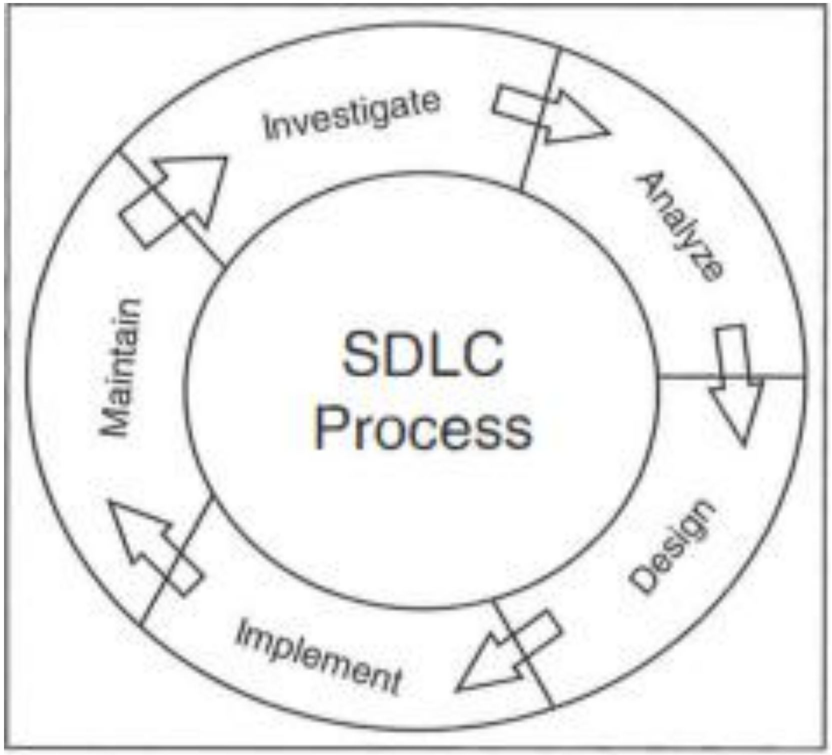
Chapter – 3

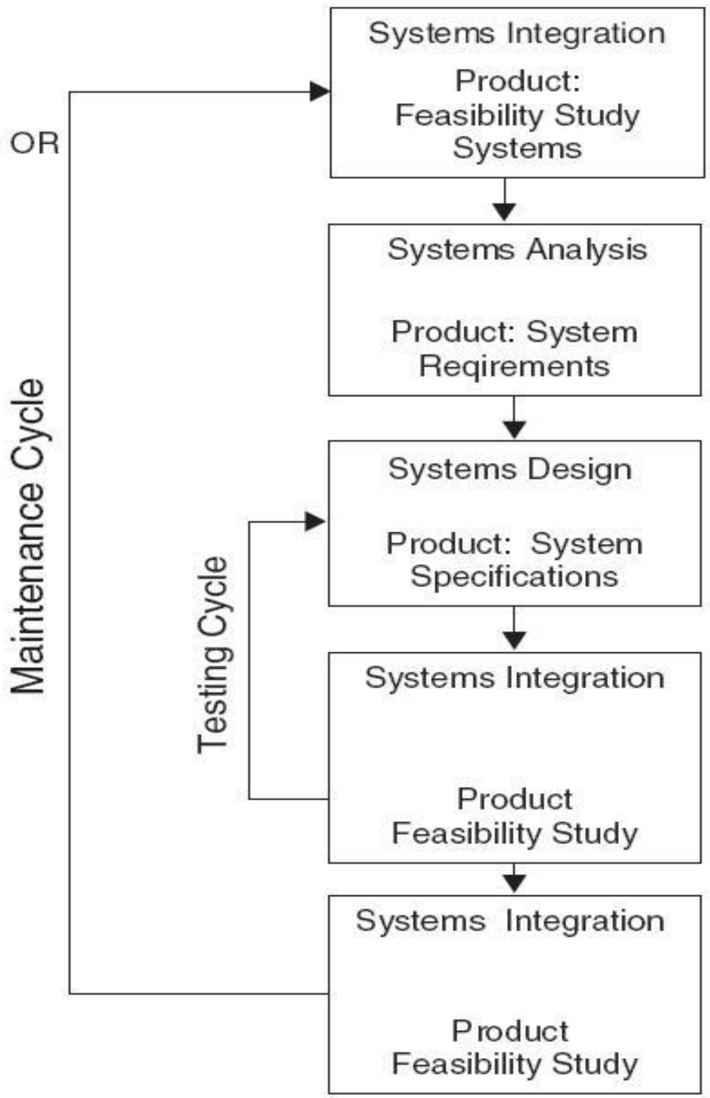
Development of Enterprise Systems

Widely accepted five-phase SDLC methodology:

1. Investigation
2. Analysis
3. Design
4. Implementation
5. Maintenance



SDLC Approach



UNDERSTAND THE BUSINESS PROBLEM OR OPPORTUNITY

DEVELOP AN INFORMATION SYSTEM (IS) SOLUTION

IMPLEMENT IS SOLUTION

Maintenance Cycle

Testing Cycle

Systems Integration

Product: Feasibility Study Systems

Systems Analysis

Product: System Reqirements

Systems Design

Product: System Specifications

Systems Integration

Product Feasibility Study

Systems Integration

Product Feasibility Study

- ■ Determine whether a business problem or opportunity exists
- ■ Conduct a feasibility study to determine whether a new or improved information system is needed
- ■ Analyze in detail the information needs of end users, the organizational environment, and any system presently used
- ■ Develop the logical input, processing, output, storage, and control
- ■ Develop specifications for the hardware (machines and media), software (programs and procedures), People (specialists and end users), data resources, and information products that will satisfy the information needs of end users
- ■ Acquire (or develop) and install hardware and software
- ■ Test and document the system
- ■ Train people to operte and use the system
- ■ Convert to the new system
- ■ Use a postimplementation review process to monitor, evaluate, and modify the system as needed

Chapter – 3

Development of Enterprise Systems

- This includes a systematic process of planning, designing, and creating an information system for organizations.

- It is often better to have a structured methodology to avoid mishaps and coordinate the design and development tasks properly among the members of a large systems development team.

