

SECD2613

System Analysis and Design



TOPIC I

Fundamentals of Systems Analysis & Design

www.utm.my
innovative • entrepreneurial • global



univteknologimalaysia



utm_my



utmofficial

PART 1

- Organization impact on Information System
- Types of Information Systems

■ OBJECTIVES

By the end of this lecture, you will be able to:

- Recall the basic types of computer-based systems that a systems analyst needs to address
- Understand how users working in context with new technologies change the dynamics of a system
- Realize what the many roles of the systems analyst are
- Know the steps of the SDLC as they relate to HCI and how to apply them to a real system
- Understand what CASE tools are and how they help a systems analyst
- Explore other methodologies such as object-oriented systems design and prototyping

■ INFORMATION – A KEY RESOURCE

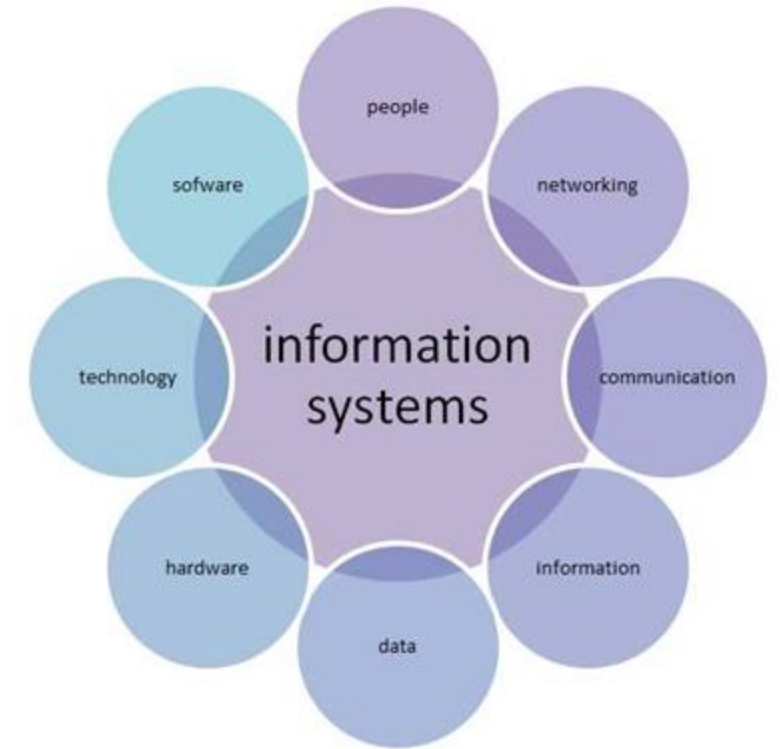
- Fuels business and can be the critical factor in determining the success or failure of a business
- Needs to be managed correctly
- Managing computer-generated information differs from handling manually produced data

■ WHAT IS INFORMATION SYSTEMS (IS)?

A system which assembles, stores, processes and delivers information relevant to an organization, in such a way that the information is accessible and useful to those who wish to use it, including managers, staff, clients and citizens.

An IS is human activity (social system) which may or may not involve the use of computer systems

...Buckingham et al. (1987)



■ WHAT & WHY SYSTEM ANALYSIS AND DESIGN (SAD)?

Systems analysis and design is a systematic approach to identifying problems, opportunities, and objectives; analyzing the information flows in organizations; and designing computerized (in this case) information systems to solve a problem



■ WHAT & WHY SYSTEM ANALYSIS AND DESIGN (SAD)?

- System development needs proper planning.
- There is a cost in system development.
- It is time consuming and needs project management.
- SAD is about problem solving that needs to be creative, critical, and innovative.

■ ORGANIZATION AS SYSTEMS

Organization as systems, is composed of subsystems and so on.

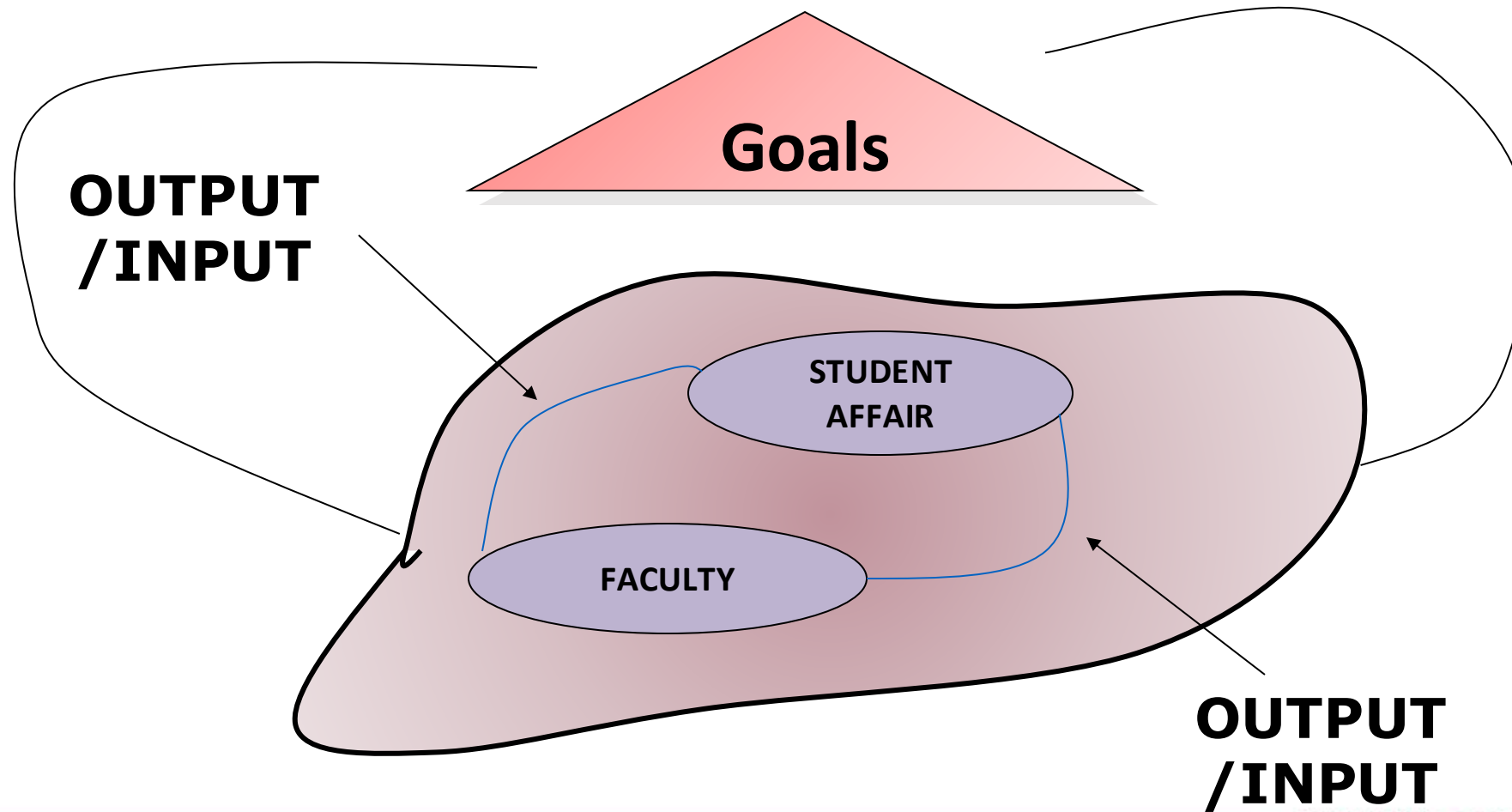
It is mainly involves levels of management and culture.

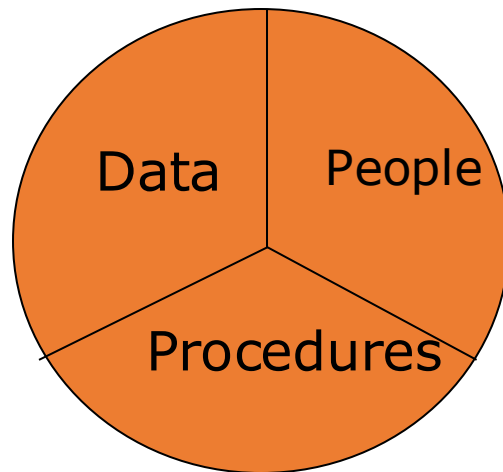
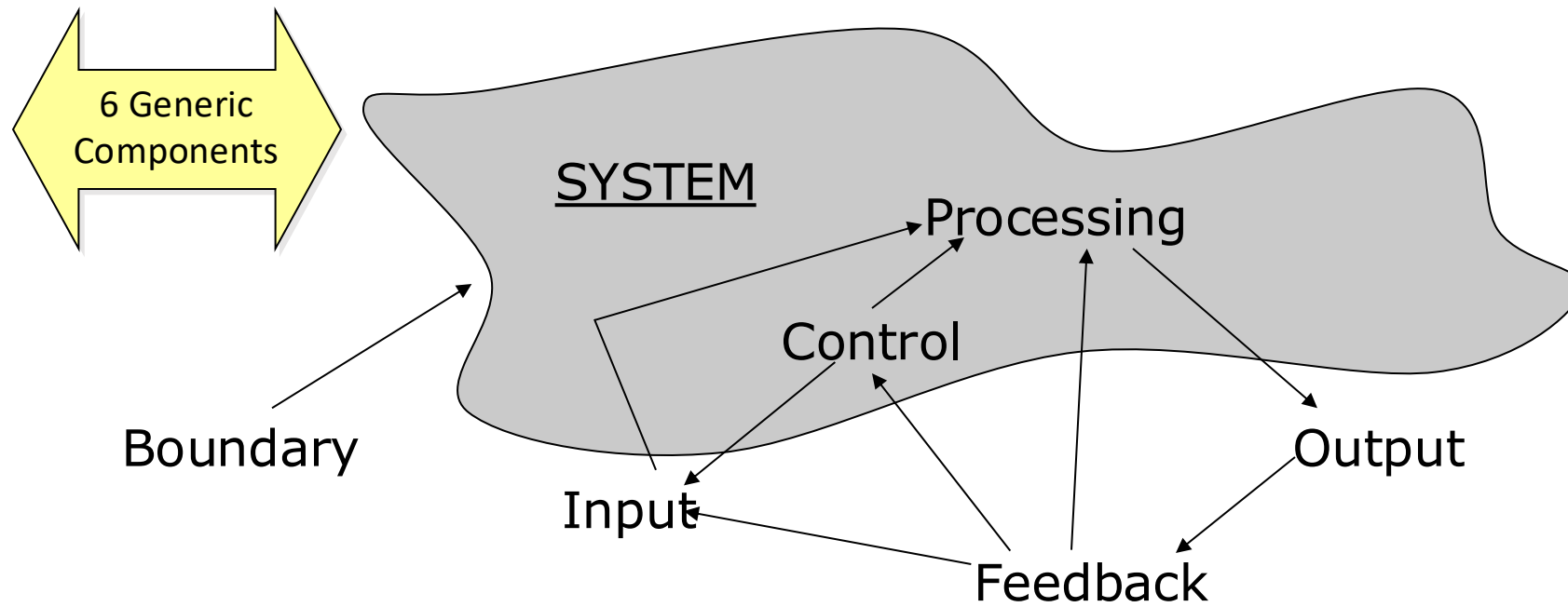
These will impact the information system development.

The different levels of management will produce the different levels of decisions.

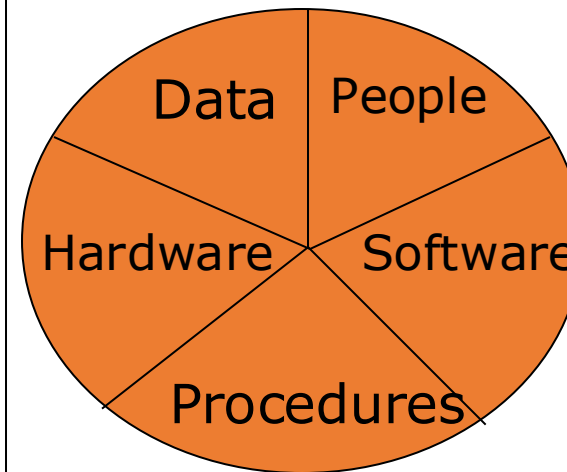
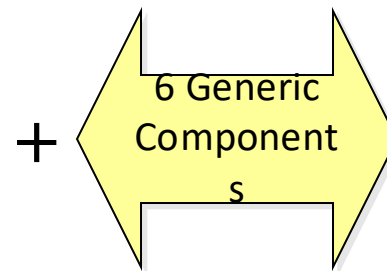
The culture influences the way people in subsystems interrelate.

