

# Office Solutions Development

Topic 4:

An Introduction to End-User Software Development

# Scope and Coverage

- ▶ This topic will cover:
  - ▶ The need to address both user and business requirements, including a discussion of the end-user's role in developing application software and important features of interface design.

# Learning Outcomes - 1

- ▶ By the end of this topic, students will be able to:
  - ▶ Discuss the need for application software to meet user and business requirements
  - ▶ Broadly define 'end-user' and 'end-user development'
  - ▶ List the range of software tools available for end-users to use when developing application software solutions.

# Learning Outcomes - 2

- ▶ Identify the benefits and disadvantages of end-user development
- ▶ Discuss the need for analysis, design, testing and documentation
- ▶ Discuss the importance of efficient software interface design
- ▶ Identify end-user development guidelines
- ▶ Identify and discuss interface development in Microsoft Office

# Terminology

- ▶ Terminology will be explained in the lecture, tutorial and laboratory session. You should take notes!
- ▶ A glossary of terms will be provided.
- ▶ Ask questions if you do not understand.

# The Need for Application Software to Meet User and Business Requirements - 1

Application software should be:

- ▶ Efficient in its *functionality*, for example, helping to solve problems, making decisions and improving system functions, such as automating regular processes, e.g. data entry
- ▶ *User-friendly*, e.g. easy to operate

# The Need for Application Software to Meet User and Business Requirements - 2

Application software should be (cont.):

- ▶ **Reliable**, e.g. producing the correct results of database query
- ▶ **Secure**, e.g. locking spreadsheet cells
- ▶ **Maintainable**, e.g. modification of a spreadsheet

# End-User Application Software - 1

- ▶ In Topic 1, we discussed the two main types of software - *commercial* and *bespoke*.
- ▶ Application software can also be created and/or modified by an organisation's own non-programming/technical staff. This development method is described as: *end-user development* (EUD).



# End-User Application Software - 2

- ▶ End-user application software is developed by end-users (EUs) of computer systems to be used by them in their day-to-day work and is often also used by their colleagues in the same department and/or in other departments.
- ▶ Applications developed by end-users are also described as *User Developed Applications* (UDAs).

# Software Tools Available for Developing Application Software Solutions

- ▶ Standard application software advanced functions in databases, spreadsheet and word processing
  - ▶ Macros
  - ▶ Visual Basic for Applications (VBA)
- ▶ Macros and VBA refer to application software programming methods and we will study how to use them in Topics 5-9.

# Which Organisations Use EUD?

- ▶ Mainly, but not exclusively, *small to medium enterprises (SMEs)* for small scale development, e.g.
  - ▶ Automating processes such as data entry and data output
    - ▶ Information management and retrieval
    - ▶ Facilitating data transfer between software applications, such as databases and spreadsheets
- ▶ Development can result from internal business processes, information systems and external business needs.