- **Systems Approach**—Complex problems are broken up into smaller manageable problems using a systems' hierarchy, and then developing a solution for each problem within the hierarchy

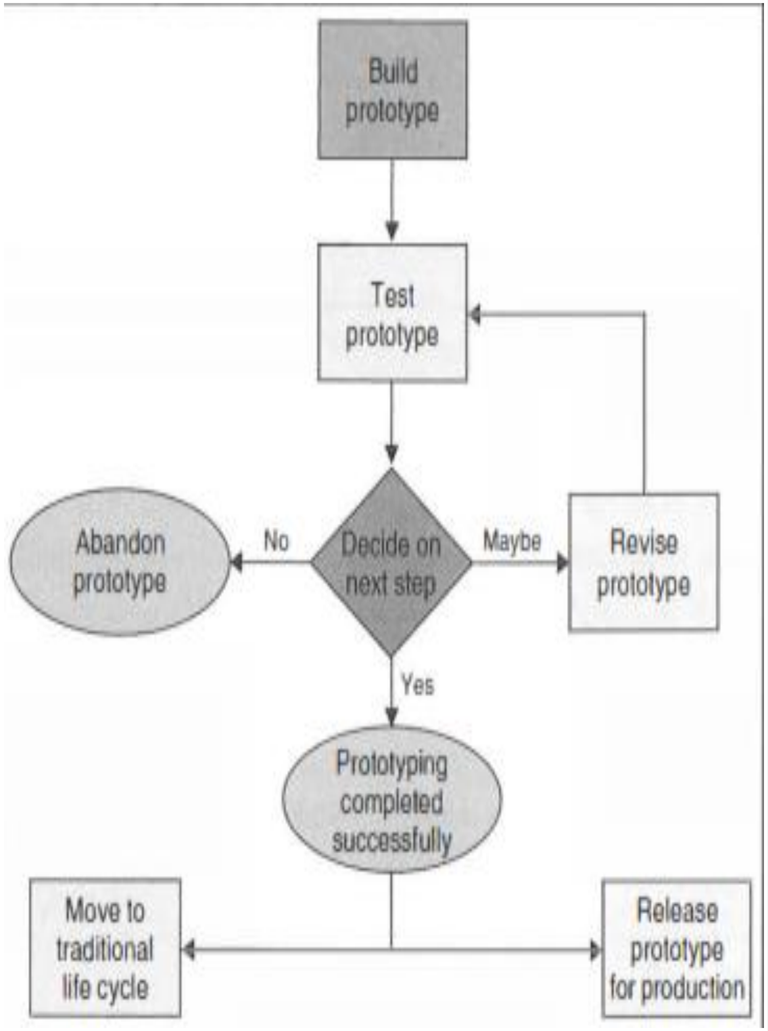
Rapid Development Approaches for ES Development

- **Prototyping**
  - This approach **does not go through the analysis and design phase.**
  - It implements a skeleton or a prototype of the actual system with a focus on data input and output.
  - The idea is to demonstrate the system functionality to the users.
  - **Feedback is incorporated into the new system and demonstrated back to the users.**
  - This approach has proven to be very effective with user interactive systems because the prototype is eventually converted into a full-scale system

Development of Enterprise Systems

Rapid SDLC Approaches: Prototype

- This approach does not go through the analysis and design phases;
- instead, **it implements a skeleton or a prototype of the actual system** with a focus on input (i.e. , user interface) and output (i.e., screen displays and reports generated with dummy data).
- This approach has proven to be very effective with user interactive systems because the prototype is eventually converted into a full-scale system


Build prototype
Test prototype
Abandon prototype
No
Decide on next step
Maybe
Revise prototype
Yes
Prototyping completed successfully
Move to traditional life cycle
Release prototype for production

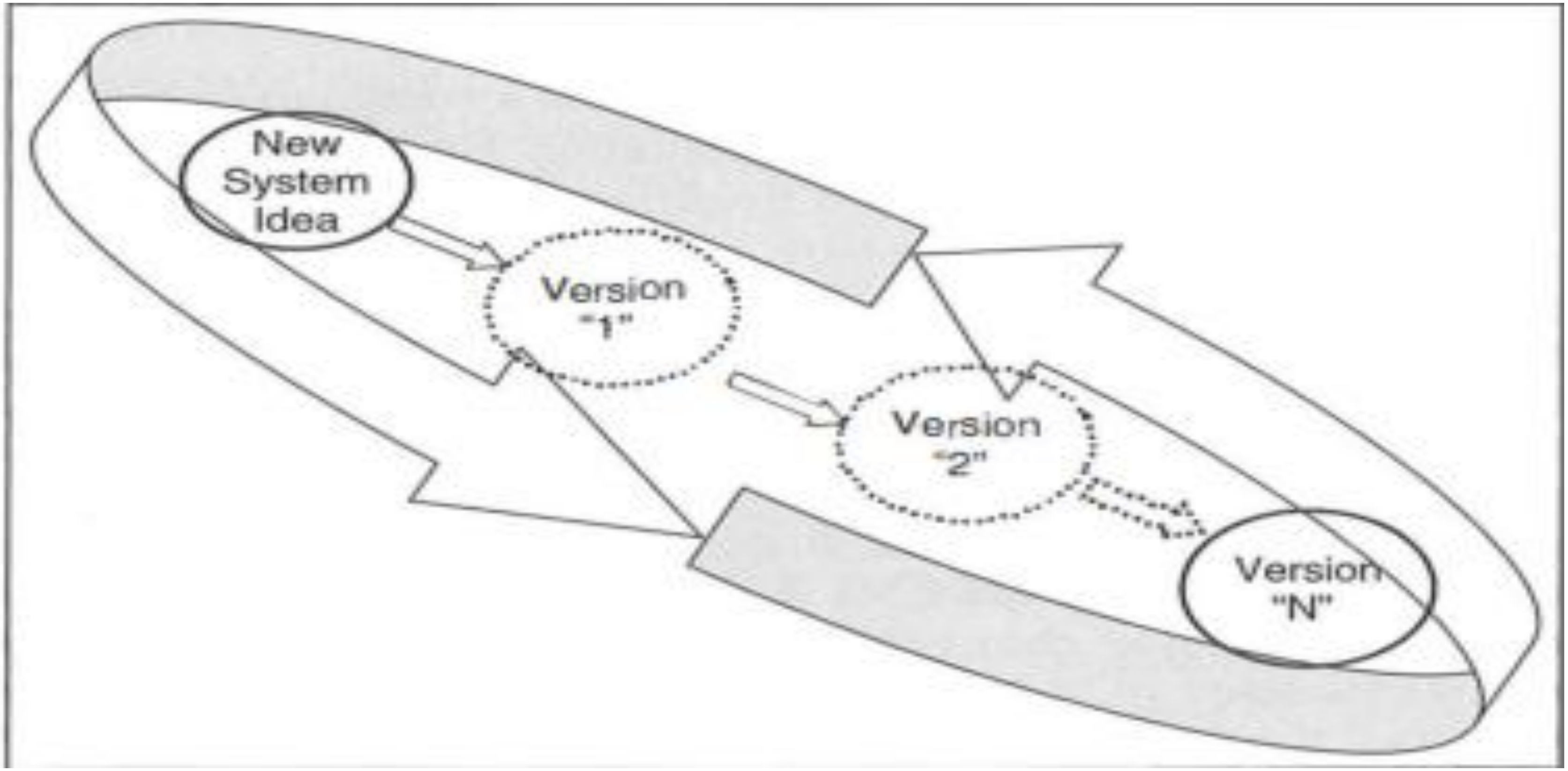
Rapid Development Approaches for ES Development

- **End User Development (EUD)**
  - End-users create information systems using spreadsheets and databases
  - Not effective for large-scale development
  - Users are trained to develop their own applications (e.g., a departmental employee tracking system with an Access database