■ ORGANIZATION AS SYSTEMS

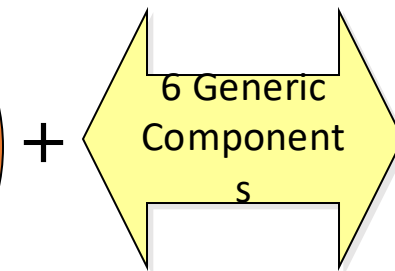




Information Systems

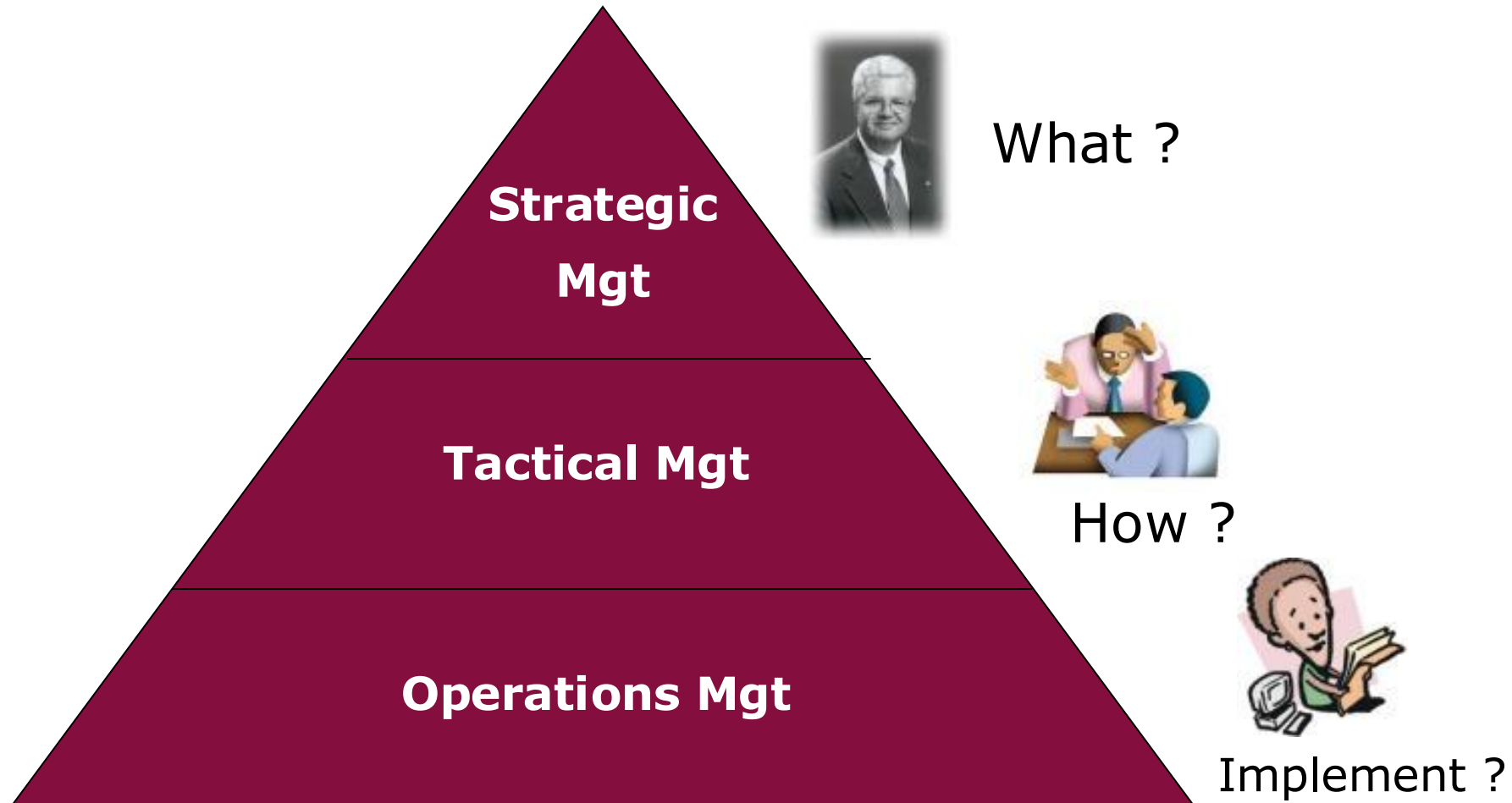


Automated Information Systems



■ ORGANIZATION AS SYSTEMS

Level of Management



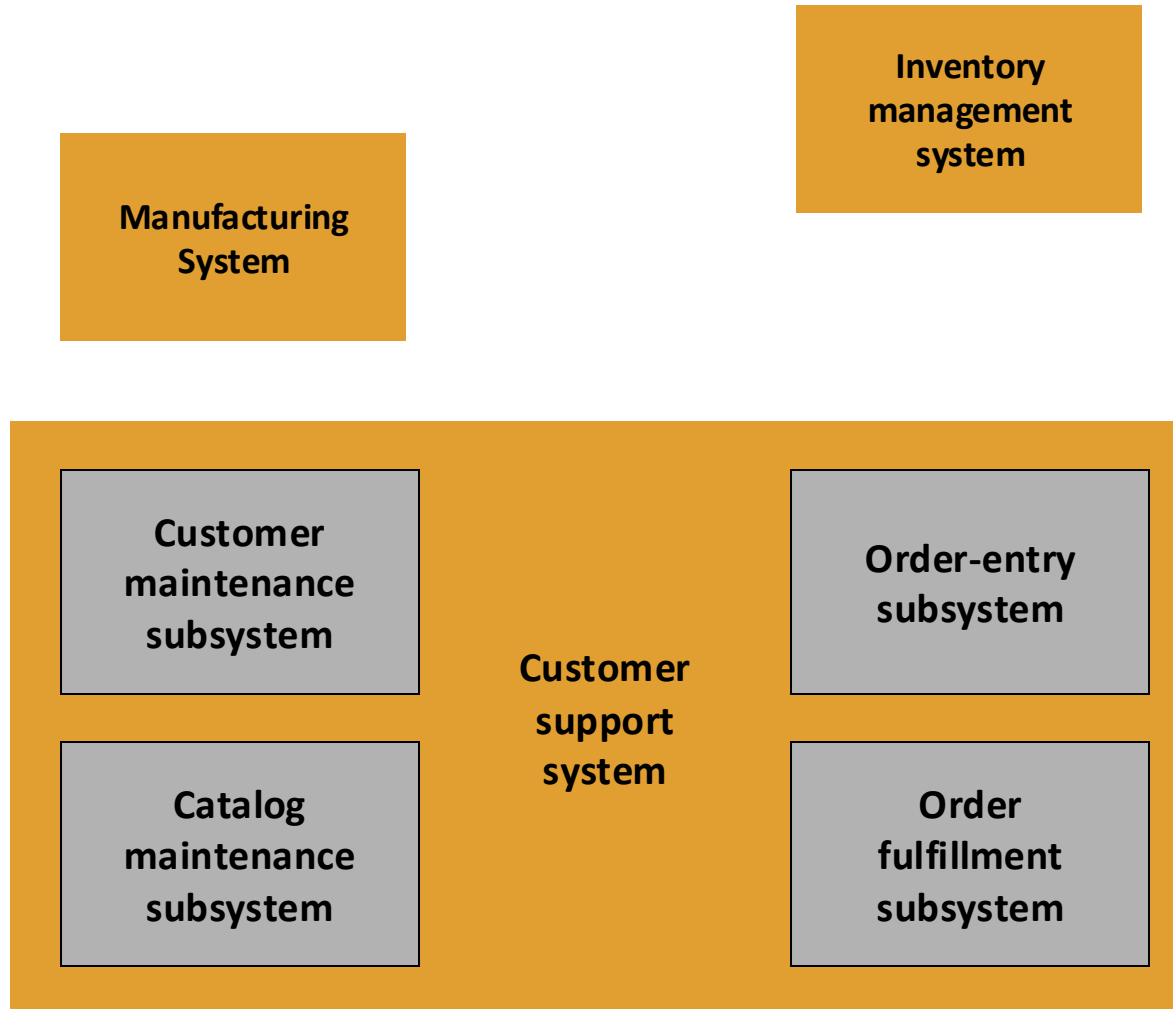
System: a collection of interrelated components that function together to achieve some outcome

Information System: a collection of interrelated components that collect, process, store, and provide as output the information needed to complete business tasks.

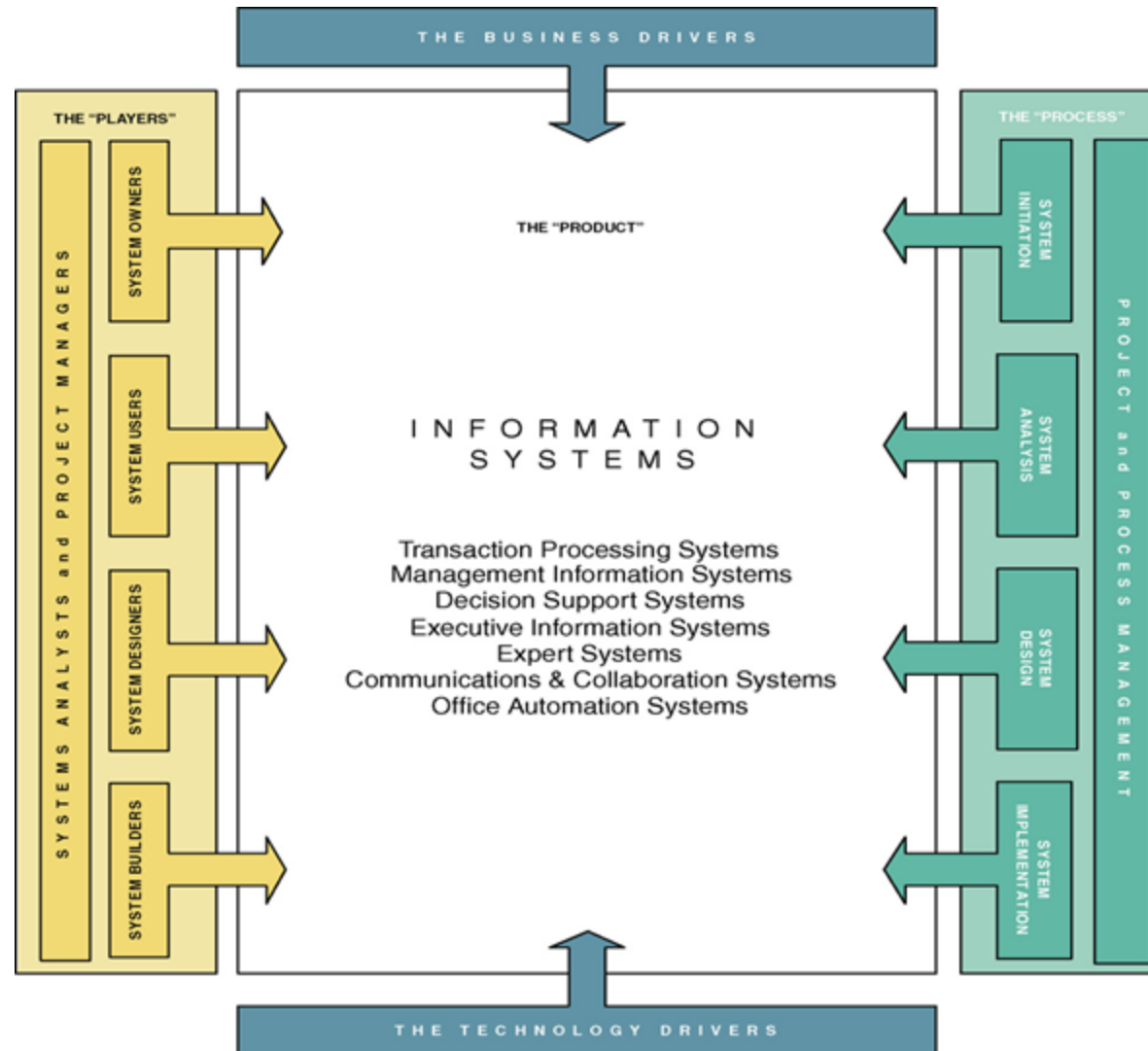
Sub-system: a system that is part of a larger system

Super-system: a larger system that contains other systems

PRODUCTION SYSTEM (SUPER-SYSTEM)



■ A FRAMEWORK FOR IS



■ BUSINESS DRIVERS FOR TODAY'S INFORMATION SYSTEMS

- Globalization of the Economy
- Electronic Commerce and Business
- Security and Privacy
- Collaboration and Partnership
- Knowledge Asset Management
- Continuous Improvement and Total Quality Management
- Business Process Redesign

■ BUSINESS DRIVERS FOR TODAY'S INFORMATION SYSTEMS

- Networks and the Internet
- Mobile and Wireless Technologies
- Object Technologies
- Collaborative Technologies
- Enterprise Applications