# Business Processes that End-Users Develop

**Accounting  
and Finance**

**Web  
Applications**

**EUD**

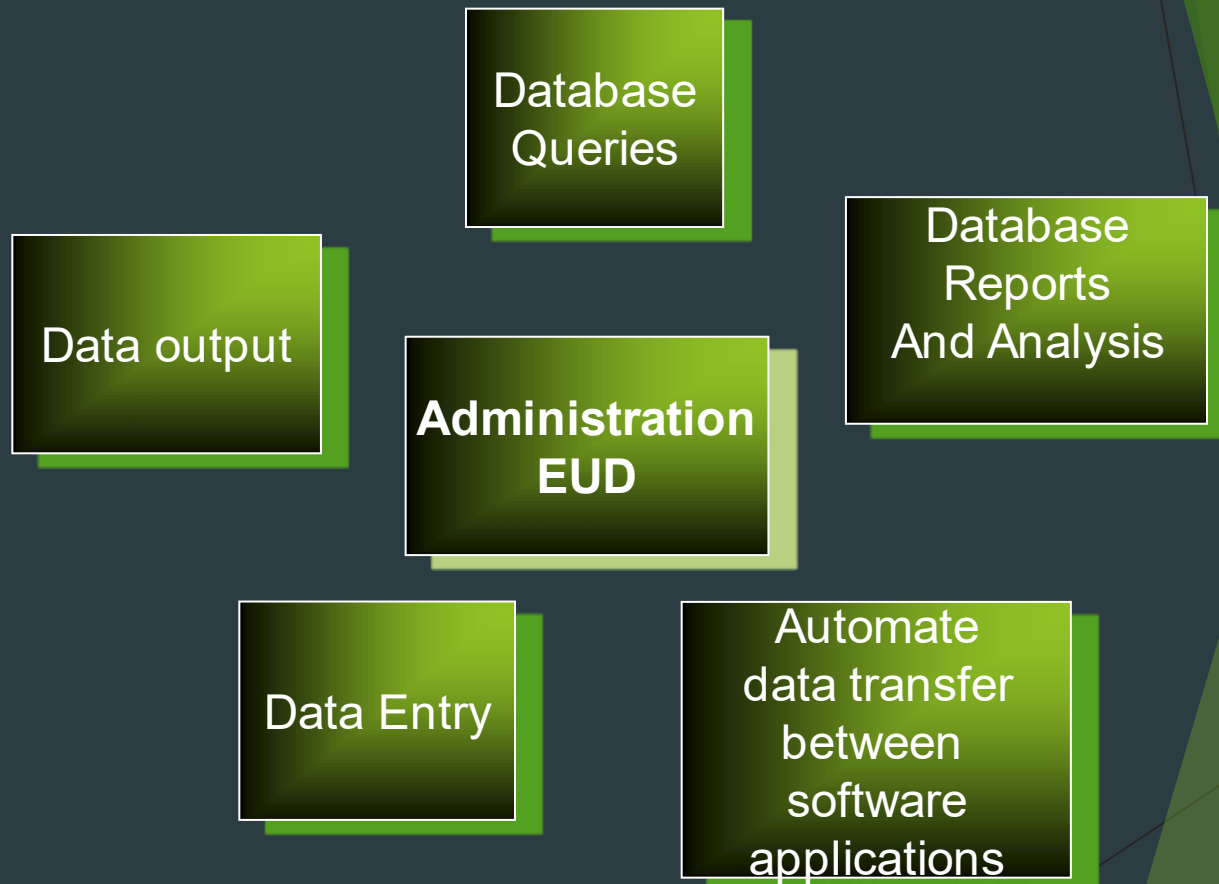
**Administration**

**Marketing**

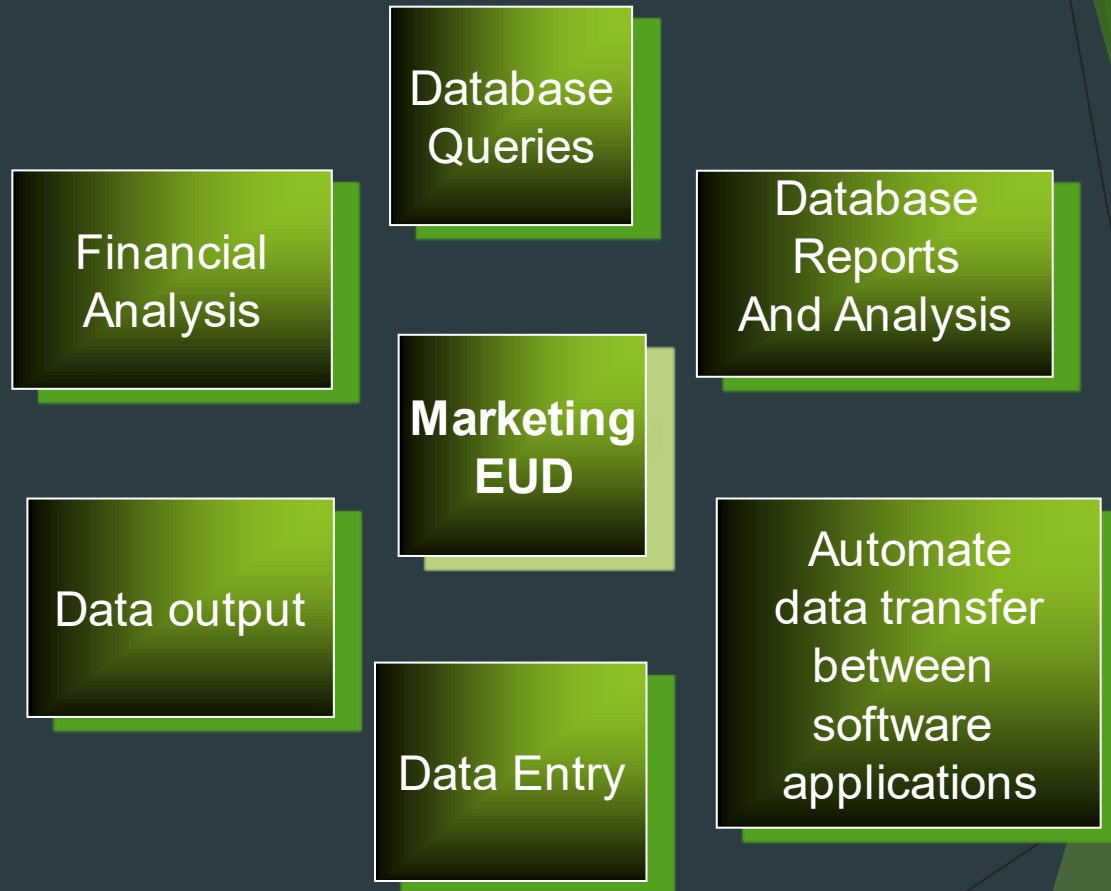
# Accounting and Finance EUD



# Administration EUD



# Marketing EUD



# Benefits of EUD

- ▶ Greater user involvement
- ▶ User has knowledge of the functionality of the system
- ▶ Good use of resources
- ▶ Less costly method
- ▶ Improved decision making
- ▶ More user satisfaction



# EUD Disadvantages

- ▶ Inaccurate (EU lacks training)
- ▶ Lack of planning
- ▶ Can be slow
- ▶ Use of incorrect formulae
- ▶ Not secure
- ▶ Not tested
- ▶ Not documented

# Analysis, Design, Testing and Documenting Requirements - 1

- ▶ The user should concentrate on *what* needs to be done as well as *how* it should be done and *plan* carefully their software development:
  - ▶ The task/problem should be *analysed* in detail and the user should understand clearly what needs to be done.
  - ▶ The user and business requirements must be identified clearly.

# Analysis, Design, Testing and Documenting Requirements -

2

- ▶ A solution must be *designed* that meets all requirements.
- ▶ The solution and results should be *tested* thoroughly.
- ▶ The solution should be *documented*.