Chapter – 3

Development of Enterprise Systems

Rapid Application Development Process



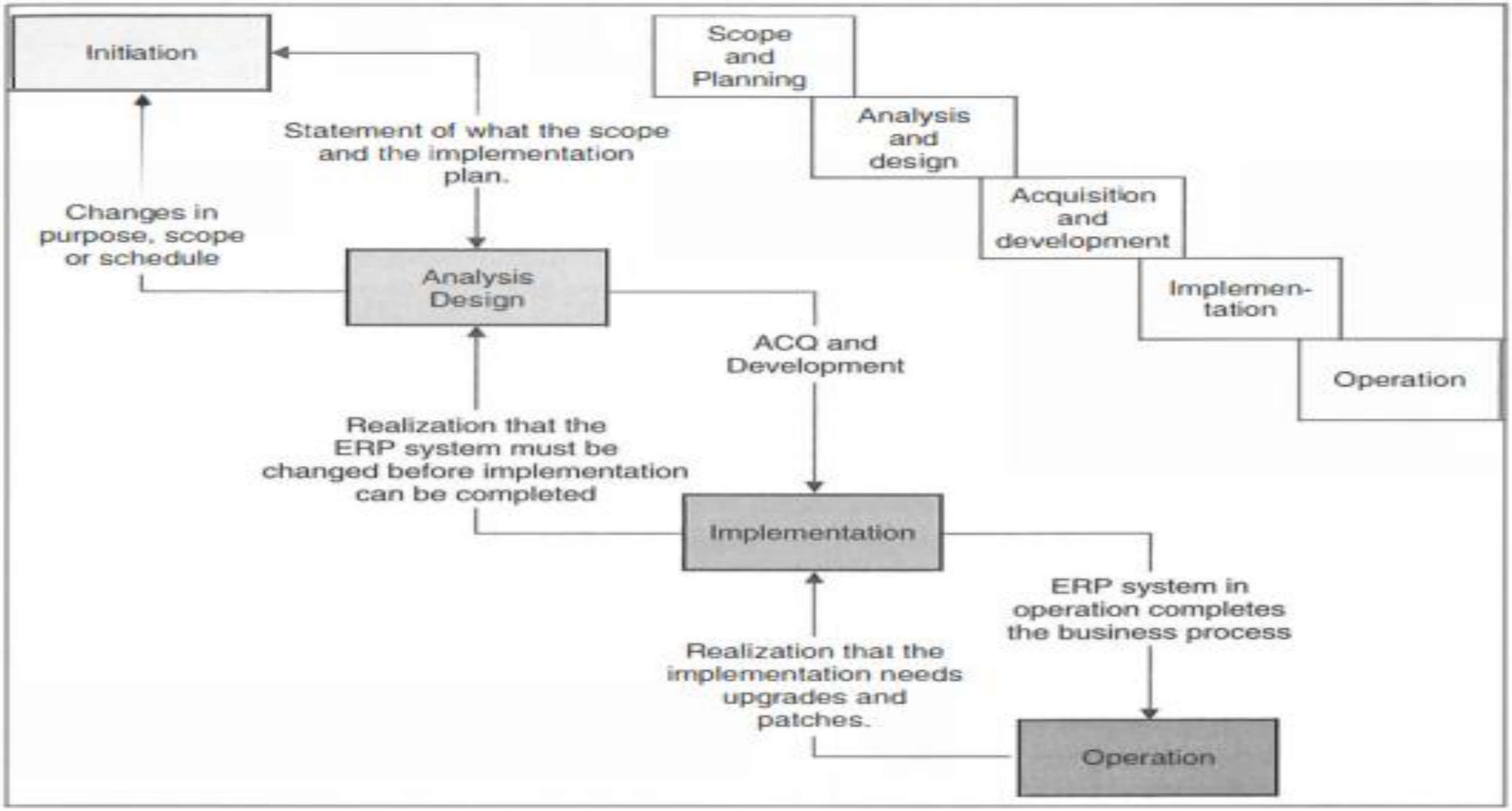
Traditional ERP Life Cycle

- The traditional ERP life cycle approach has a deliverable at the end of each stage (e.g., a report with supporting documents) that is reviewed by management and upon which a decision is made either to continue with the project or not.

- It has FOUR Phases
  1. Scope and Commitment Stage [Initiation]
  2. Analysis & Design
  3. Implementation
  4. Operation

Development of Enterprise Systems

Traditional ERP Life Cycle

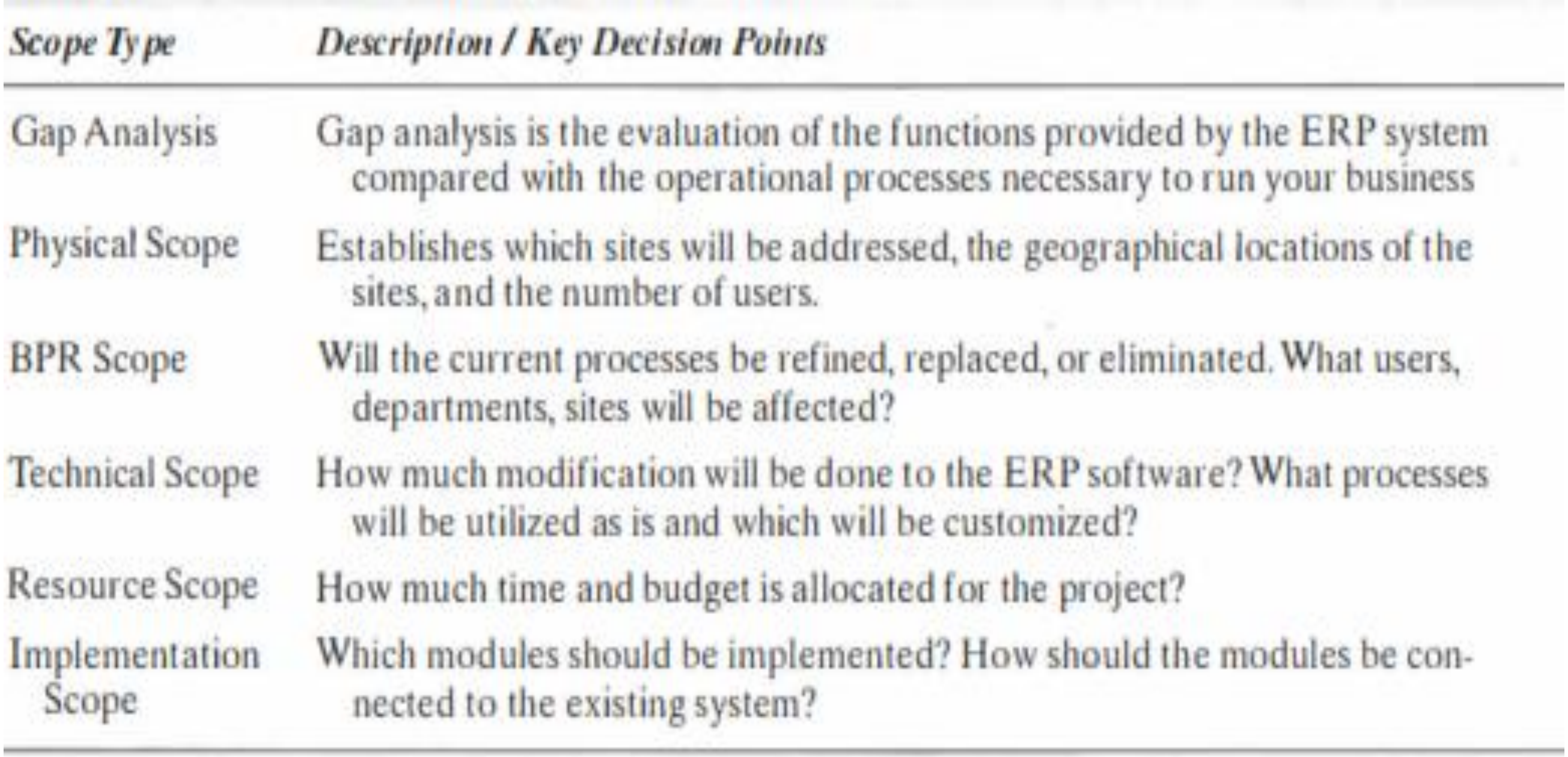

Initiation
Scope and Planning
Statement of what the scope and the implementation plan.
Analysis and design
Changes in purpose, scope or schedule
Acquisition and development
Analysis Design
Implementation
ACQ and Development
Operation
Realization that the ERP system must be changed before implementation can be completed
Implementation
ERP system in operation completes the business process
Realization that the implementation needs upgrades and patches.
Operation

