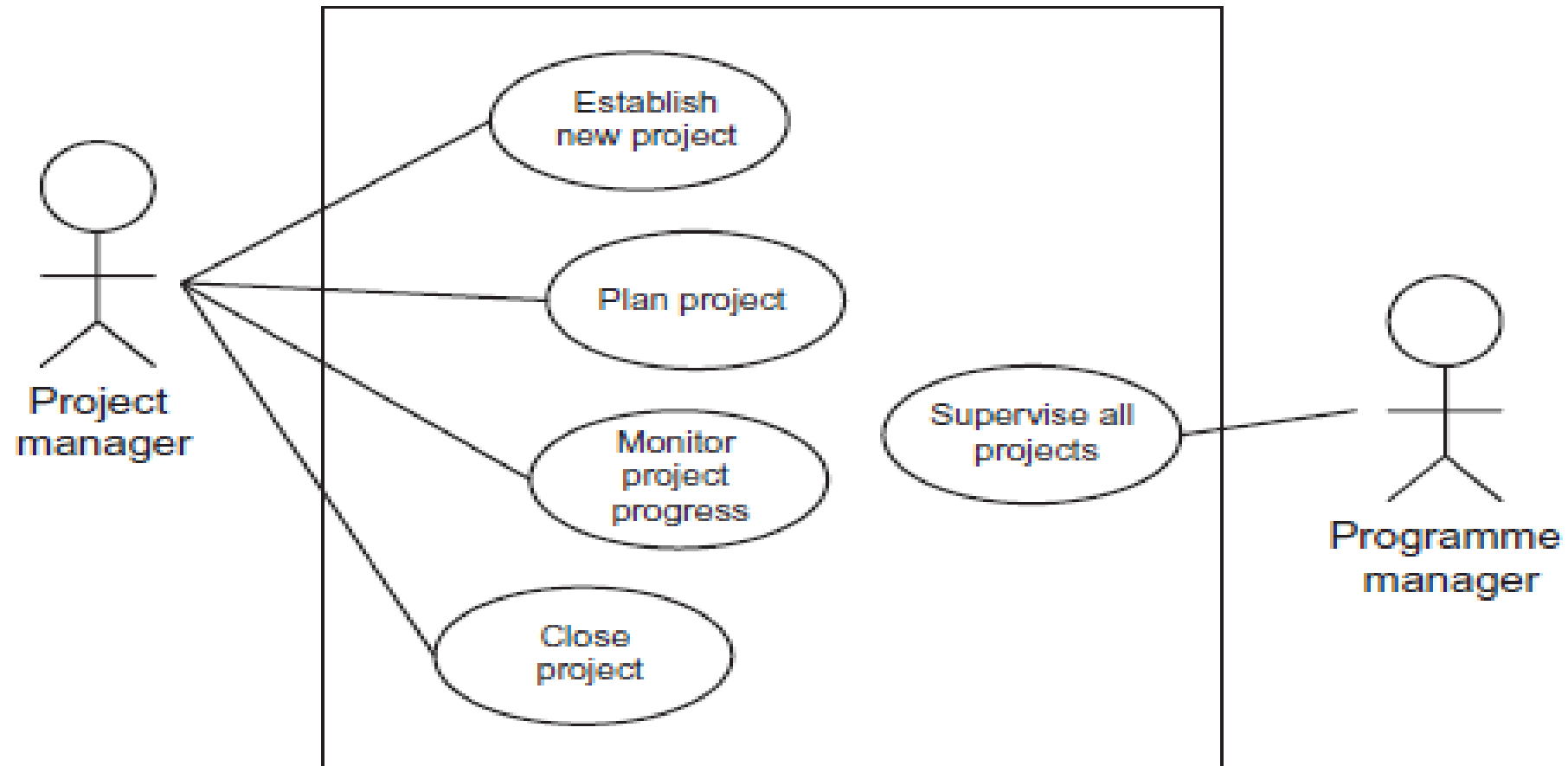
An example is shown in Figure 12.1; there are just two actors, a project manager and a program manager, and the use cases represent what they require of the business system. Business use case diagrams can be extremely useful when scoping a project or gaining an overview of the areas to be investigated and analyzed.