MAJOR TOPICS

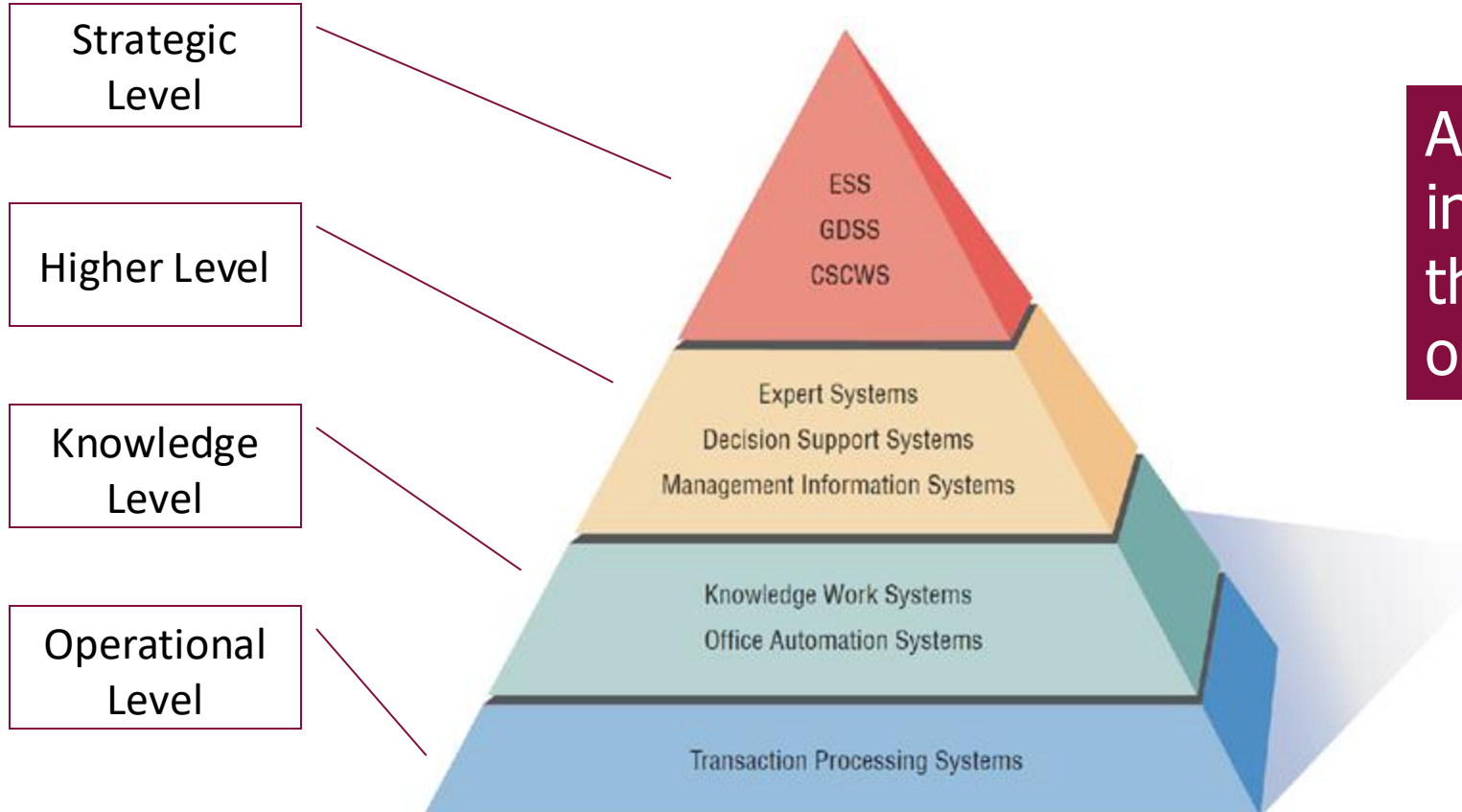
- Fundamentals of different kinds of information systems
- Roles of systems analysts
- Phases in the systems development life cycle as they relate to Human-Computer Interaction (HCI) factors
- Computer-Aided Software Engineering (CASE) tools



■ TYPES OF SYSTEMS

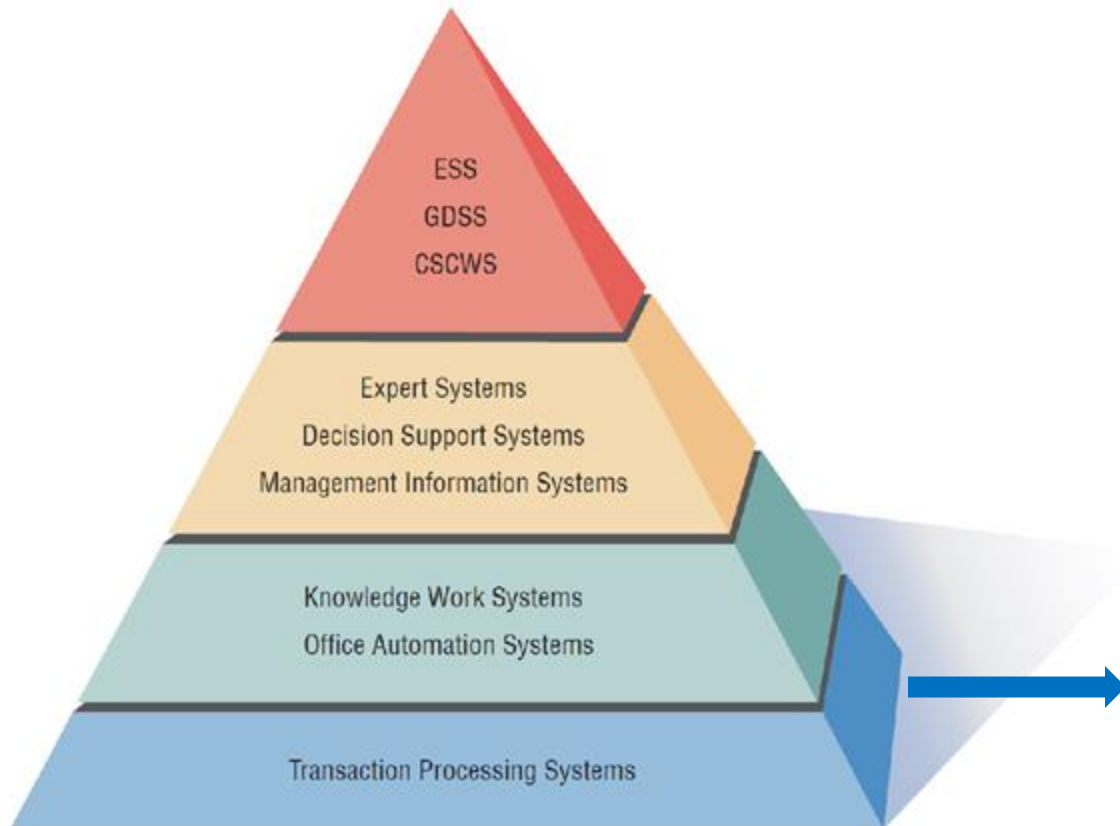
Systems analysts recommend, design, and maintain many types of systems for users:

- Transaction Processing Systems (TPS)
- Office Automation Systems (OAS)
- Knowledge Work Systems (KWS)
- Management Information Systems (MIS)
- Decision Support Systems (DSS)
- Expert Systems (ES)
- Executive Support Systems (ESS)
- Group Decision Support Systems (GDSS)
- Computer-Supported Collaborative Work Systems (CSCWS)



A systems analyst may be involved with any or all of these systems at each organization level

■ OPERATIONAL LEVEL



Transaction Processing System (TPS)

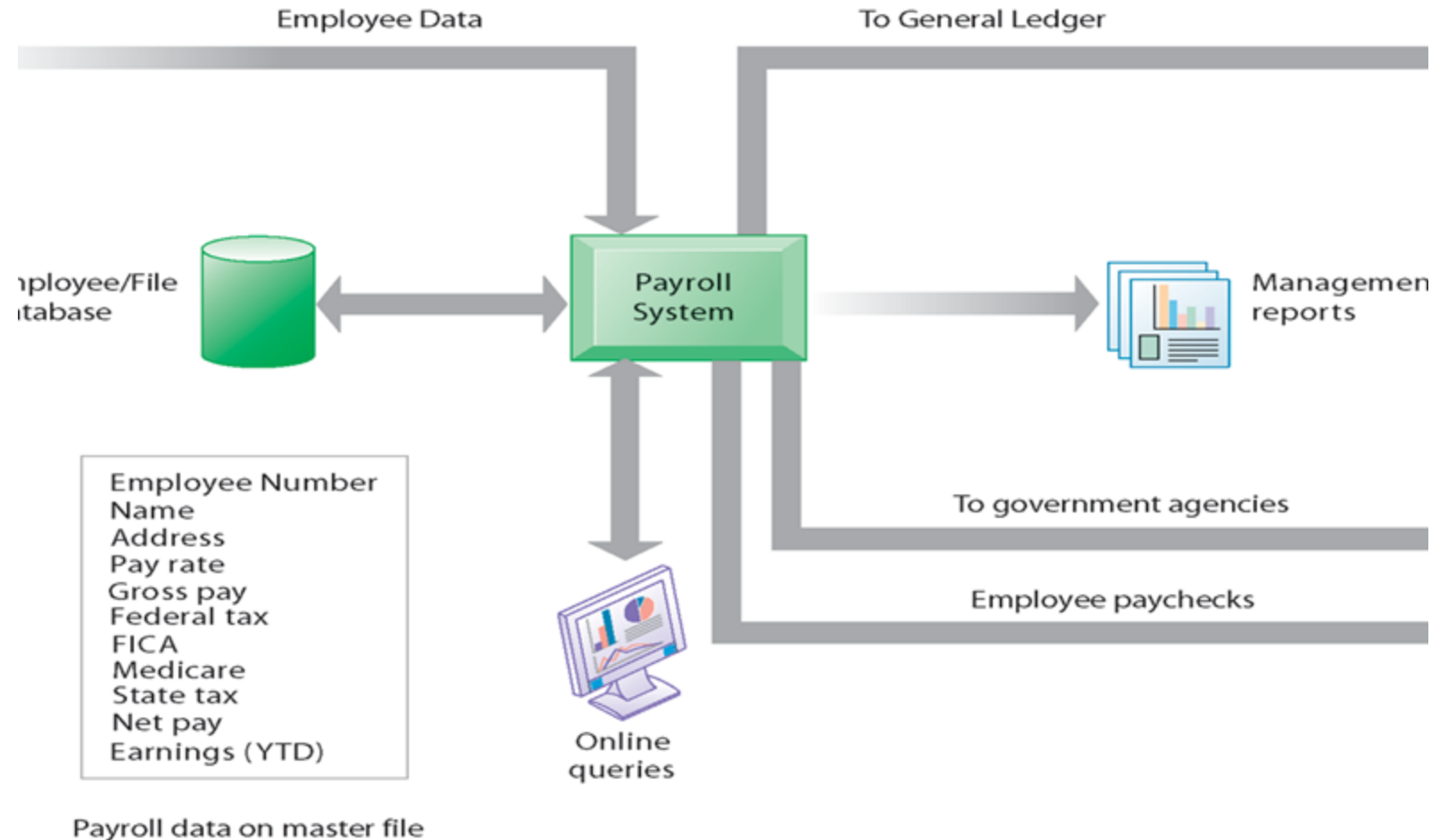
- Process large amounts of data for routine business transactions
- Boundary-spanning
- Support the day-to-day operations of the company
- Examples: Payroll Processing, Inventory Management

■ OPERATIONAL LEVEL

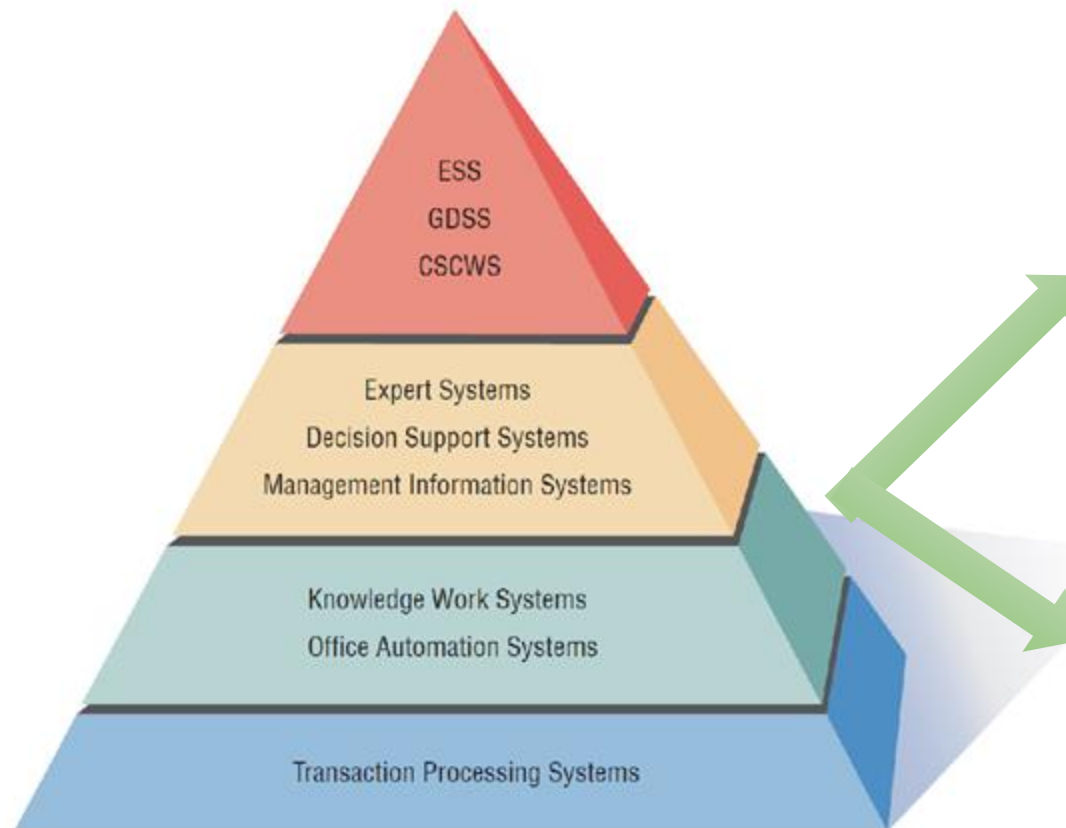
A Payroll TPS

A TPS for payroll processing captures employee payment transaction data (such as a time card).