Chapter – 3

Development of Enterprise Systems

Traditional ERP Life Cycle

1. Scope and Commitment Stage [Initiation]



Scope Type	*Description / Key Decision Points*
Gap Analysis	Gap analysis is the evaluation of the functions provided by the ERP system compared with the operational processes necessary to run your business
Physical Scope	Establishes which sites will be addressed, the geographical locations of the sites, and the number of users.
BPR Scope	Will the current processes be refined, replaced, or eliminated. What users, departments, sites will be affected?
Technical Scope	How much modification will be done to the ERP software? What processes will be utilized as is and which will be customized?
Resource Scope	How much time and budget is allocated for the project?
Implementation Scope	Which modules should be implemented? How should the modules be connected to the existing system?

Chapter – 3

Development of Enterprise Systems

Traditional ERP Life Cycle

2. **Analysis & Design**

- Analysis of user requirements, the ERP team has first to make a decision on the software, and decide on consultants and SMEs.
- Another key activity is to map the differences between the current business process and the embedded process in the ERP software or gap analysis, and
- to develop a long-term plan on whether to change the business processes of the organization, or to customize the ERP software to support existing processes

Development of Enterprise Systems

Traditional ERP Life Cycle

3. Acquisition and Development stage

- ❑ The organization has to purchase the license for the production version of the software and build the production version of the system, which is eventually to be made available to end users.
- ❑ The entire production platform must be configured and built with the necessary hardware, network, security, software, database, and real production data processes
- ❑ Data mapping, missing data, and data dictionary design are major tasks. Finally, the ERP system needs to be configured with proper security, implement the authentication and authorization policy for accessing the system, and contain other modifications as recommended by the design plan.

Development of Enterprise Systems

Traditional ERP Life Cycle

4. Implementation Stage

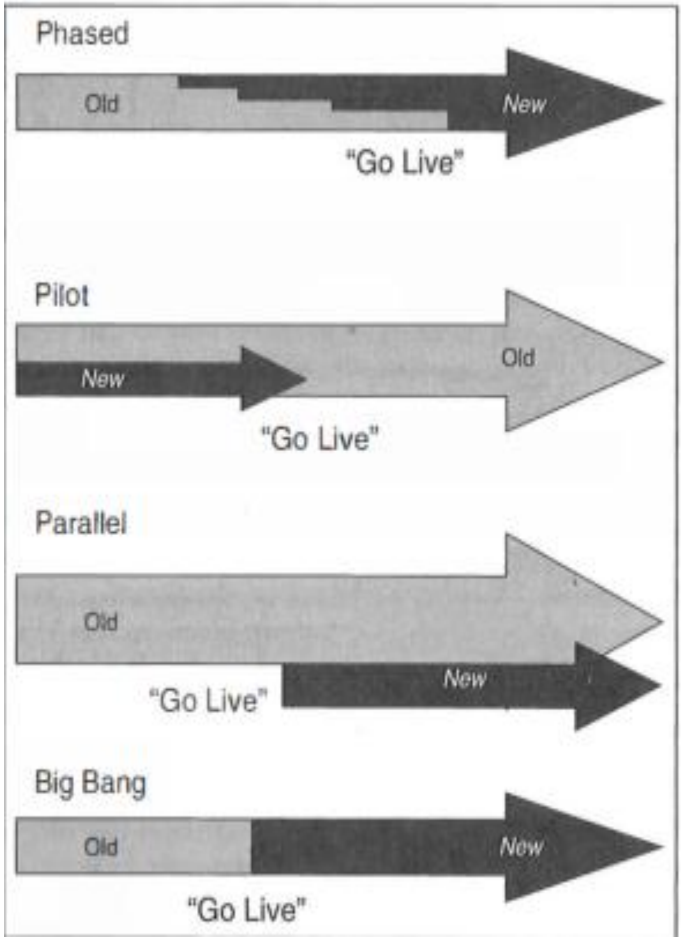
- The focus is on installing and releasing the system to the end-users (i.e., "Go-Live") and on monitoring the system release to
- the end-users.
- This platform is a mirror of the development version of the sys
- tem, but no changes can be made to the production platform without shutting down the system and following the steps in the conversion plan.
- Errors found in the production version have to go through the help desk or support staff

Development of Enterprise Systems

Traditional ERP Life Cycle 5. ERP Conversion approaches

Operation Stage:

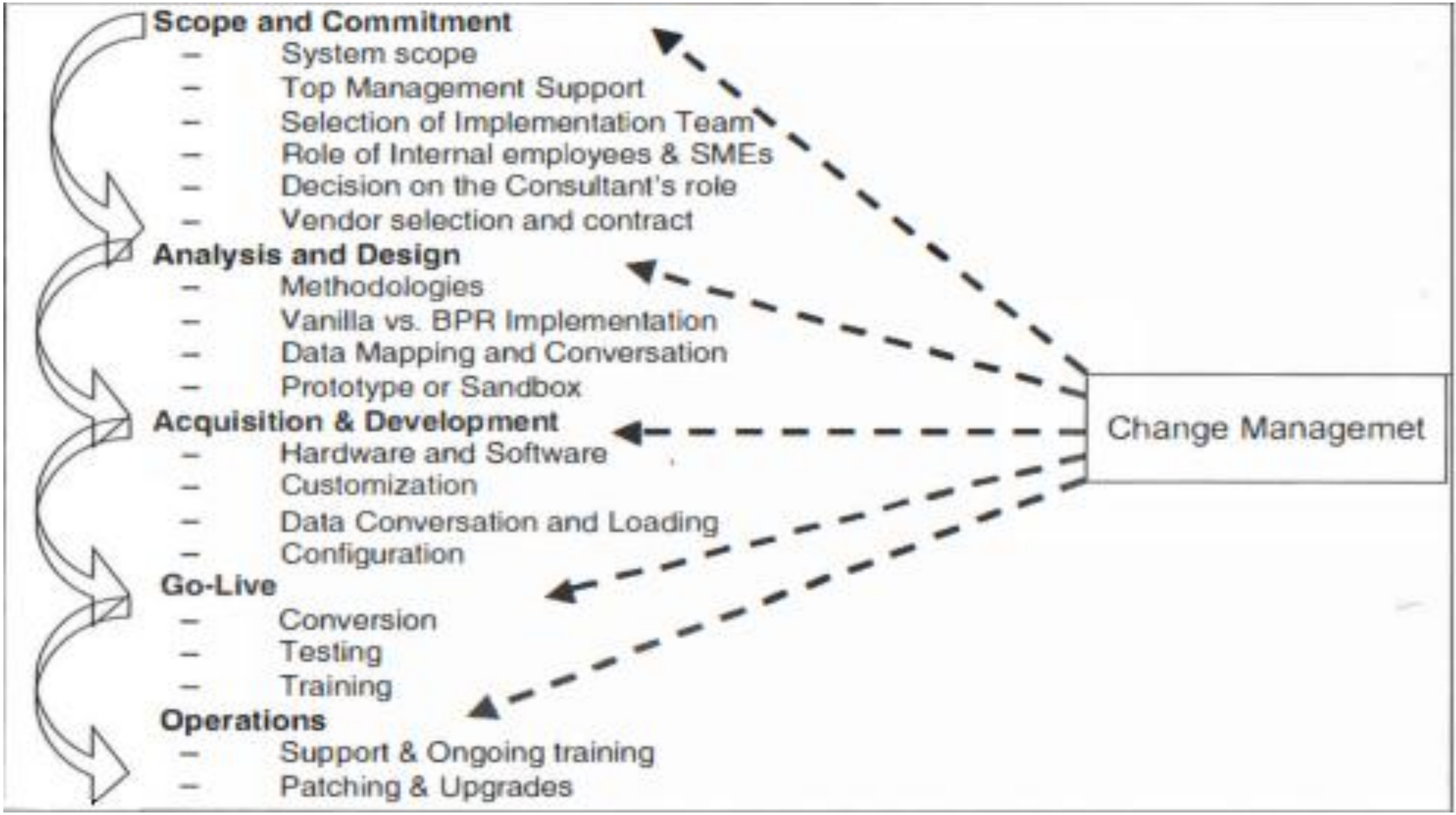
- ❑ This is often managed by the operation team with assistance from the implementation team.
- ❑ Handover or knowledge transfer is the major activity as support for the new system is migrated to the help desk and support staff
- ❑ Another key activity is management of new releases of the software, installation of patches and upgrades to the system, and managing the software contract with the ERP vendor.



Chapter – 3

Development of Enterprise Systems

ERP Life Cycle Summary



Scope and Commitment
- System scope
- Top Management Support
- Selection of Implementation Team
- Role of Internal employees & SMEs
- Decision on the Consultant's role
- Vendor selection and contract

Analysis and Design
- Methodologies
- Vanilla vs. BPR Implementation
- Data Mapping and Conversation
- Prototype or Sandbox

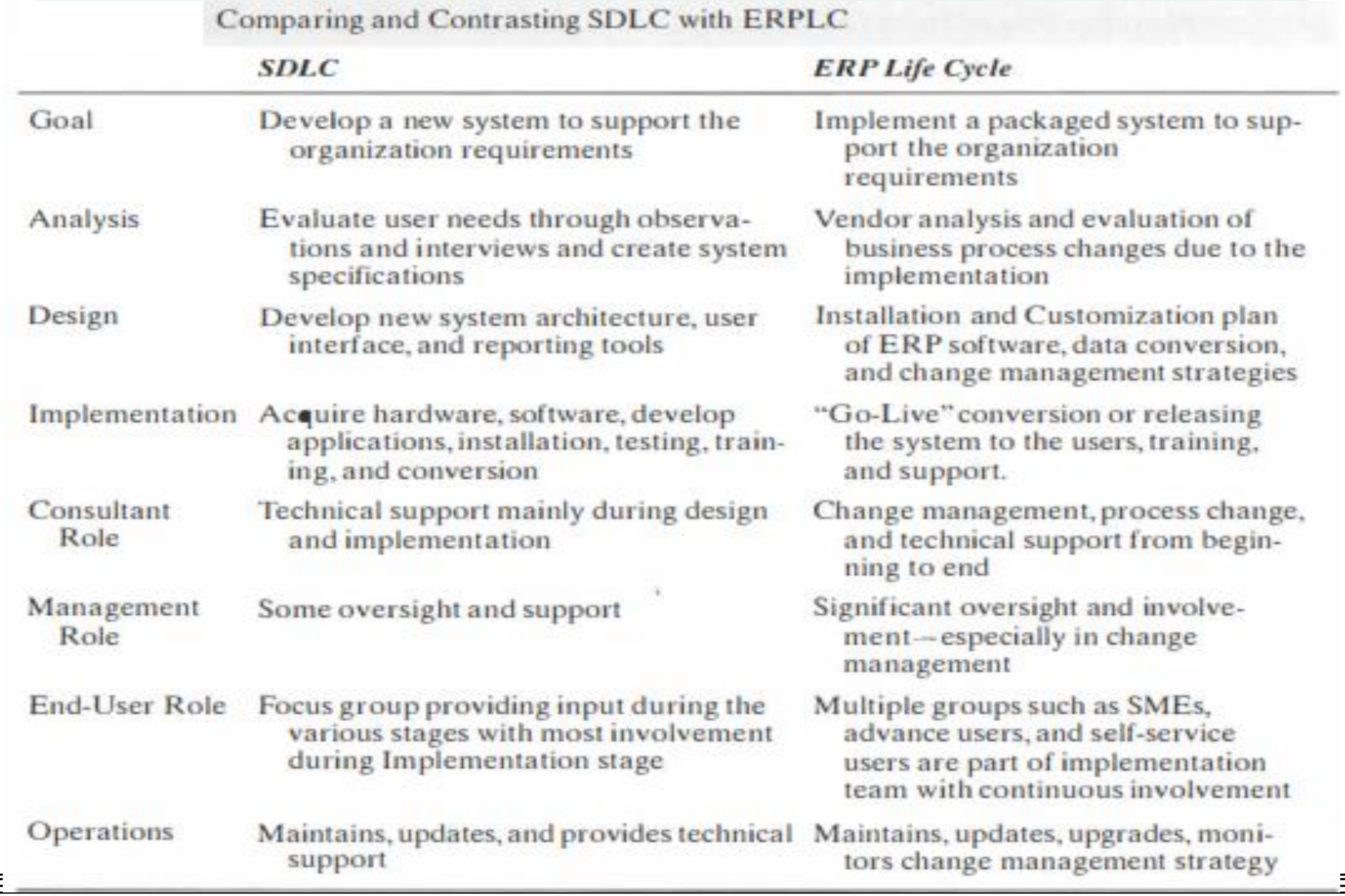
Acquisition & Development
- Hardware and Software
- Customization
- Data Conversation and Loading
- Configuration

Go-Live
- Conversion
- Testing
- Training

Operations
- Support & Ongoing training
- Patching & Upgrades

Chapter – 3

Development of Enterprise Systems



Comparing and Contrasting SDLC with ERPLC

	SDLC	*ERP Life Cycle*
Goal	Develop a new system to support the organization requirements	Implement a packaged system to support the organization requirements
Analysis	Evaluate user needs through observations and interviews and create system specifications	Vendor analysis and evaluation of business process changes due to the implementation
Design	Develop new system architecture, user interface, and reporting tools	Installation and Customization plan of ERP software, data conversion, and change management strategies
Implementation	Acquire hardware, software, develop applications, installation, testing, training, and conversion	"Go-Live" conversion or releasing the system to the users, training, and support.
Consultant Role	Technical support mainly during design and implementation	Change management, process change, and technical support from beginning to end
Management Role	Some oversight and support	Significant oversight and involvement—especially in change management
End-User Role	Focus group providing input during the various stages with most involvement during Implementation stage	Multiple groups such as SMEs, advance users, and self-service users are part of implementation team with continuous involvement
Operations	Maintains, updates, and provides technical support	Maintains, updates, upgrades, monitors change management strategy