Some of the use cases on a business use case diagram may be fulfilled by an IT system but for others this may be only partially the case, and for some the solution may be completely manual.

Some of the business use cases shown in Figure 12.1 are taken down to the system level in Figures 12.2 to 12.4.

MODELLING BUSINESS USE CASES

Figure 12.1 Use case diagram for a project management organisation



MODELLING BUSINESS USE CASES

The saying ‘one picture is worth ten thousand words’ applies directly to the definition of IT system requirements.

For example, a function might be ‘Record customer’ and the actions here would include the following:

- accept the customer details;
- validate the customer details;
- store the customer details that have been entered.

In the **Unified Modelling Language (UML)**, a use case is something that an actor wants the IT system to do; it is a ‘case of use’ of the system by a specific actor and describes the interaction between an actor and the system.

MODELLING REQUIREMENTS

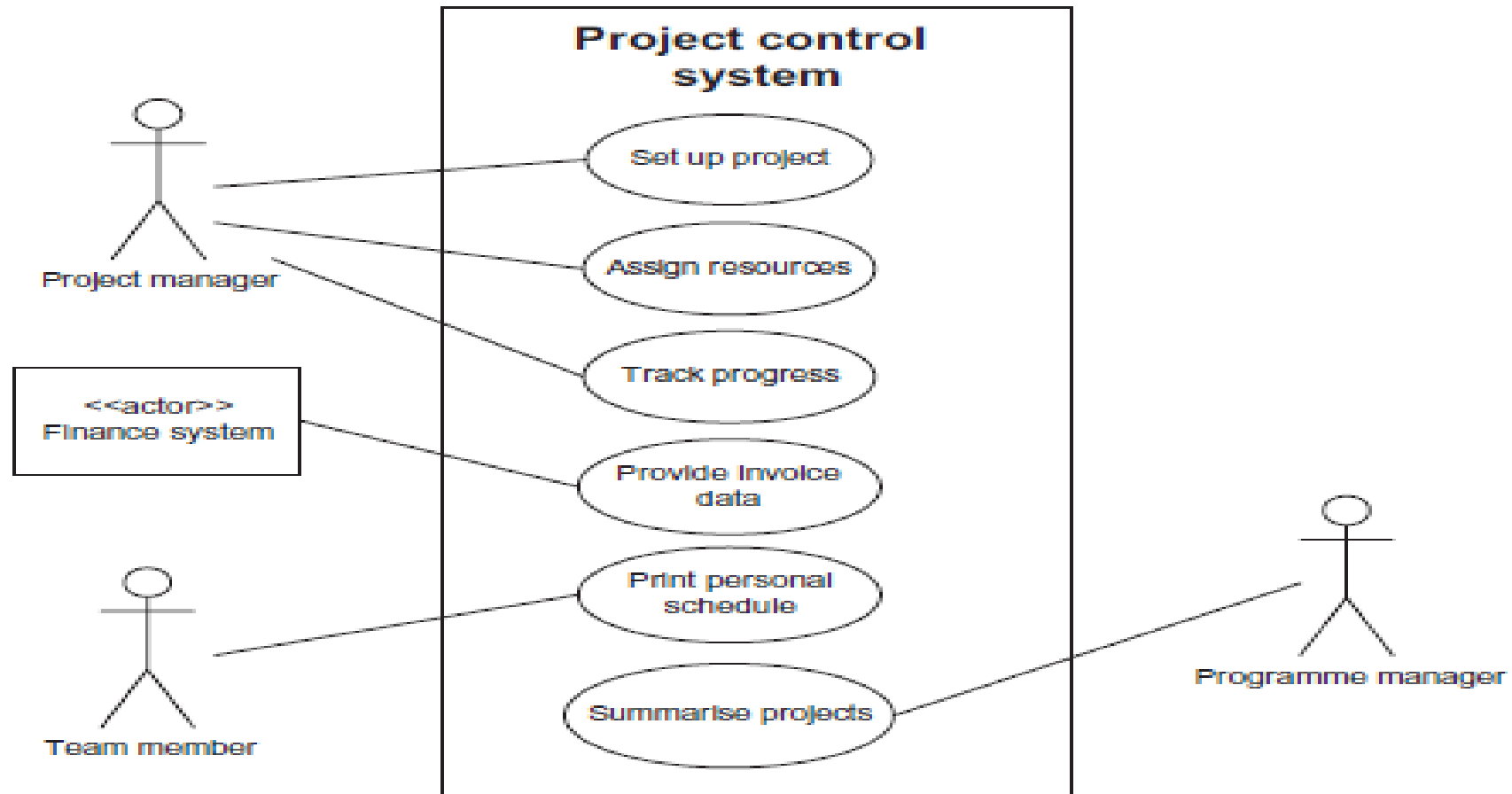
The use case model will consist of a diagram showing the actors, the use cases and the associations between them, plus a set of use case descriptions. Lines are drawn linking actors with the appropriate use cases.