System outputs include online and hard-copy reports for management and employee paychecks.



■ KNOWLEDGE LEVEL



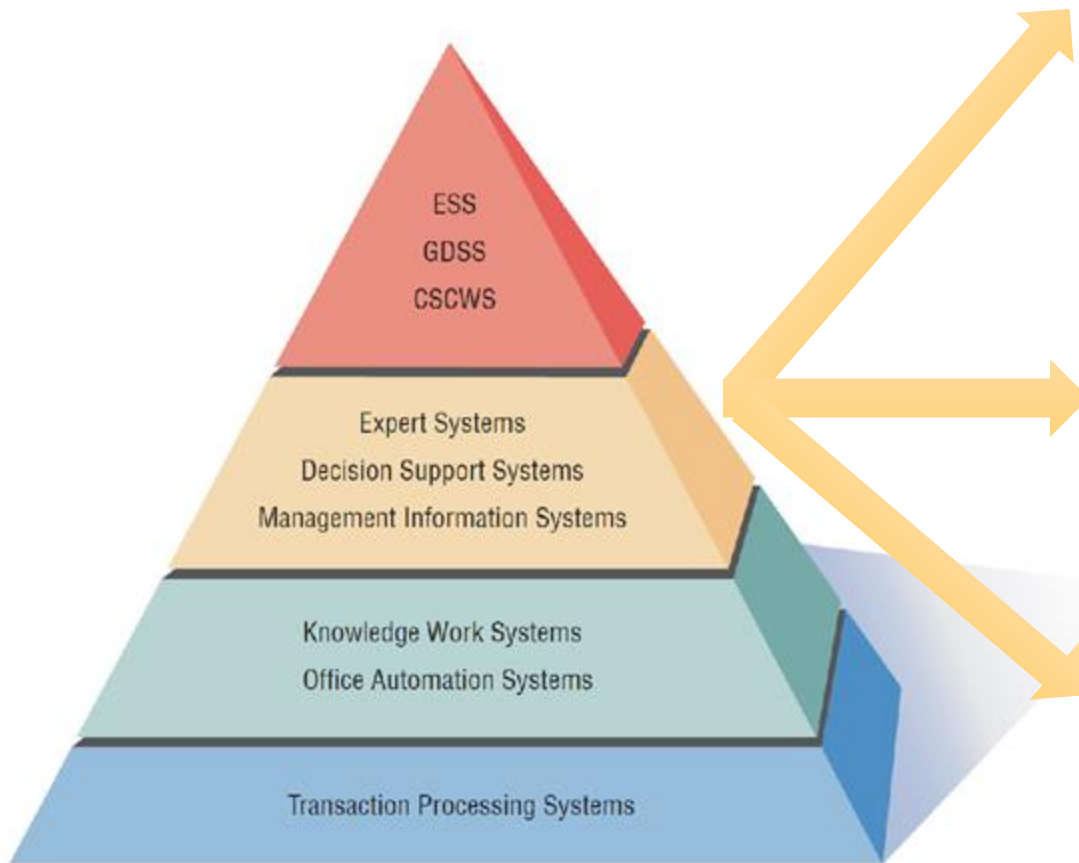
Office Automation System (OAS)

- Supports data workers who share information, but do not usually create new knowledge
- Examples: Word processing, Spreadsheets, Desktop publishing, Electronic scheduling, Communication through voice mail, Email, Video conferencing

Knowledge Work System (KWS)

- Supports professional workers such as scientists, engineers, and doctors
- Examples: computer-aided design systems, virtual reality systems, investment workstations

■ HIGHER LEVEL



Management Information System (MIS)

- Support a broad spectrum of organizational tasks including decision analysis and decision making
- Examples: profit margin by sales region, expenses vs. budgets

Decision Support System (DSS)

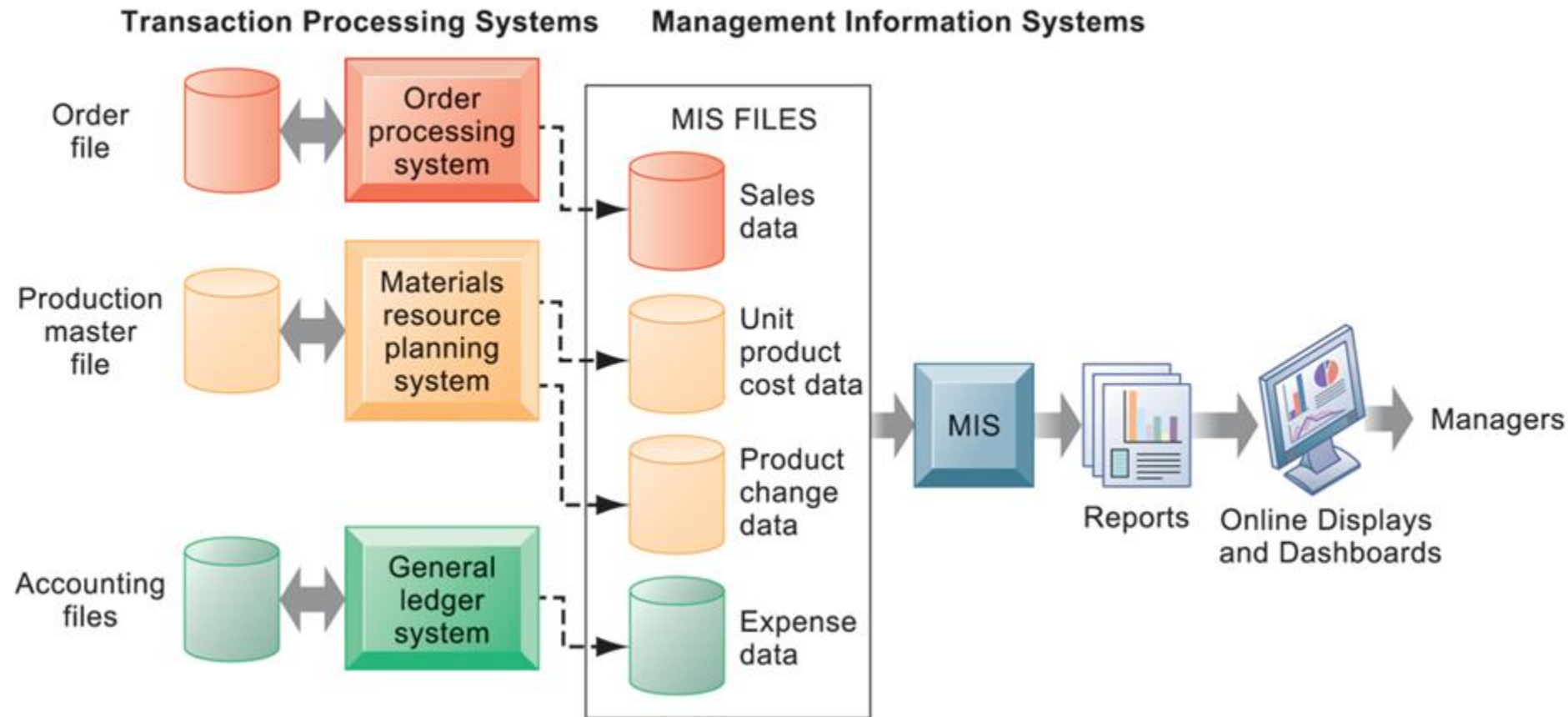
- Aids decision makers in the making of decisions
- Examples: financial planning with what-if analysis, budgeting with modeling

Expert System (ES)

- Captures and uses the knowledge of an expert for solving a particular problem which leads to a conclusion or recommendation
- Examples: MYCIN, XCON

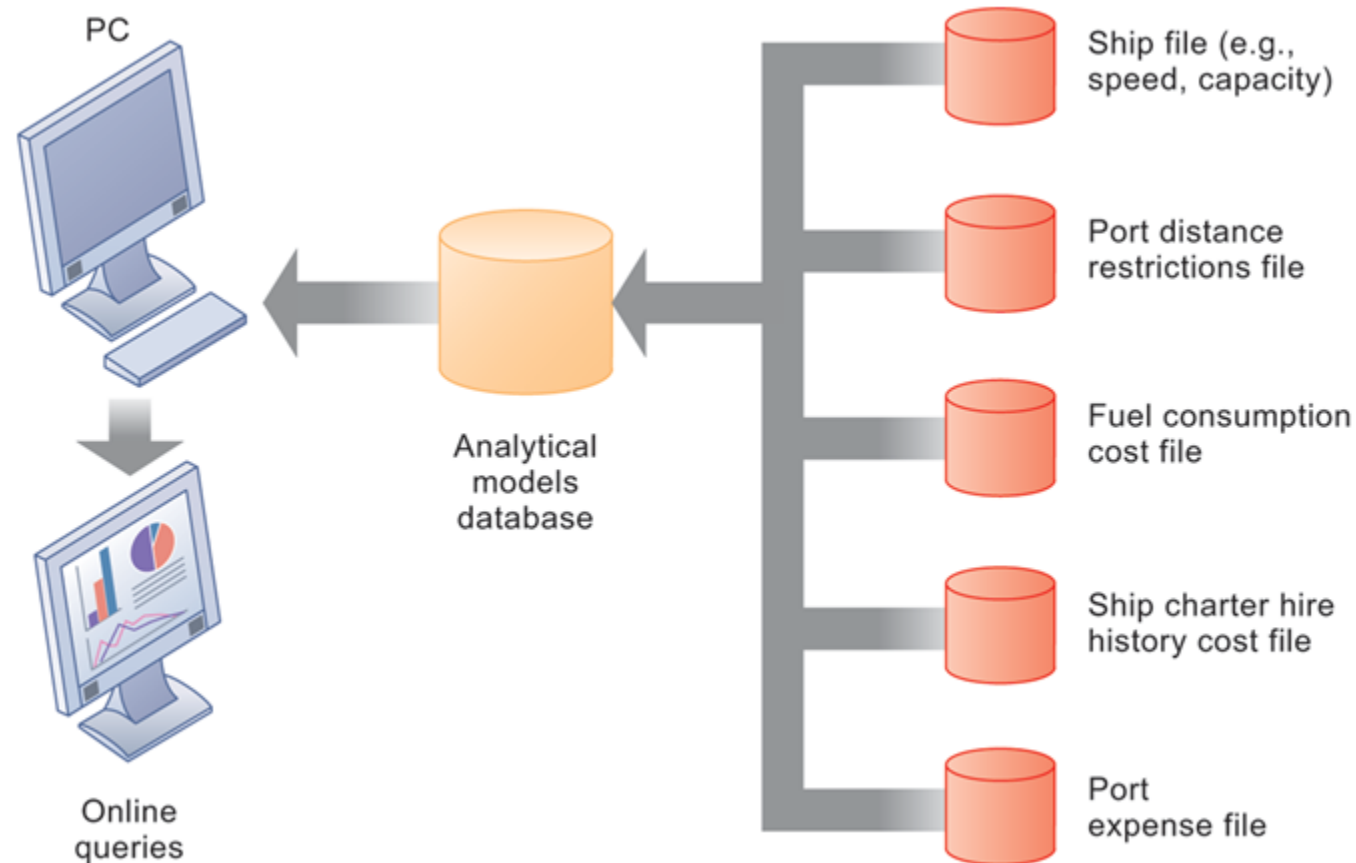
■ HIGHER LEVEL

Management Information System (MIS)