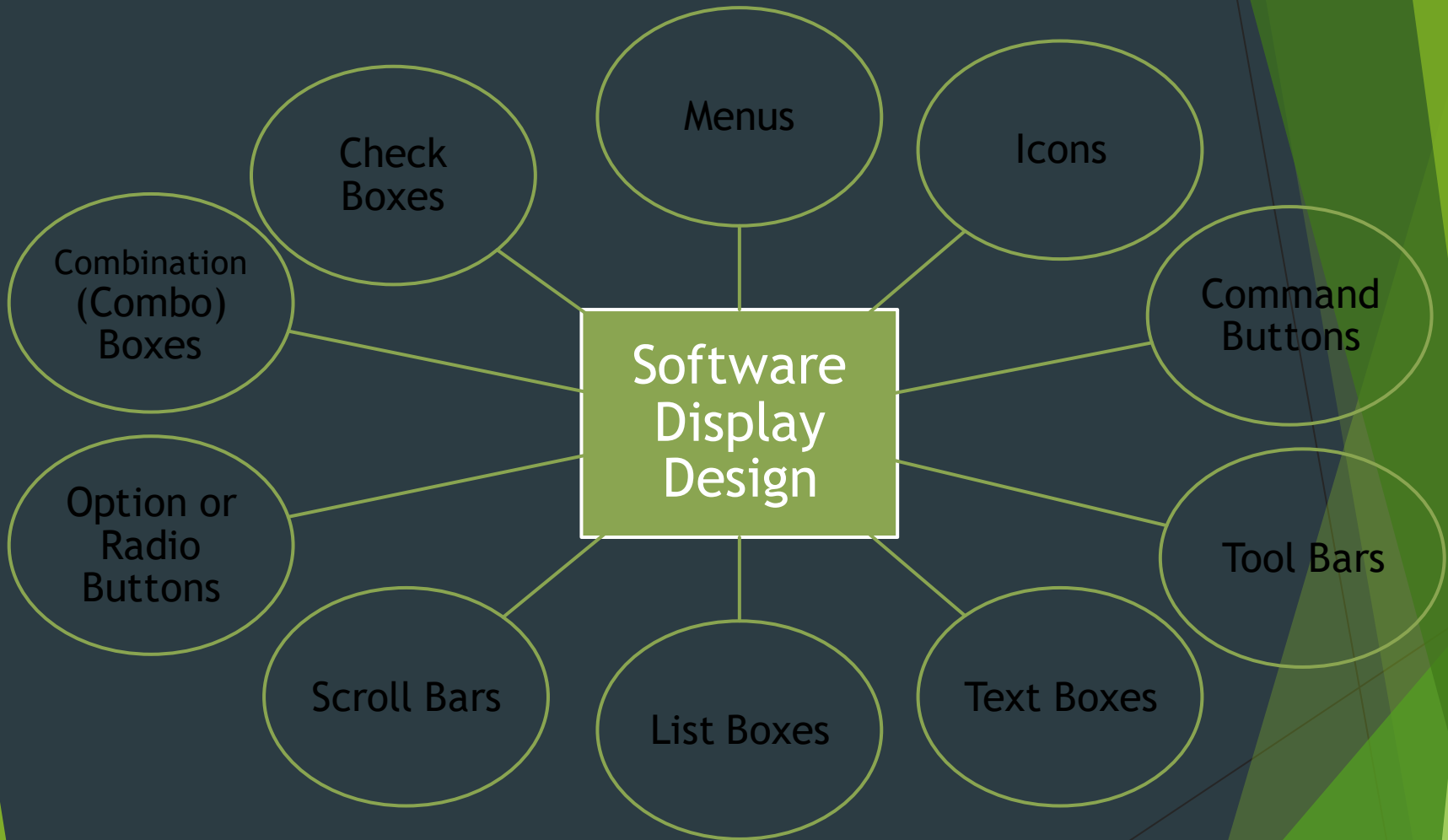
# Human Computer Interface Design

- ▶ *Human Computer Interaction (HCI)* refers to the interaction between users and the computer system that is enabled by using the hardware and software.
- ▶ Efficient HCI is necessary for usability of the system, e.g. the user should be able to interact with the system as simply and efficiently as possible.
- ▶ There are many aspects of HCI design, but we will concentrate on software interface design when developing our office solutions.

# Software Interface Design

- ▶ **Text** (font type, font size, letter and line spacing, line length, justification, line endings, paragraph spacing)
- ▶ **Colour** (how it is used to draw attention, show meaning, etc.)
- ▶ **Images** (still and moving, how it is used to communicate, persuade, etc.)
- ▶ **Sound** (sound effects, music speech, how it is used to attract attention)

# Software Display Design



# EUD Guidelines

- ▶ EUs should be aware of and refer to the following guidelines when developing application software:

- ▶ **ISO 9126 Software Quality Characteristics**  
[Available Online]

- <http://www.sqa.net/iso9126.html>

# Topic 4 - An Introduction to End-User Software Development

Any Questions?