The use case diagram in Figure 12.2 shows part of the project control system.

This is represented in **Figure 12.3** where the <<include>> stereotype is shown on a dotted line with an arrowhead pointing to the included use case. It may also emerge during more detailed investigation and specification that there are some elements of the use cases that are not mandatory for the current development iteration or release of software (in other words, the relevant requirement has not been classified as a 'must' in MoSCoW terms).

MODELLING REQUIREMENTS

Figure 12.2 System use case diagram for a project control system

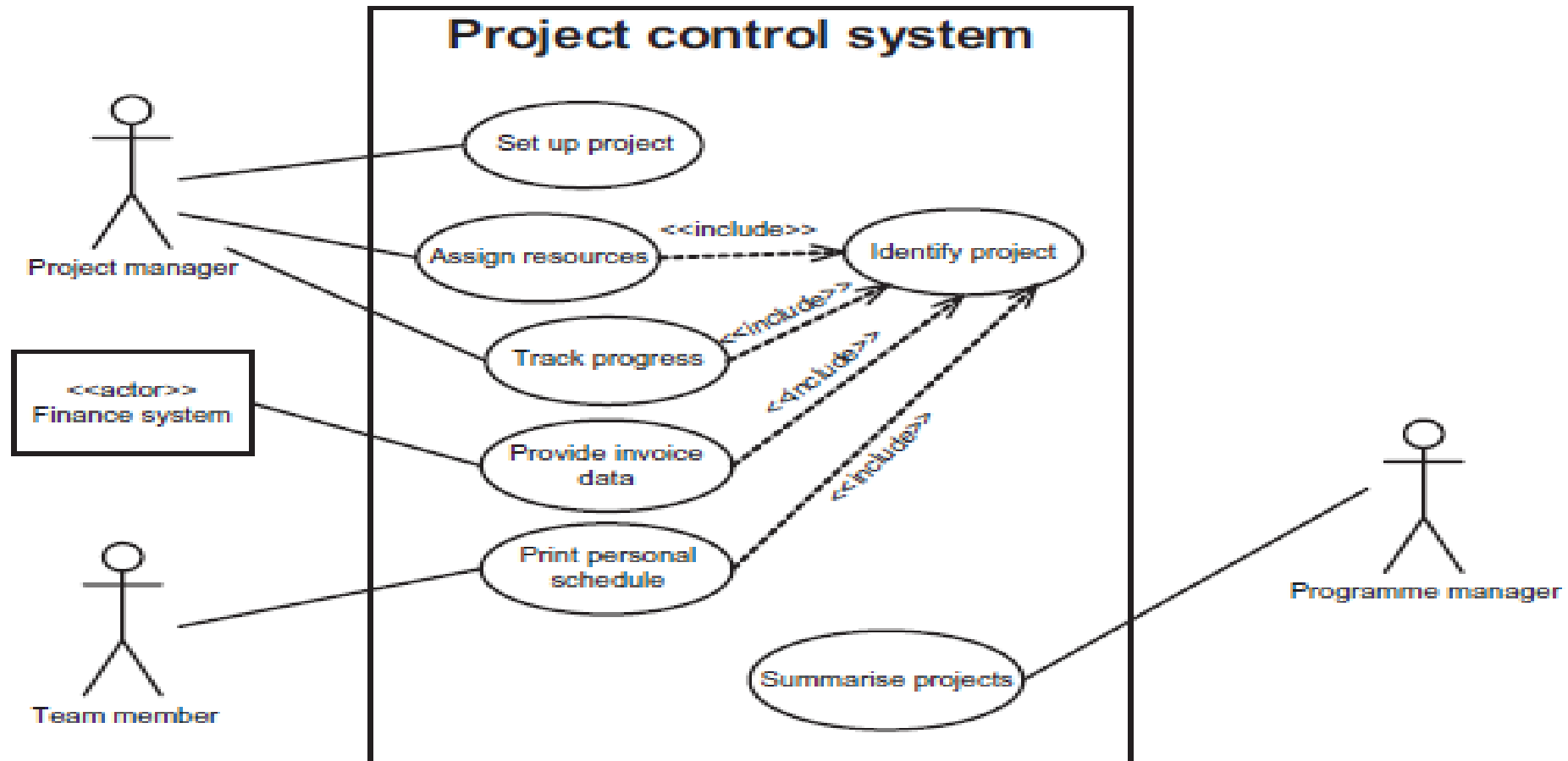


MODELLING REQUIREMENTS

❑ **The <<include>> and <<extend>> constructs- (project control system)**
This is represented in Figure 12.3 where the <<include>> stereotype is shown on a dotted line with an arrowhead pointing to the included use case. It may also emerge during more detailed investigation and specification that there are some elements of the use cases that are not mandatory for the current development iteration or release of software (in other words, the relevant requirement has not been classified as a ‘must’ in MoSCoW terms).