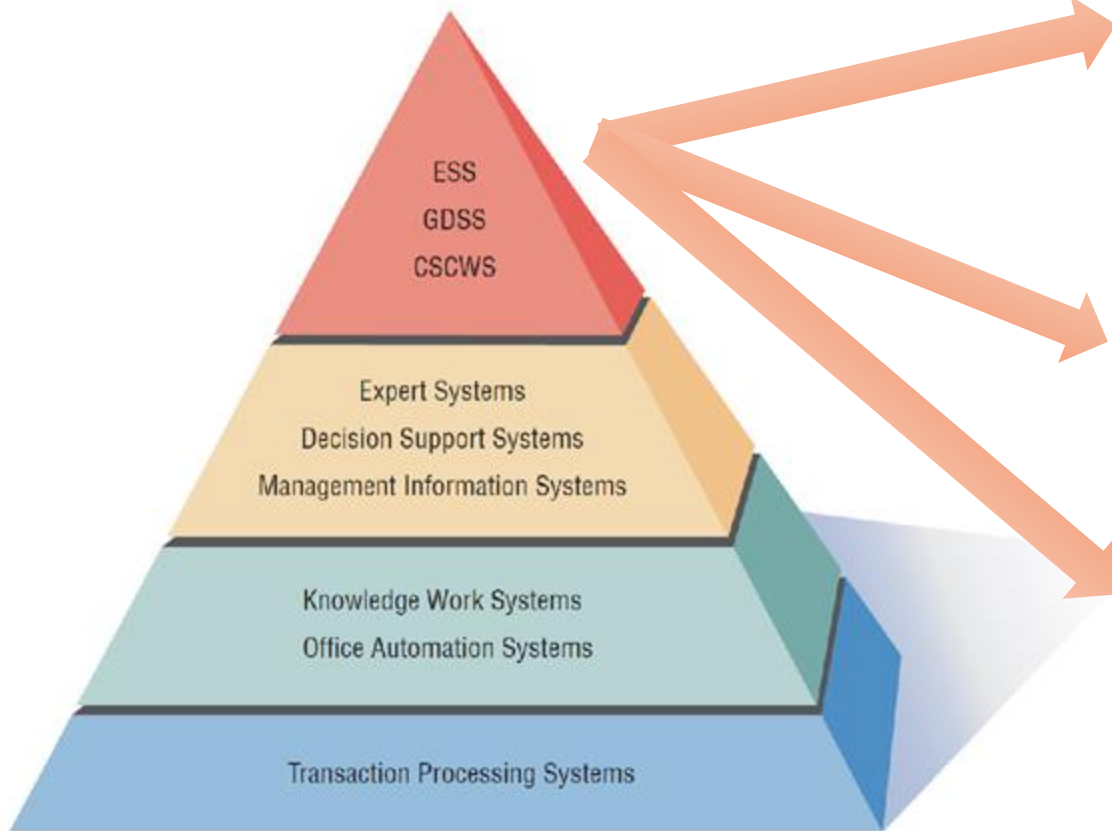


■ HIGHER LEVEL

Decision Support System (DSS)



STRATEGIC LEVEL



Executive Support System (ESS)

- Helps executives to make unstructured strategic decisions in an informed way
- Examples: drill-down analysis, status access

Group Decision Support System (GDSS)

- Permit group members to interact with electronic support
- Examples: email, Google Doc

Computer-Supported Collaborative Work System (CSCWS)

- CSCWS is a more general term of GDSS
- May include software support called "*groupware*" for team collaboration via network computers
- Example: video conferencing, Web survey system

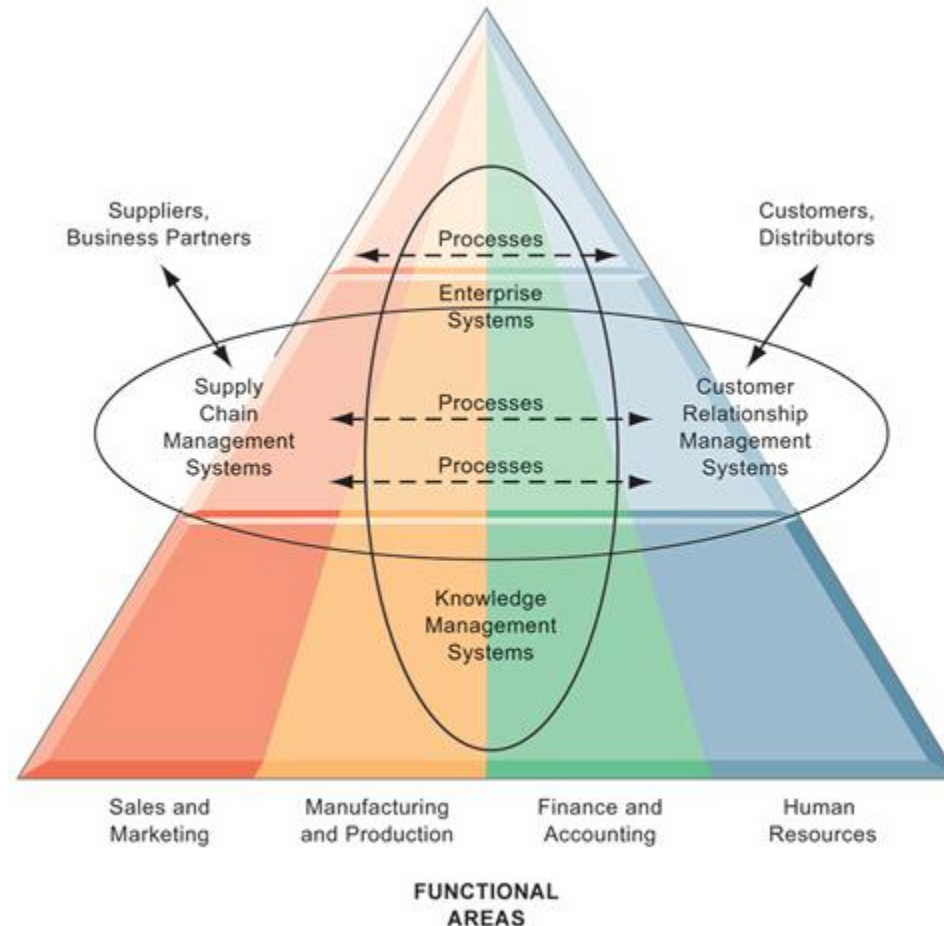
INTEGRATING NEW TECHNOLOGIES INTO TRADITIONAL SYSTEMS

- Ecommerce and Web Systems
- Enterprise Resource Planning Systems
- Wireless Systems
- Open Source Software
- Need for Systems Analysis and Design



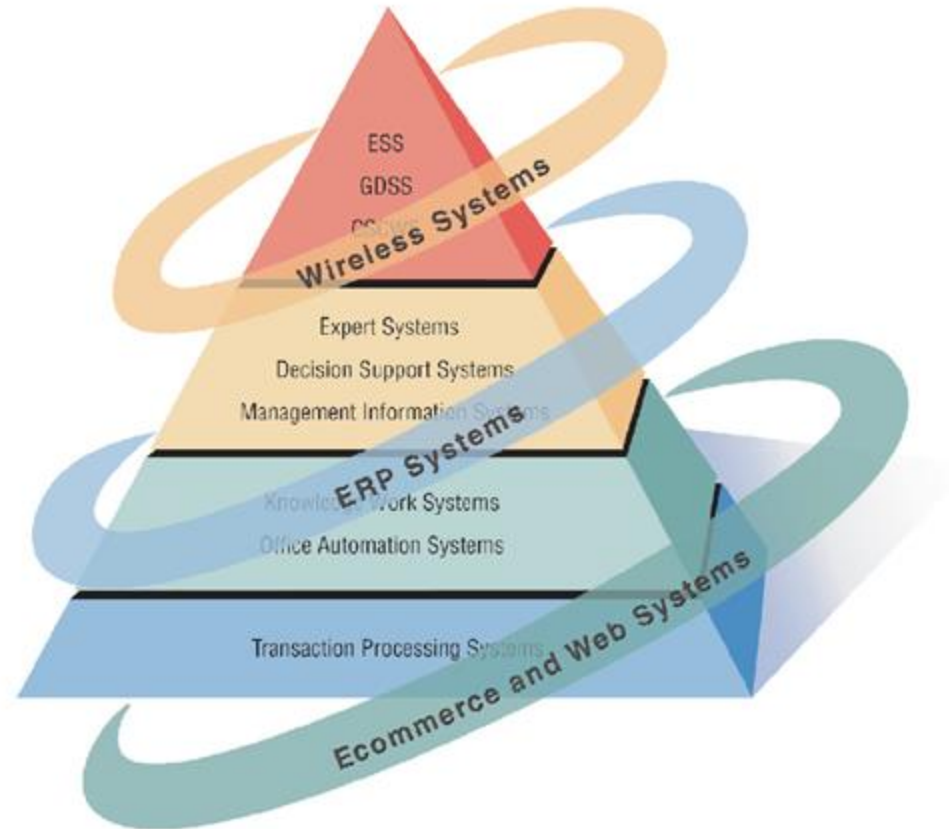
■ ENTERPRISE APPLICATION ARCHITECTURE

Enterprise applications automate processes that span multiple business functions and may extend outside the organization.



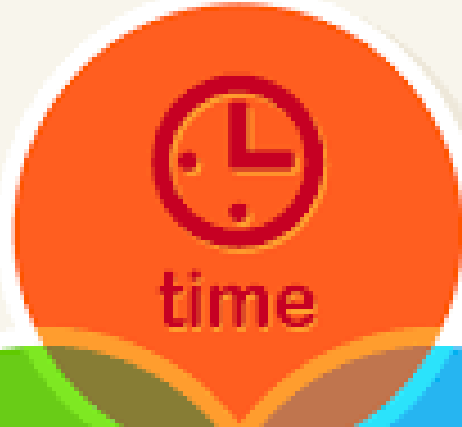
ENTERPRISE APPLICATION ARCHITECTURE

- Systems analysts need to be aware that integrating technologies affects all types of systems.



■ E-COMMERCE AND WEB SYSTEMS

BENEFITS



time

5¹
cost

quality



Increasing user awareness of the availability of a service, product, industry, person, or group



The possibility of 24-hour access for users



Creating a system that can extend globally rather than remain local, thus reaching people in remote locations without worry of the time zone in which they are located



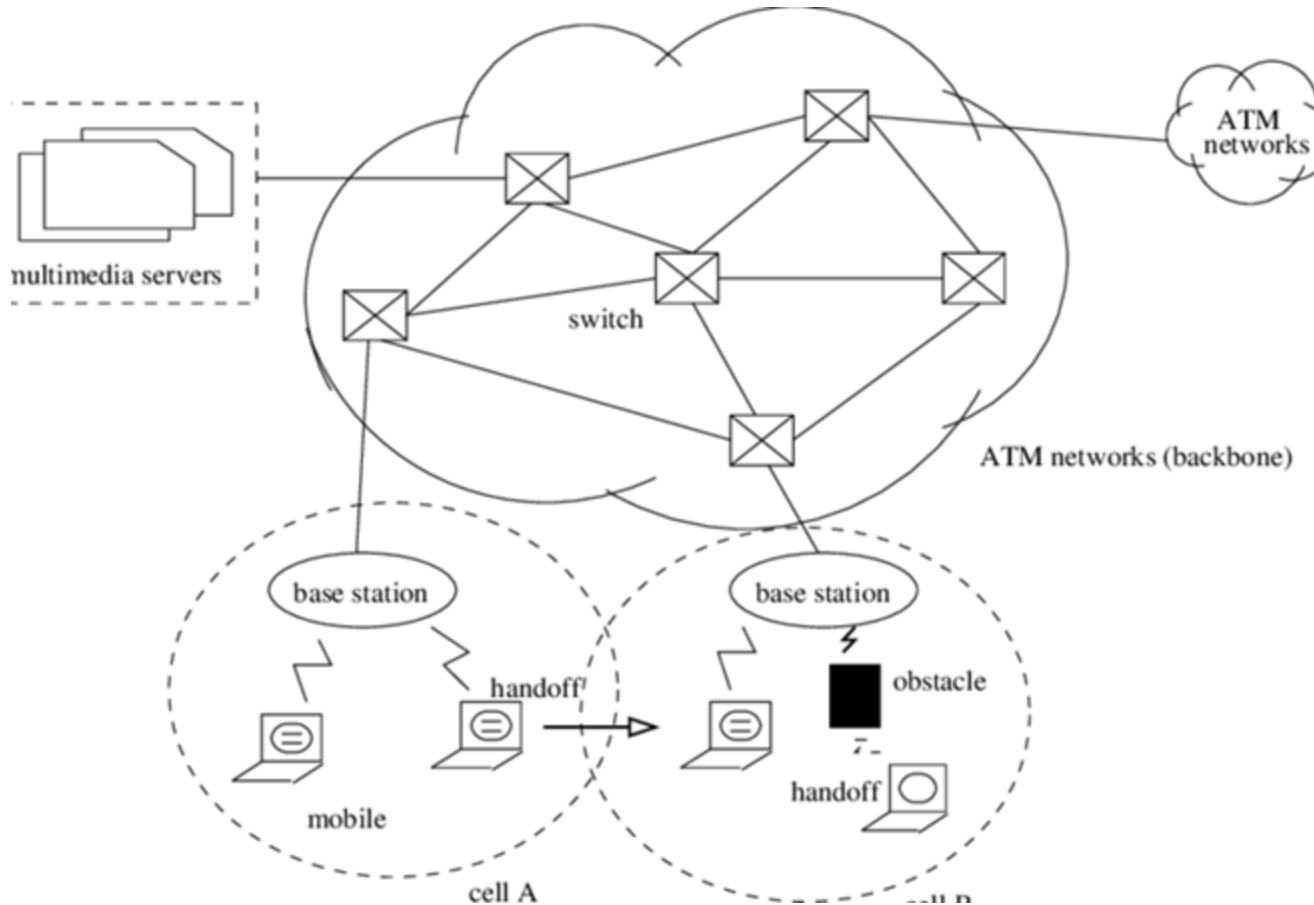
Improving the usefulness and usability of interface design

■ ENTERPRISE RESOURCE PLANNING SYSTEMS (ERP)

- Performs integration of many information systems existing on different management levels and within different functions
- Example: SAP, Oracle



■ WIRELESS SYSTEMS



- System analyst may be asked to design standard or wireless communication networks that integrate voice, video and email into organizational intranets or industry extranets
- System analyst may also be asked to develop intelligent agents
- Example: Microsoft's new software based on Bayesian statistics
- Wireless communication is referred as m-commerce (mobile commerce)