Product Software Development

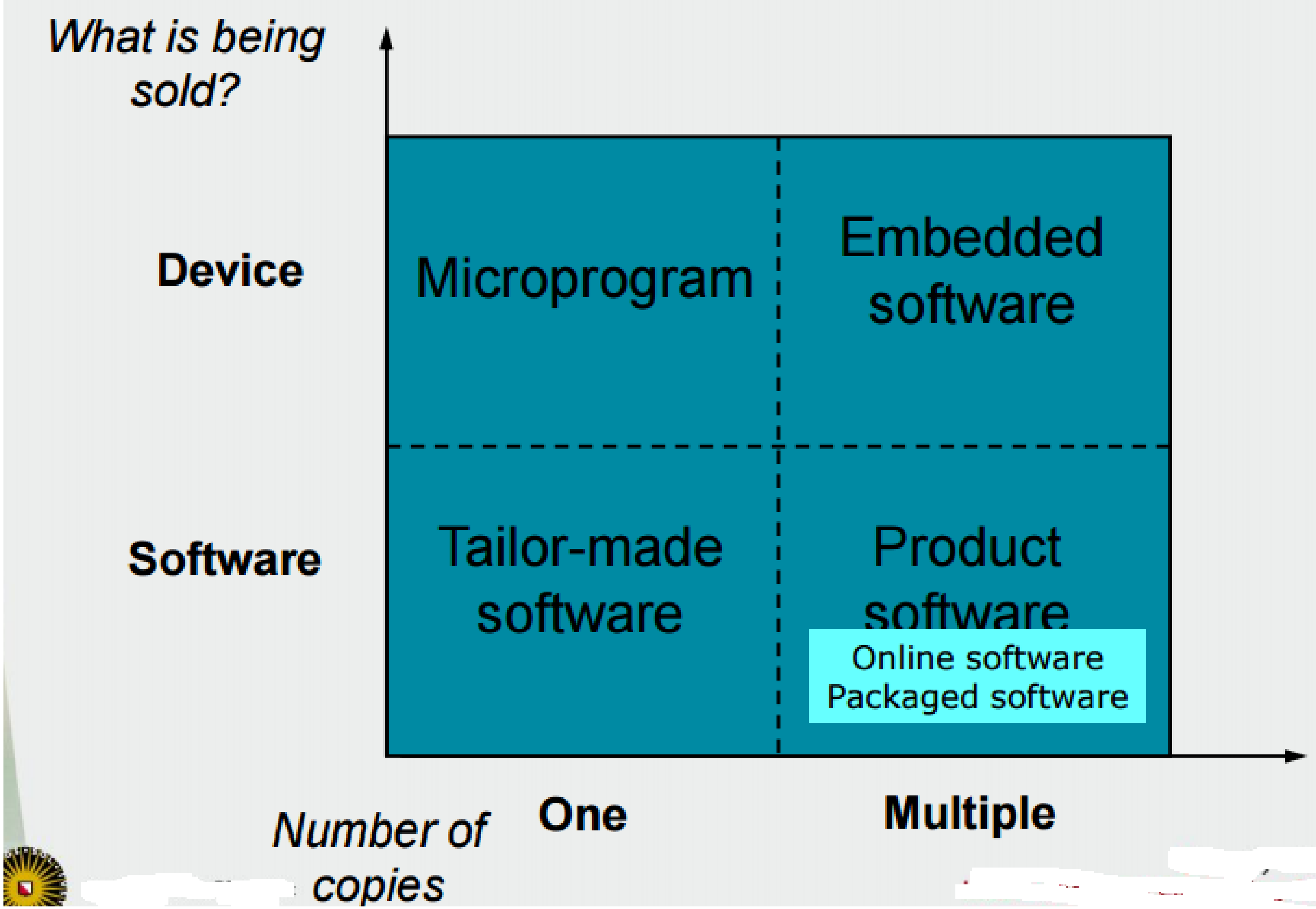
A Software Product is defined as

a packaged configuration of software components, or a software-based service with auxiliary materials, which is released for and traded in a specific market (Xu & Brinkkemper, EJIS 2007)

Examples:

- ERP software
- Bookkeeping service
- Operating systems
- Desk-top publishing
- Computer-aided design
- Software development environments
- Customer-relationship mgmt

What is Product Software …?



What is being sold?

	One	Multiple
Device	Microprogram	Embedded software
Software	Tailor-made software	Product software Online software Packaged software

Number of copies

Product Software Development

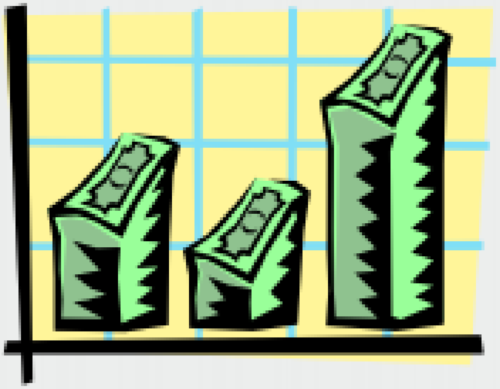
- ❑ **Software is present in a multitude of products**
  - ❑ such as social, business and domestic human–machine interactive systems.
  - ❑ It **includes application software and system software.**
    - ❑ **Application software offers functionality to an end-user**, while
    - ❑ System software consists of low-level programs that interact with a computer at a basic level.

Product Software Development Economic Context

OECD (2008): *"The product software sector is among the most rapidly growing sectors in OECD countries, with strong increases in value added, employment and R&D investment."*

EU

- 2.7 M employees



NL

- Import 1,2 B€, Export 1,6 B€
- VS: 78% market share!
- NL: 40.000 employees in NL companies
- 16 % of 250.000 total ICT employment

Different types of Product Software & Software Based Services

❑. Shrink-wrapped software

- ❑is software on media that are boxed, shrink-wrapped and sold in stores.
- ❑A **shrink wrap** license is an end user agreement (EULA) that is enclosed with **software** in plastic-wrapped packaging.
- ❑Nowadays shrink wrapped software can be also downloaded from the Web.
- ❑Shrink-wrapped software also implies a widely supported standard platform.
- ❑ **Examples:-___________________________________**

Different types of Product Software & Software Based Services

COTS software (commercial off-the-shelf)

- is developed for a whole market instead of individual customers.
- COTS software is either used as is, or moderately personalized within the bounds of the application's ability to be readily altered without changing its original functionality (e.g., modifying its appearance).
- A COTS product, such as an application or a component, is sold, leased, or licensed to the general public, available in multiple, identical copies and used without source code modification.
- offered by a vendor trying to profit from it; supported and evolved by the vendor, who retains the intellectual property rights;

Different types of Product Software & Software Based Services

❑ Packaged software

- ❑ **describes ready-made software products** that can be readily obtained from software vendors and which generally require little modification or customization.
- ❑ Nowadays the term typically refers to upscale enterprise software suites, such as enterprise resource planning (ERP) or customer relationship management (CRM) systems.
- ❑ Large packaged software typically requires weeks or months of deployment and implementation work to set it up for the specific needs of each individual business and often require organizational changes in the business

Different types of Product Software & Software Based Services

- **Commercial software**
  - is a software which is purchased through the retail market and must be licensed before usage.
  - Making copies of this software without the express permission of the author or controlling party is usually prohibited.
- **Standard software**
  - is that software that, by certain consent, is routinely installed by the vendor and/or IT staff on most of computers within certain organizations.
  - includes business applications and operating systems.