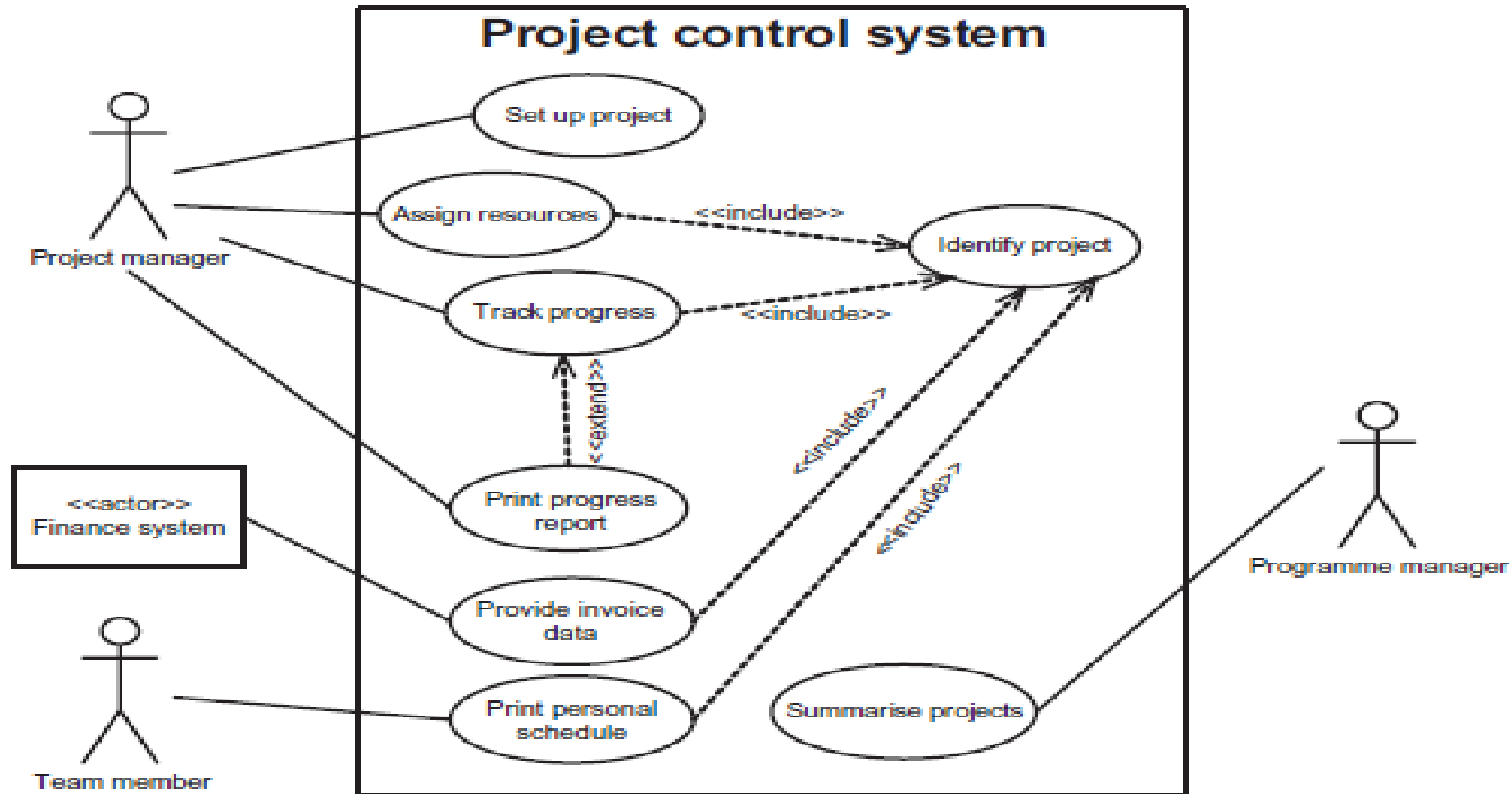
MODELLING REQUIREMENTS

Figure 12.3 Use case diagram showing <<include>>



MODELLING REQUIREMENTS

Figure 12.4 Use case diagram showing <<include>> and <<extend>>



MODELLING REQUIREMENTS

□ The `<<include>>` and `<<extend>>` constructs- (project control system)
This is represented in Figure 12.4 where the `<<include>>` and `<<extend>>` concepts allow use cases to be connected to each other. This is the only way that use cases are linked as these diagrams are not intended to show the flow – or sequence – of the processing.

MODELLING SYSTEM DATA

It is essential to model the data to be stored within the IT system.

❑ ENTITY RELATIONSHIP DIAGRAMS

Data modelling is concerned with identifying and understanding:

- the data items (attributes) which the organization (or system) needs to keep;
- the grouping of the attributes (into entities);
- the relationships between entities.

The notation used here for Entity Relationship Diagrams (ERDs) is that developed by Harry Ellis and Richard Barker and is also known as the CACI notation (after the consulting firm where they worked at the time).

MODELLING SYSTEM DATA

Existing file or database content and information needs also give pointers to system entities and their attributes.

Attributes- are not generally shown on the entity relationship diagram itself but are typically defined in supporting documentation (entity descriptions) or recorded in a data dictionary.

Relationships- A relationship is a relevant business connection between two entities. At the other end of the line, a solid line indicates that an Employee is related to exactly one Office. Similarly, in an order processing entity model there appears to be an obvious

MODELLING SYSTEM DATA

- ✓ **1:m relationship between Customer and Relationships** are often of the degree one-to-many (1:m). For example, an Employee is allocated to one Office, but each Office must have one or more Employees allocated to it.
- ✓ **One-to-one relationships** If we were looking at a system to hold data about a company, its offices and the employees working in the offices and each office was allocated to only one employee and each employee allocated to only one office, then the relationship between Office and Employee would be one-to-one.
Similarly, an order is related to one invoice and an invoice is concerned with only one order.
- ✓ **Many-to-many relationships (m:m)**- occur frequently. For example an employee may be contracted to work on one or more projects and a project may have one or more employees contracted to it.

MODELLING SYSTEM DATA

✓ **Optionality**- More detailed information about the business rules that underpin the data model is represented by including optionality in the relationship between two entities.

In this case, each Order input to the system must always have at least one Order line and each Order line must be related to exactly one Order. The complete opposite of a fully mandatory relationship occurs where a relationship is fully optional.

MODELLING SYSTEM DATA

The extended structure shows the following:

- an Employee is linked to one or more Contracts and a Contract is for exactly one Employee;
- Project is linked to one or more Contracts and a Contract is for exactly one Project. This allows all the contract details which an individual employee is associated with to be accessed as detail entity occurrences.

Note that the name of the link entity is normally the noun form of the verb which described the relationship, thus the contracted relationship is replaced by a Contract entity. If we look at an order processing example there appears to be a many-to-many relationship between the entities Product and Order.

CLASS MODELS

In a business system, a class model captures information about the particular things involved in the organization's operations, for example projects, customers and team members for a project control system.

- **Objects-** An object is something that we wish to hold data about because that data is needed within the system we are analyzing. For example, an object might be Order number UK74563 and we may wish to hold the following information about this object:

- ✓ Order number: UK74563
- ✓ Customer: C66430
- ✓ Date placed: 25 August
- ✓ Date fulfilled: 1 September
- ✓ Total value: £147.50

Objects are sent messages that invoke them to respond in some way, typically by changing data. For example, a message might be sent to the object UK74563 to change the date the order was fulfilled.

CLASS MODELS

- **Classes-** To build a model of the system data we consider classes of objects rather than individual objects.

In object class modelling, we have classes that provide the generic definition of the data items or attributes, and objects that are the instances of a particular class.

If the name has more than one word then each word is joined and capitalized in the class name, for example, Order Line. The attributes – the individual items of data about the class – are stored in the middle section. The attribute names are usually shown in lower case with constituent parts shown with a capital letter.

CLASS MODELS

Figure 12.17 Definition of the class 'Order'

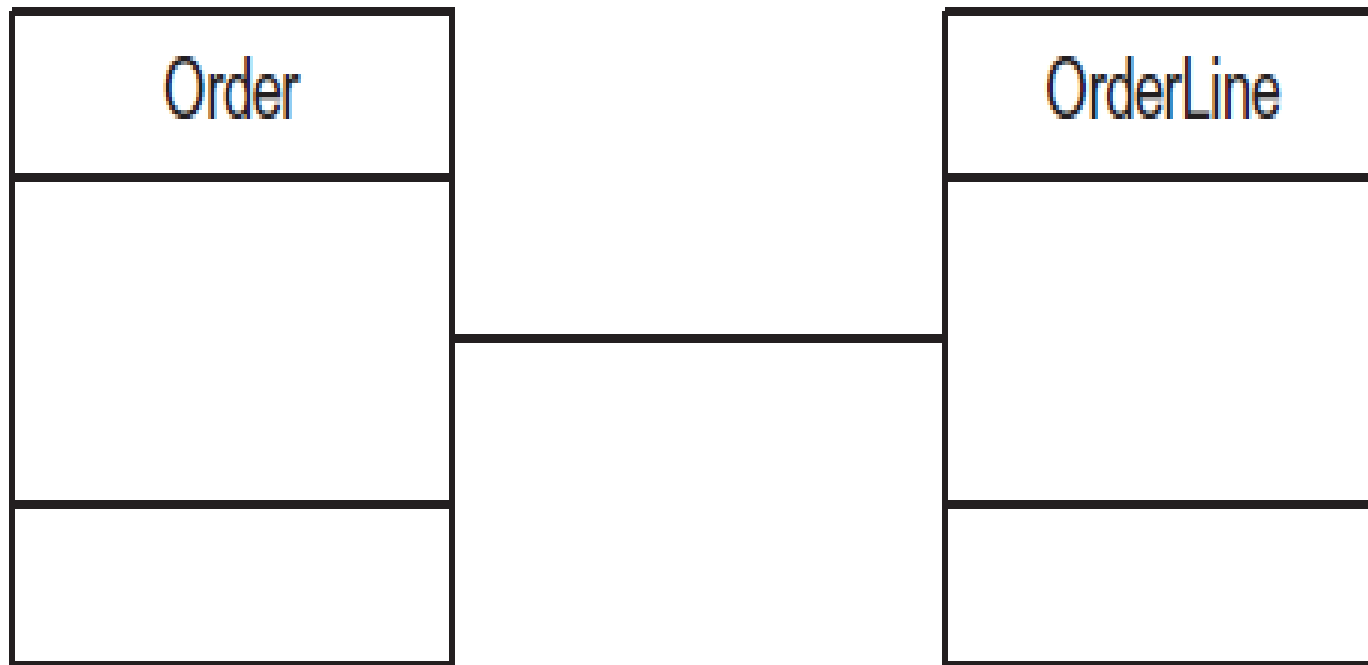
Order
orderNumber customer datePlaced dateFulfilled totalValue
createOrder updateDateFulfilled