Different types of Product Software & Software Based Services

Open Source software

- ❑ is that software which the underlying 'source' code is readily available for inspection, distribution, and modification by any interested person.
- ❑ Most open source software has some type of license agreement for its use that may cover rights to modify, redistribute, use for commercial purposes, and so on.
- ❑ Open source software is not necessarily free in price, redistribution is allowed
- ❑ Both open source software and commercial software companies can make profit. The difference is formed by open source companies only being able to change for services related to the software product.

Product Software Development

Finally:

Product software is defined as a packaged configuration of software components or a software-based service, with auxiliary materials, which is released for and traded in a specific market.

In this definition, we emphasize four concepts:

'**packaged components**',

'software-based services',

'**auxiliary materials**',

and '**release and trading**'.

Product Software Development

- **'Packaged components'** refers to all software discussed above which implies code, executables and web pages.
- **'Software-based services'** covers concepts like ASP sold commercial software services
- **'Auxiliary materials'** consists of software documentation, web pages (in e.g. HTML, XML, etc.), user manuals, training material, brochures and the like.
- Finally, the concept of '**release and trading**' identifies product software's commercial value..

Product Software Development

Examples

- ❑ The two “800 pound gorillas” of **ERP system examples** are **SAP and Oracle**, with the most market share
- ❑ There are versions of SAP’s products that operate on mainframe systems, UNIX or Windows/Intel platforms.
- ❑ Many of Infor’s products use the IBM System*i* (AS/400) platform and integral DB2 database (Infor XA, Infor LX, Infor Prism, Infor System21) but other Infor products reside on WinTel platforms (VISUAL, Syteline).

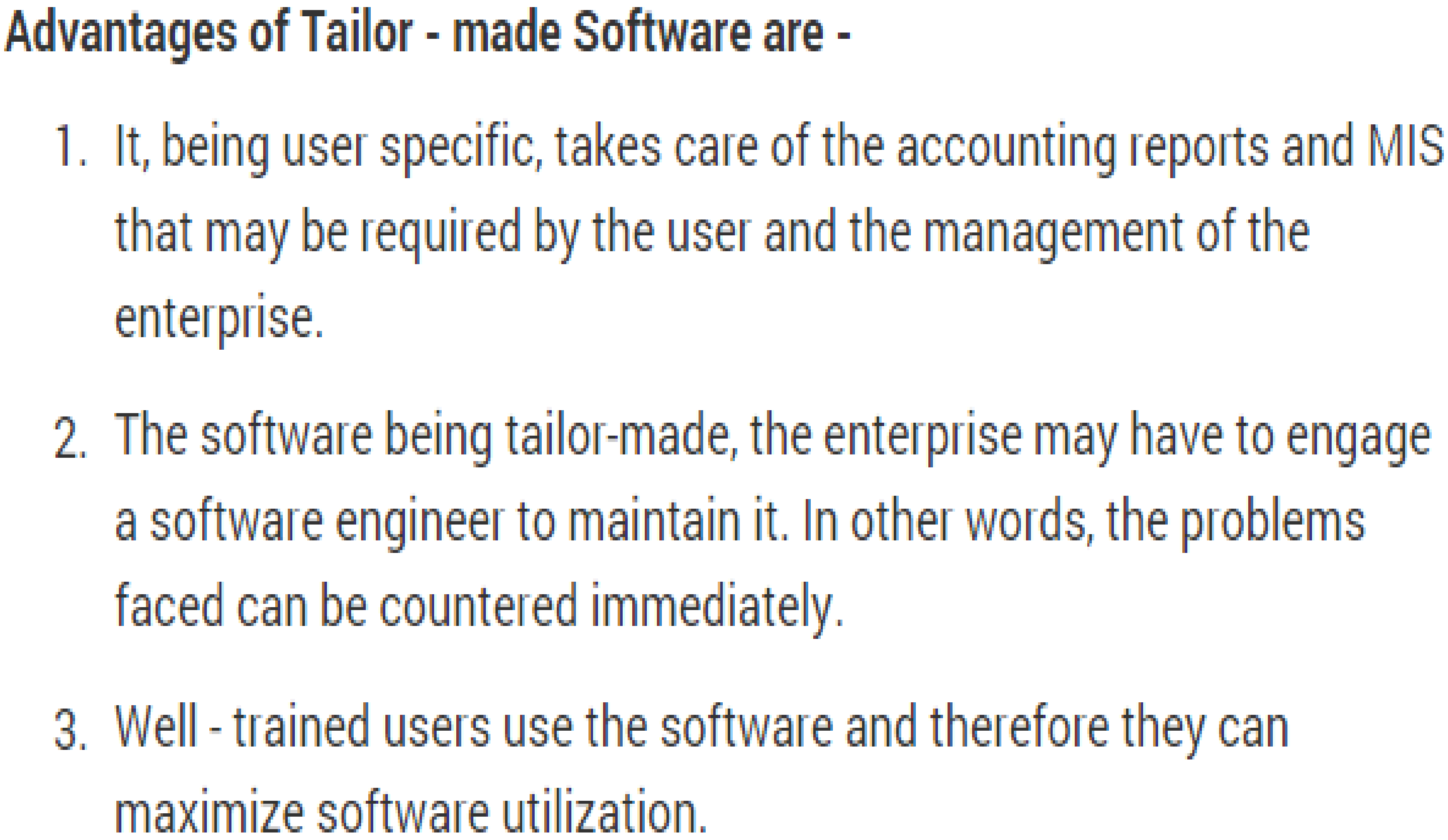
Tailor made Software Development

The term **Tailor made software** refers to designing and developing **user specific software.**

These software's being user specific, are not available **off-the-shelf** but are developed to meet the requirements of the user on the basis of the discussion between users and the developers.