CLASS MODELS

- **Associations-** As in entity relationship modelling, we now need to establish how different classes are linked to each other and the nature of these connections. We call the connections between classes '*associations*'. For instance, an Order class must have an association with an Order Line class so that the system will be able to identify the items on the order. **Figure 12.18 shows this association.**

CLASS MODELS

Figure 12.18 Association between two classes



CLASS MODELS

- **Generalization and inheritance**

Sometimes, as we get further into the analysis, we discover that there are different sub-categories for a class. For instance, a general and a private are both soldiers but while some of the information that we would like to hold about them is the same, other information may differ greatly. UML handles this situation through a concept known as generalization.

MODELLING IN AGILE APPROACHES

It is sometimes thought that models are not used in projects that follow the Agile approach because, in these projects, there is not the need to fully define the system 'up front'. Indeed, the Agile Manifesto stated that working software is valued over comprehensive documentation.

- However, the Manifesto does not say that there is *no* value in documentation, only that working software is *more* valuable. It is our view that, even in an Agile project, models can be useful to help the user community understand the scope and scale of the proposed system before detailed development commences.

THE USE OF MODELS IN SYSTEM MAINTENANCE

It is a fact that, if the lifetime cost of an IT system is considered, by far the bulk of the expenditure – 70 per cent according to some authorities – is incurred after the system has been implemented.

A common problem that faces support and maintenance teams – which often consist of different people and sometimes even different firms than originally built the system – is understanding the overall scope of a system and how it all fits together. In addition, although inspection of the code can usually reveal *how* a system works, the business reasons for it being built that way are often lost in the mists of time.

CONCLUSION

Well defined and traceable requirements are undeniably key to a successful business change project. They set out what a solution needs to deliver and the relative priorities of the features to be included. The requirements document needs to be structured clearly so that a thorough requirements validation can be conducted by the reviewers. Traceability of the requirements is also critical if the business is to be able to confirm that their needs have been met and all decisions regarding the requirements are to be transparent and auditable.

Moreover, the benefit of using models is that they provide an unambiguous view of the system, albeit from one specific perspective. These views may be compared and cross-checked. So we might develop a use case diagram and cross-check it against a model of the data in order to identify any gaps such as missing data items or use cases. The models also help to generate further questions and improve our understanding of the business requirements and the business rules to be implemented in the system.

REFERENCES

- **Required Reference:**

Debra Paul and James Cadle, et al, (2020). “Business Analysis”. BCS, 4th Edition. ISBN:978-1-78017-5102

- **Recommended Reading**

James Cadle, Debra Paul, Jonathan Hunsley, Adrian Reed, David Beckham, Paul Turner (2021). “Business Analysis Techniques” 3rd Edition, BCS. ISBN:978-1-78017-5690



مجموعة جامعة
الجميع الذكية
للتعليم القابضة

MIDOCEAN
UNIVERSITY
BY EVERYONE'S SMART UNIVERSITY



The End of Lecture

Thank You