Custom software (also known as bespoke **software** or **tailor-made software**) is **software** that is specially developed for some specific organization or other user

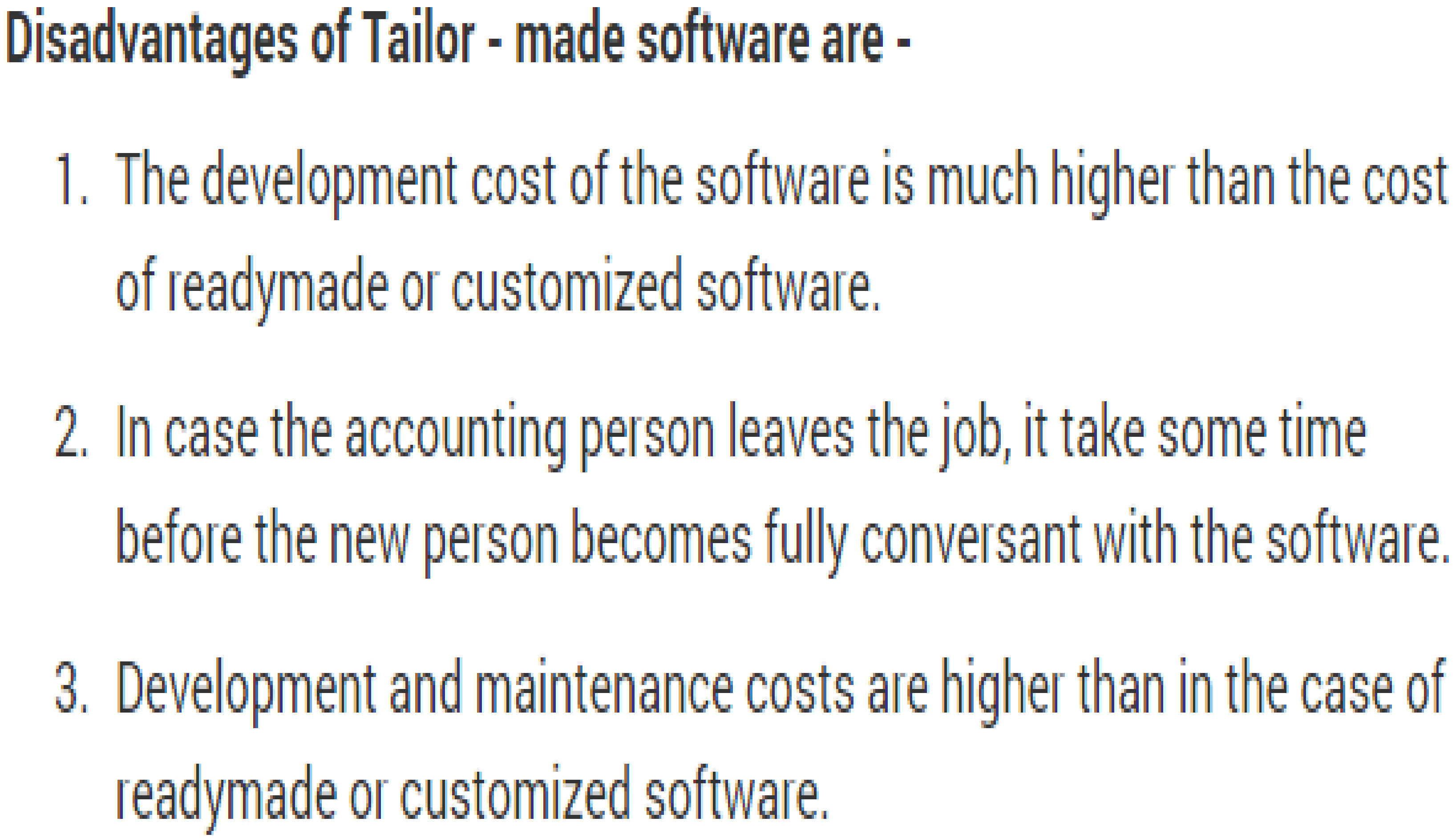
Tailor made Software Development



Advantages of Tailor - made Software are -

1. It, being user specific, takes care of the accounting reports and MIS that may be required by the user and the management of the enterprise.
2. The software being tailor-made, the enterprise may have to engage a software engineer to maintain it. In other words, the problems faced can be countered immediately.
3. Well - trained users use the software and therefore they can maximize software utilization.

Tailor made Software Development



Disadvantages of Tailor - made software are -

1. The development cost of the software is much higher than the cost of readymade or customized software.
2. In case the accounting person leaves the job, it take some time before the new person becomes fully conversant with the software.
3. Development and maintenance costs are higher than in the case of readymade or customized software.

Examples of Tailor made Software

Web application:

- A tailor-made web application is software customized to the clients' needs, so that the system will fit perfectly into their business processes.
- The software will be installed on a web server, so that the web application can be accessed via the internet anywhere in the world.
- If required, we will also be able to make the application compatible with,
- for example, mobile phones or tablets. Another advantage of this software is that the usage can be scaled:
- it is relatively easy to increase the number of users by using a larger web server. It is also possible to make links to existing applications.

Examples of Tailor made Software

Supply system

- For Invictawatch Europe, i-design has developed a user-friendly and web-based supply system.
- With this system it is easy to manage supplies. Item files in this system can be expanded to include, for example, marketing support information.
- As such, the system can give procurement advice, based on the sales and expected sales. It is also possible to enter orders using various channels, such as the website, intranet, barcode scanner or cash till.
- The supply system can also be linked up with an administration package

Examples of Tailor made Software

Number registration system

- ❑ The number registration system that i-design developed for Saxion next, a privately-owned higher education institution, can be used,
- ❑ for example, for club membership administration or student administration at a school.
- ❑ By using this number registration system, it is also possible to print out individual school certificates.
- ❑ The registration and invoicing is based on a web-based system

Product Software vs. Tailor made software

Product Software	Tailor made software
developed for a specific market and sold many times within that market	made specifically for an organization and sold only once
can be grouped into ➢ business-to-business product software ➢ business-to-consumer software .	further classified into ➢ contractual tailor-made software ➢ in-house tailor-made software
➢ market introduction requires precise synchronization of dependable software engineering activities for product software	➢ market oriented instead of working for one customer in tailor-made software.

Product Software vs. Tailor made software

Product Software	Tailor made software
➢ product software requires installation and usage in different organizations, with different hardware and software platforms	tailor-made software has fixed conditions of the hardware and software platforms.
➢ the vendor normally retains ownership of the software and auxiliary materials and licenses customers to use the software	➢ a tailor made piece of software will generally be owned by the customer, not the vendor.
The differences in customers, markets, and goals of product software and tailor-made software are so diverse Looking from a software engineering point of view, a number of basic differences between product software and tailor-made software can be observed.	

Product Software vs. Tailor made software

<table>
<tr><th>Product Software</th><th>Tailor made software</th></tr>
<tr><td colspan="2">Developing product software is different with developing tailor-made software from the development life cycle perspective</td></tr>
<tr><td>➢ Product software is continuously under development. After finishing one release, there will be others. Some bugs will be fixed in a later release or service packs or patches will be provided</td><td>➢ At a certain time, a tailor-made program is 'finished', after which the maintenance phases starts</td></tr>
<tr><td colspan="2">Developing product software and developing tailor-made software can thus use different development methods</td></tr>
</table>

Product Software vs. Tailor made software

Product Software	Tailor made software
A future proof architecture is crucial for developing product software.	
➢ Architecture design is so crucial for developing product software	➢ Architecture design is not so crucial for developing tailor-made software
➢ must be delivered to a large number of unknown customers (potentially millions) and implemented on a huge number of unknown systems. ➢ obviously too expensive to manually deliver or customize product software for all potential customers ➢ So the delivery and implementation to market is an important consideration for product software	➢ In developing tailor-made software, the cost and specifics of delivery and implementation will have been taken into account within the contract. ➢ It will involve installation (probably manual) of the software at a limited number of known customer sites.