■ OPEN SOURCE SOFTWARE

- An alternative of traditional software development where proprietary code is hidden from the users
- Open source software is free to distribute, share and modify
- Characterized as a philosophy rather than simply the process of creating new software
- Example: Linux Operating System, Apache Web Server, Mozilla Firefox Web browser



PART 2

- System analyst role
- SDLC

NEED FOR SYSTEM ANALYSIS AND DESIGN

- Installing a system without proper planning leads to great user dissatisfaction and frequently causes the system to fall into disuse
- Lends structure to the analysis and design of information systems
- A series of processes systematically undertaken to improve a business through the use of computerized information systems

■ SYSTEM ANALYST



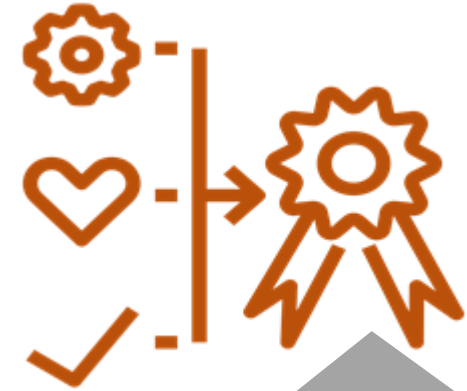
ENVIRONMENT

System analyst (SA) studies the problems and needs of an organization to determine how people, data, process, communications and information technology can best accomplish improvements for the business (organizations)



ROLES

- The analyst must be able to work with people of all descriptions and be experienced in working with computers
- Three primary roles:
 1. Consultant
 2. Supporting Expert
 3. Agent of change



QUALITIES

- Problem solver
- Communicator
- Strong personal and professional ethics
- Self-disciplined and self-motivated

SYSTEM ANALYST

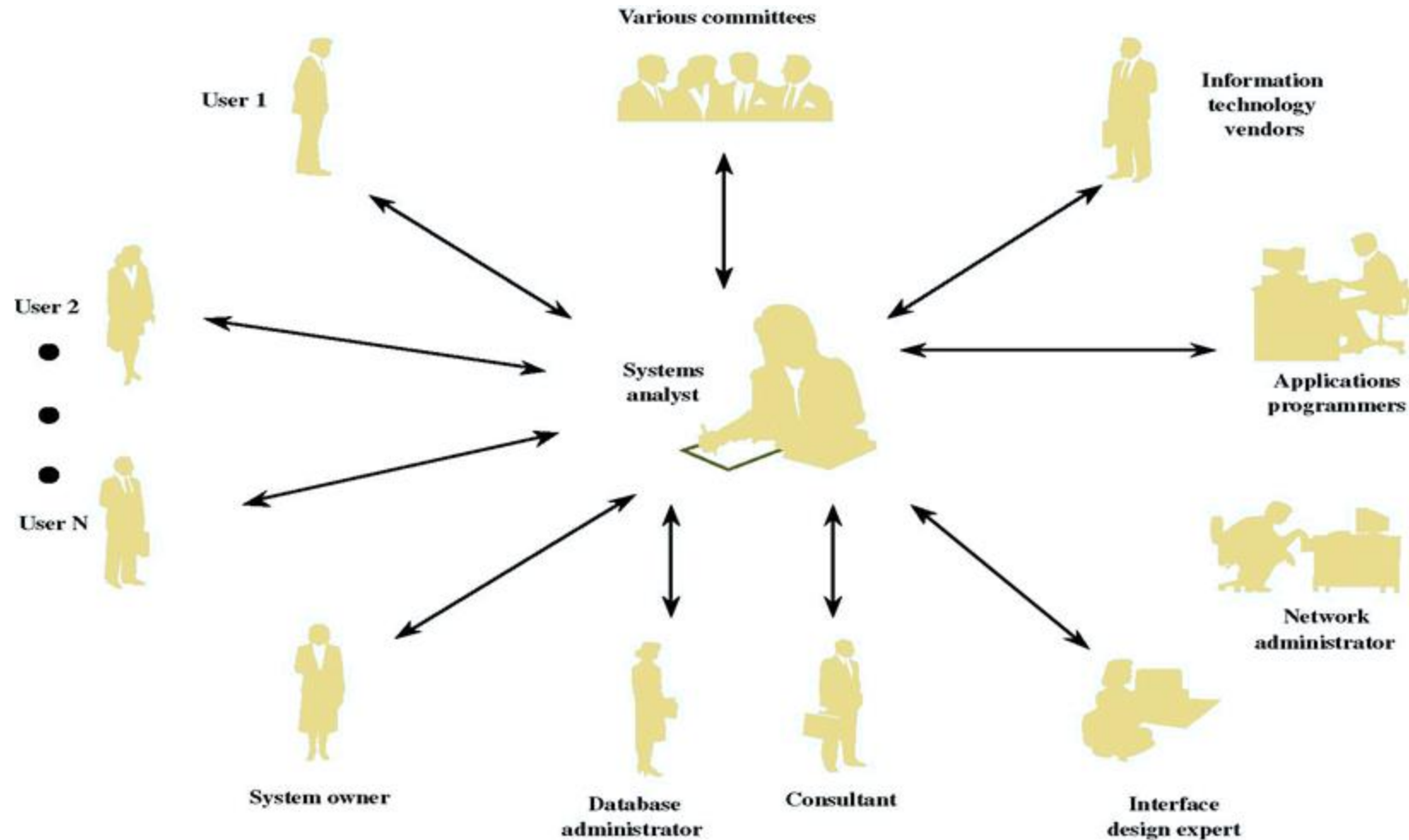


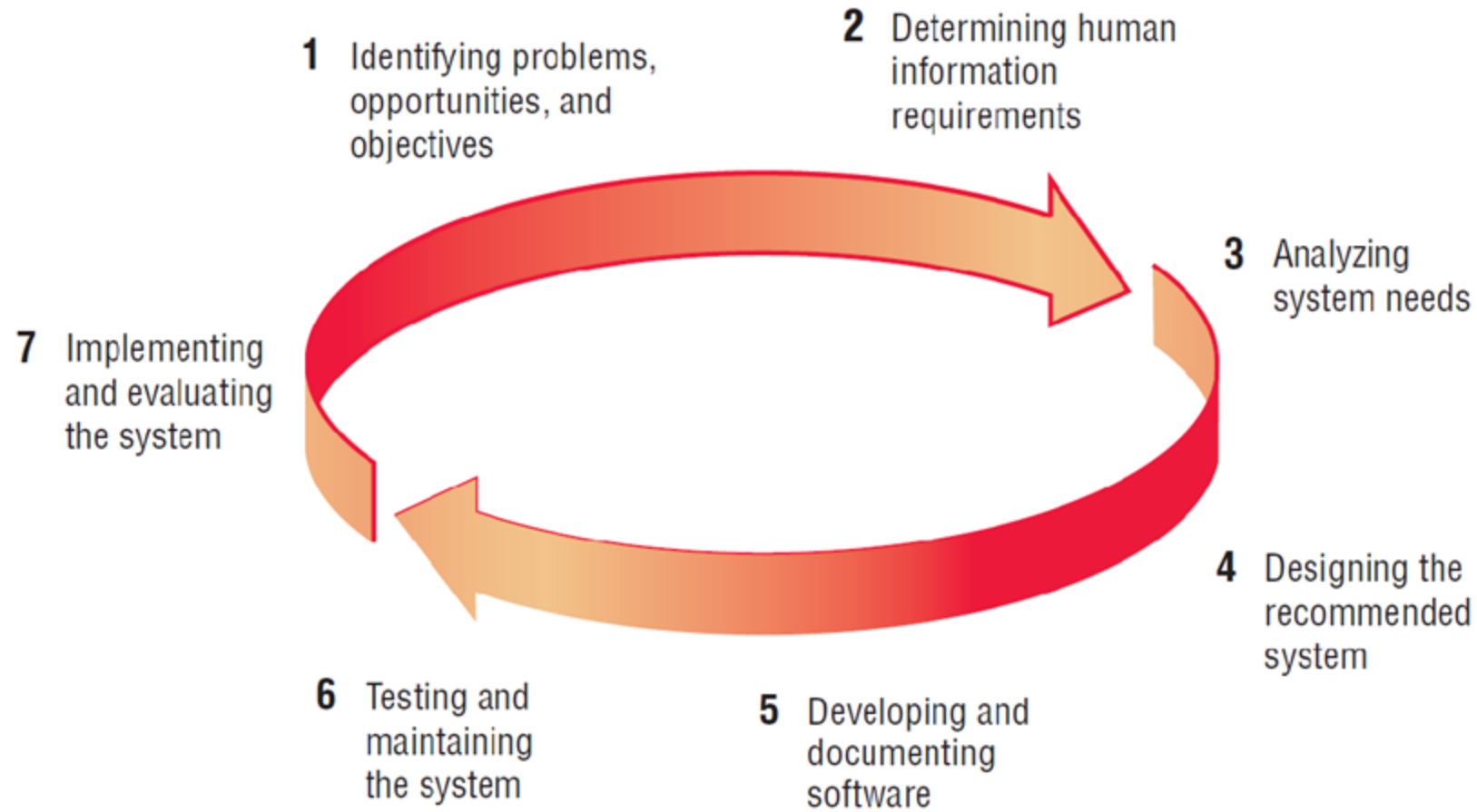
Figure 1.3 The Systems Analyst as a Facilitator

SYSTEM DEVELOPMENT LIFE CYCLE (SDLC)

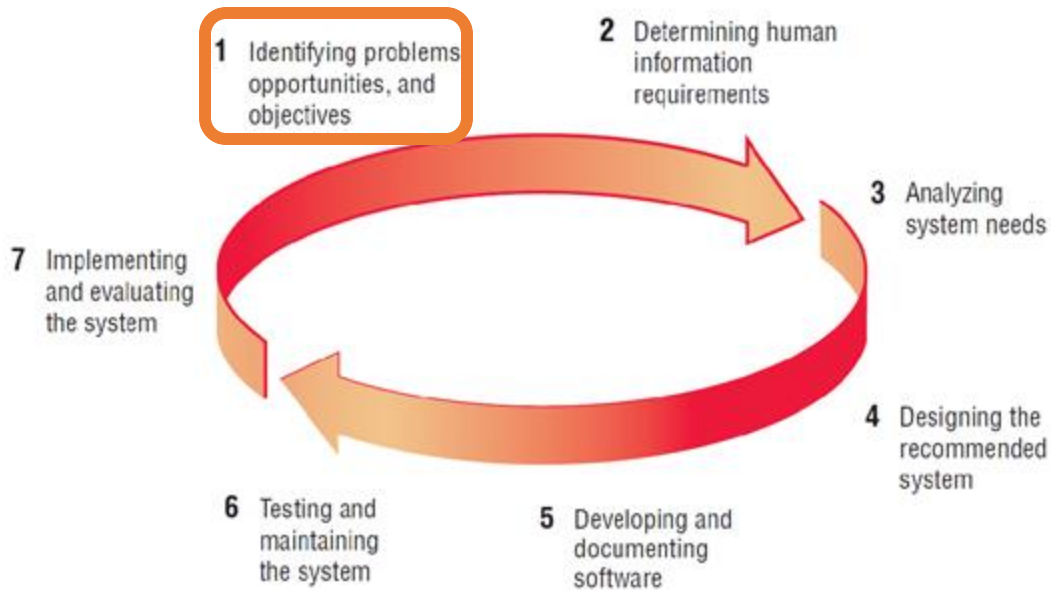
■ SYSTEM DEVELOPMENT LIFE CYCLE (SDLC)

- The systems development life cycle is a phased approach to solving business problems
- Developed through the use of a specific cycle of analyst and user activities
- Each phase has unique user activities.
- Incorporating Human-Computer Interaction (HCI) Considerations
 - The demand for analysts who are capable of incorporating HCI into the systems development process keeps increasing, as companies begin to realize that the quality of systems and the quality of work life can be improved by taking a human-centered approach at the outset of a project

■ SDLC – THE SEVEN PHASES



■ PHASE 1: Identifying Problems, Opportunities & Objectives



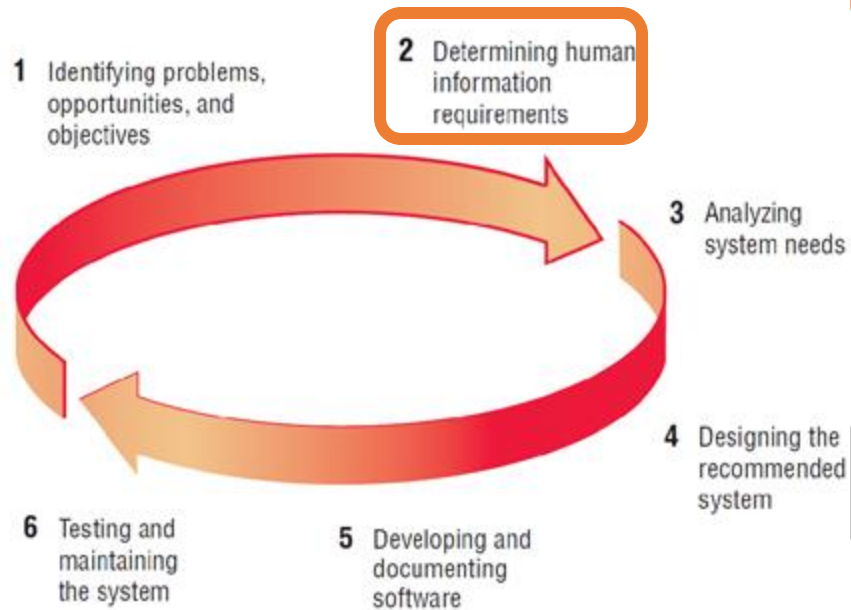
ACTIVITY:

- Interviewing user management
- Summarizing the knowledge obtained
- Estimating the scope of the project
- Documenting the results

OUTPUT: FEASIBILITY REPORT

- containing problem definition and objective summaries from which management can make a decision on whether to proceed with the proposed project (come out with your system proposal)

■ PHASE 2: Determining Human Information Requirements



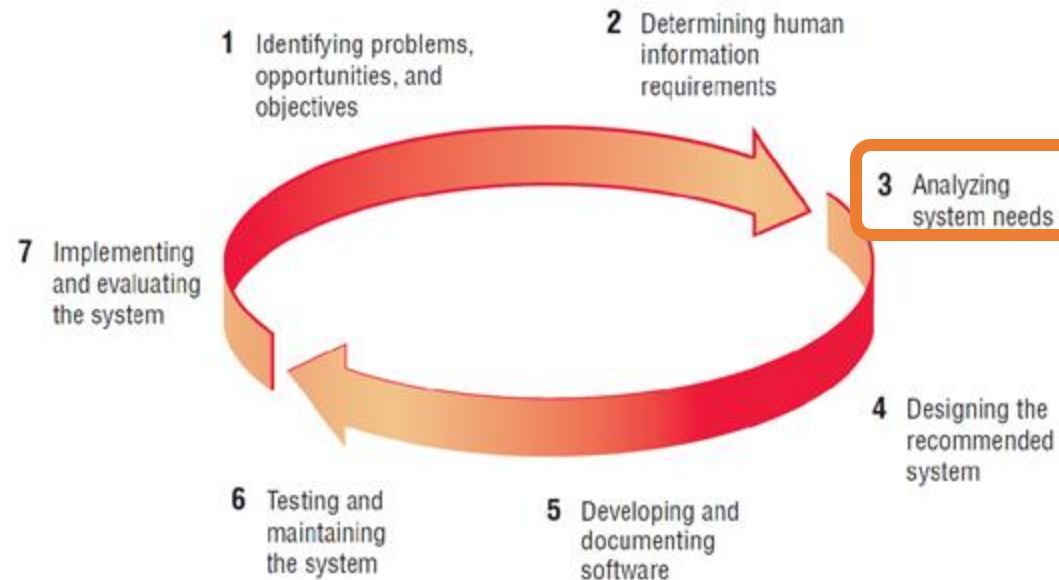
Activity:

- Interviewing
- Sampling and investigating hard data
- Questionnaires
- Observe the decision maker's behavior and environment
- Prototyping
- Learn the who, what, where, when, how, and why of the current system

Output:

- Analyst understands how users accomplish their work when interacting with a computer; and begin to know how to make the **new system more useful and usable**.
- The analyst should also know the **BUSINESS FUNCTIONS** and have complete information on the people, goals, data and procedure involved.

■ PHASE 3: Analyzing System Needs



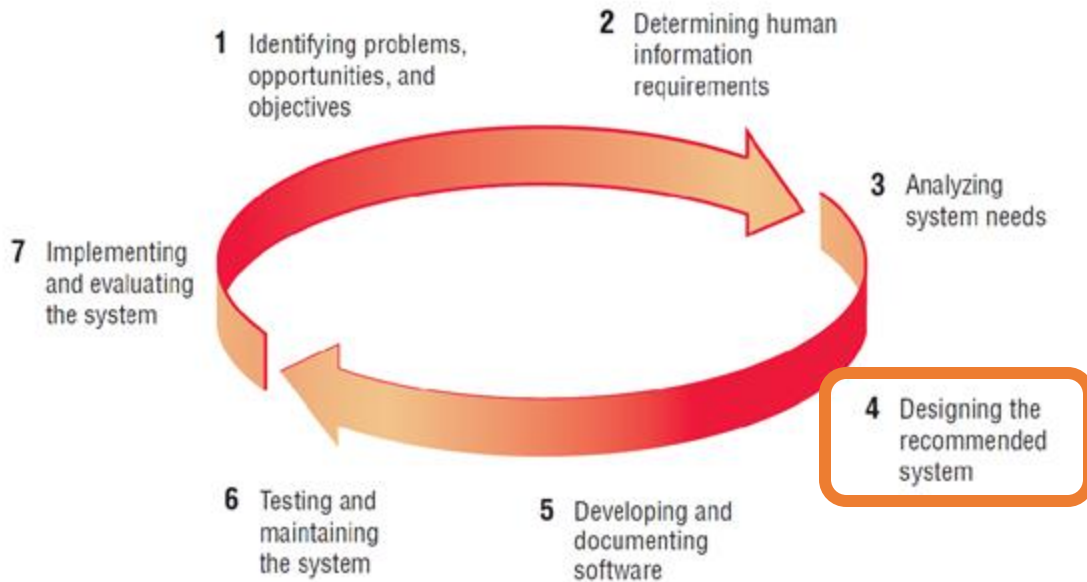
Activity:

- Create data flow diagrams
- Complete the data dictionary
- Analyze the structured decisions made
- Prepare and present the system proposal

Output:

- Recommendation on what, if anything, should be done

■ PHASE 4: Designing the Recommended System



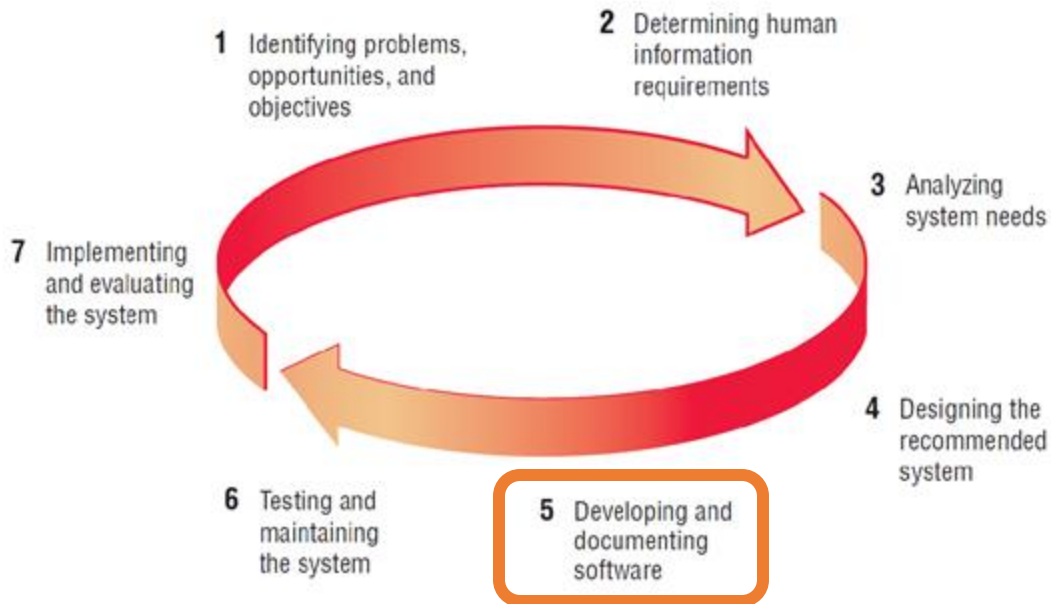
Activity:

- Design procedures for data entry
- Design the human-computer interface
- Design system controls
- Design files and/or database
- Design backup procedures

Output

- Model of the actual system

■ PHASE 5: Developing and Documenting Software



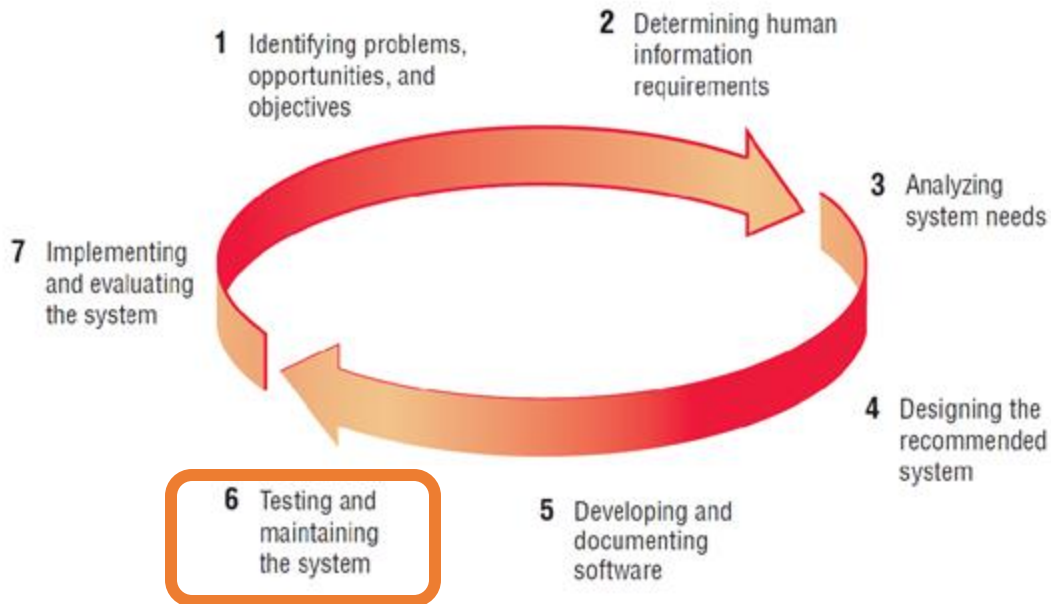
Activity:

- System analyst works with programmers to develop any original software
- Works with users to develop effective documentation
- Programmers design, code, and remove syntactical errors from computer programs
- Document software with help files, procedure manuals, and Web sites with Frequently Asked Questions

Output:

- Computer programs
- System documentation

■ PHASE 6: Testing & Maintaining the System



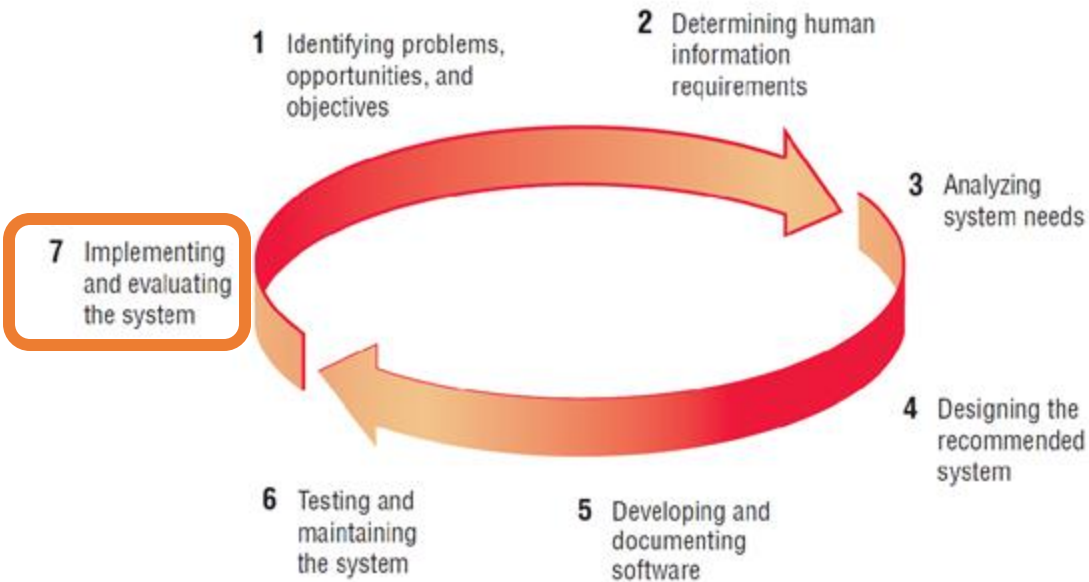
Activity:

- Test the information system
- System maintenance
- Maintenance documentation

Output:

- Problems (if any)
- Updated programs
- Documentation

■ PHASE 7: Implementing & Evaluating the System



Activity:

- Do user training
- Analyze the plan to make sure smooth conversion from old system to new system
- Review and evaluate system

Output:

- Trained personnel
- Installed system

■ SYSTEM MAINTENANCE

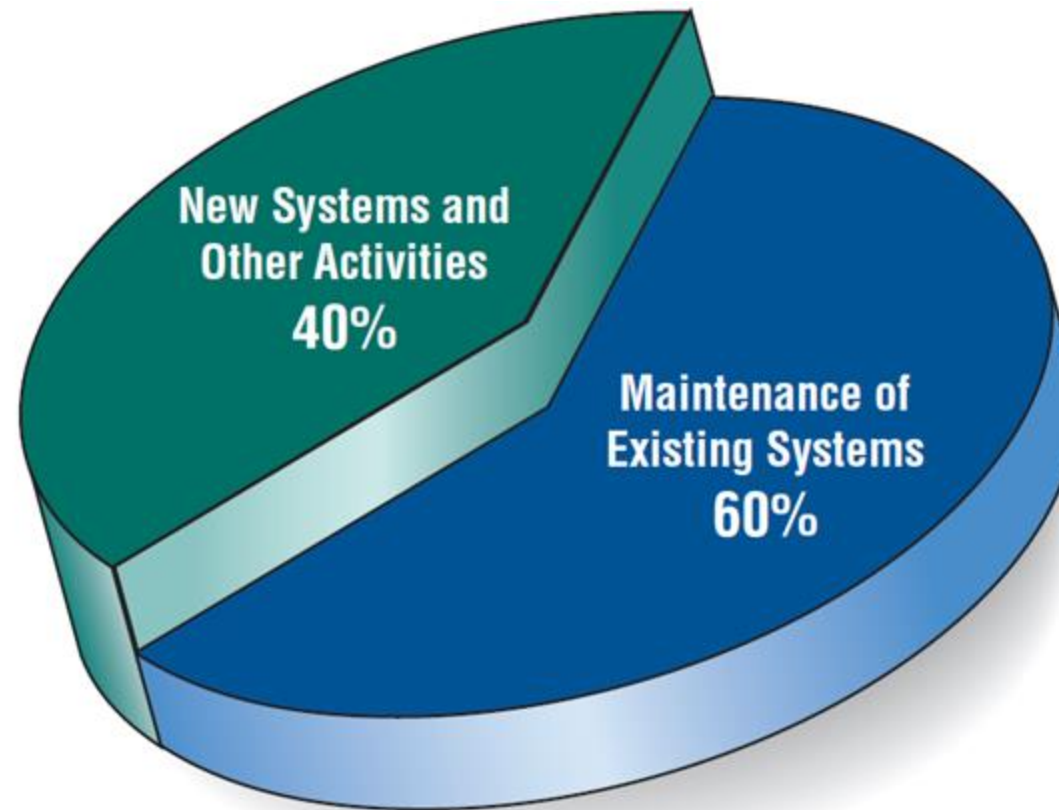


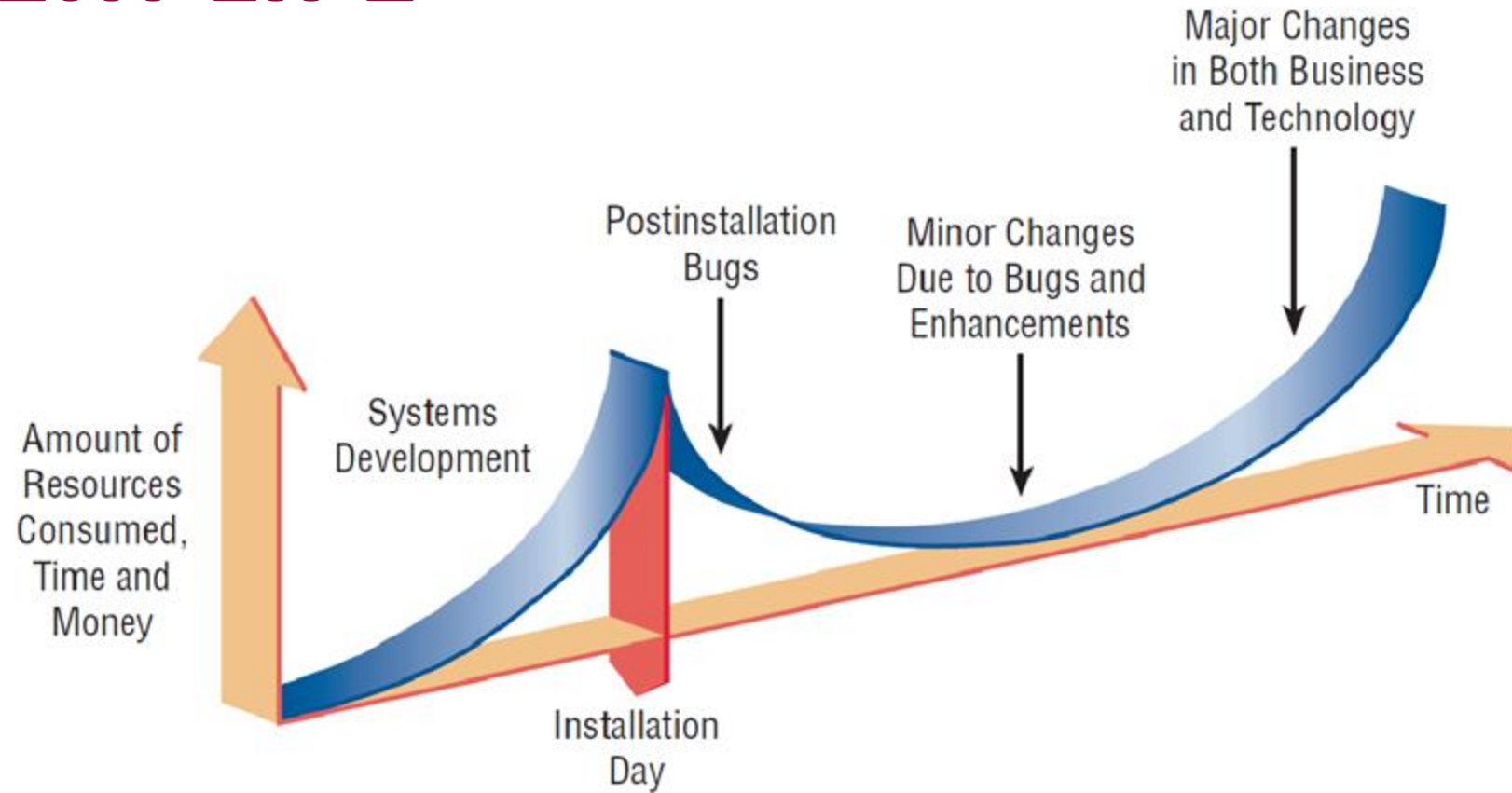
Figure 1.4 Some researchers estimate that the amount of time spent on systems maintenance may be as much as 60 percent of the total time spent on systems projects

■ THE IMPACT OF MAINTENANCE

- Maintenance is performed for TWO reasons:
 - Removing software errors, and
 - Enhancing existing software
- Over time the cost of continued maintenance will be greater than that of creating an entirely new system. At that point it becomes more feasible to perform a new systems study



■ RESOURCE CONSUMPTION OVER THE SYSTEM LIFE

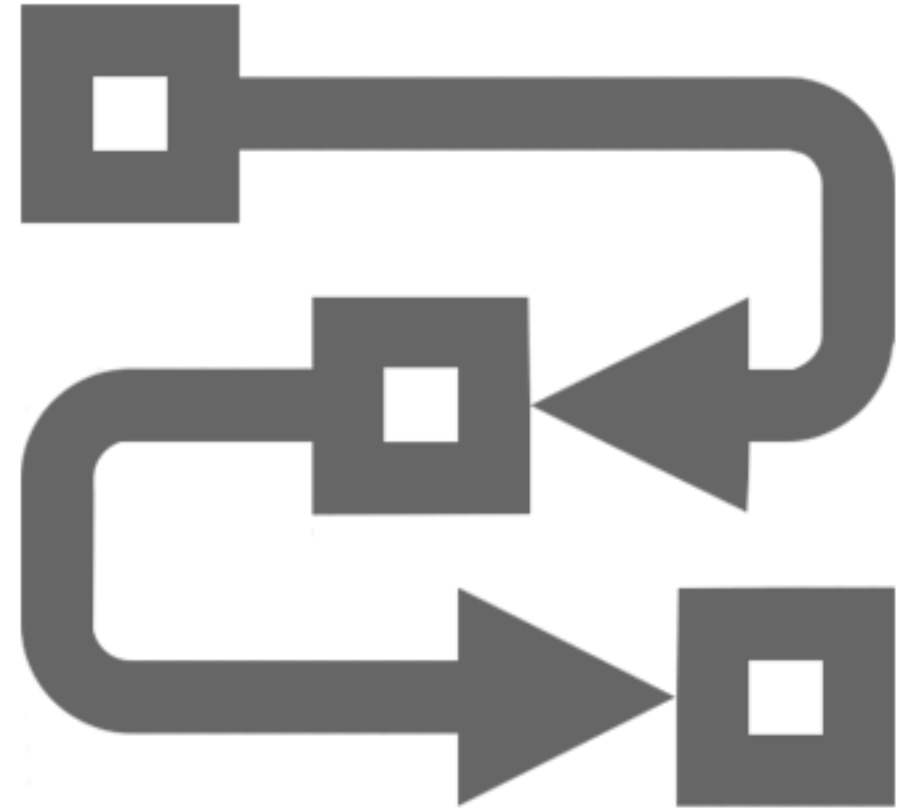


PART 3

- Types, trend and approach towards developing information system
- Summary

■ IS DEVELOPMENT METHODOLOGIES

“Comprehensive guidelines to follow for completing every activity in the systems development life cycle, including specific models, tools and techniques”



IS DEVELOPMENT METHODOLOGIES

1st formal description of Waterfall methodology

1956

1985

US Department Of Defense normalizes
WATERFALL METHODOLOGY



1986

- ▶ *No Silver Bullet* paper is published
- ▶ Spiral development model
- ▶ Ikujiro Nonaka y Hirotaka Takeuchi introduce the Scrum concept

1990

Ken Schwaber and Jeff Shuterland used what would become Scrum in the software development

1995

Scrum methodology paper is published



1996-1998

Rational Unified Process (RUP) standard was born



Extreme Programming Explained paper is released

1999

2001

The Agile Manifesto is published



Lean Software Development book is published

2003

2008

Craftmanship over Crap is included as the 5th value of Agile Manifesto

- ▶ **Software Craftmanship** movement
- ▶ **DevOps** concept became popular in a series of "devopsdays"

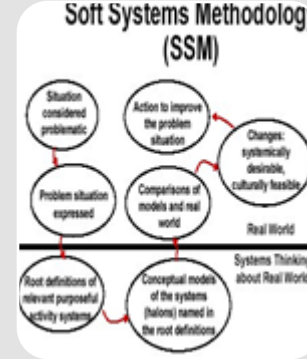
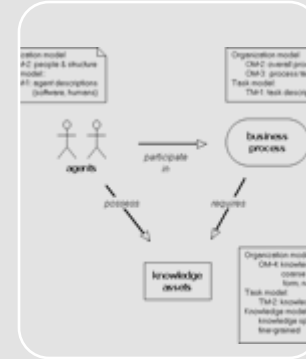


Kanban book is published

2009

2010

IS DEVELOPMENT METHODOLOGIES...



PROCESS-ORIENTED

- STRADIS Methodology - Gane & Sarson's Methodology
- Yourdon System Method (YSM)
- Jackson System Development (JSD)

BLENDED

- Structured System Analysis & Design Method (SSADM)

OBJECT-ORIENTED

- Booch Method
- OOSE
- OMT
- Unified Approach Methodology
- Rational Unified Process

RAPID

- James Martin's RAD
- Extreme Programming (XP)
- Web IS development methodology (WISDM)

PEOPLE-ORIENTED

- CommonKADS
- ETHICS
- KADS

ORGANIZATIONAL-ORIENTED

- Soft-System Methodology (SSM)
- IS work and Analysis of Changes (ISAC)
- Process Innovation (PI)
- Project in controlled environments (PRINCE)

FRAMEWORK

- Multiview
- Strategic options development and analysis (SODA)
- Capability Maturity Model (CMM)

SDLC APPROACHES

- Traditional Systems Development Life Cycle (discussed in previous Part 2)
- Object-Oriented Systems Analysis and Design
- CASE Systems Development Life Cycle



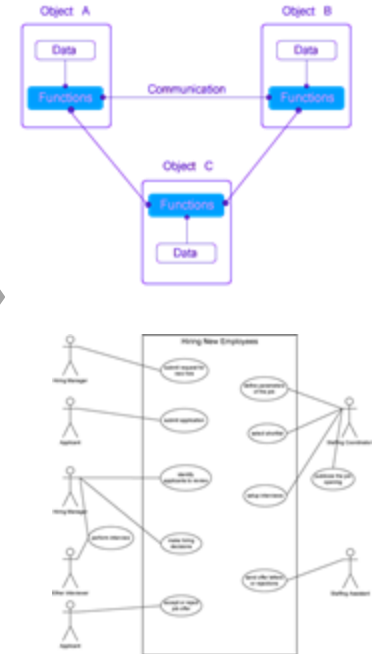
■ 2 GENERAL APPROACHES

Structured Approach



Traditional Approach

Object-Oriented Approach



“All system developers should be familiar with two very general approaches (STRUCTURED & OO) to system development, because they form the basis of virtually all methodologies”

■ OBJECT ORIENTED SYSTEM ANALYSIS & DESIGN

- Alternate approach to the structured approach of the SDLC that is intended to facilitate the development of systems that must change rapidly in response to dynamic business environments
- Analysis is performed on a small part of the system followed by design and implementation. The cycle repeats with analysis, design and implementation of the next part and this repeats until the project is complete

■ STRUCTURED vs OBJECT ORIENTED

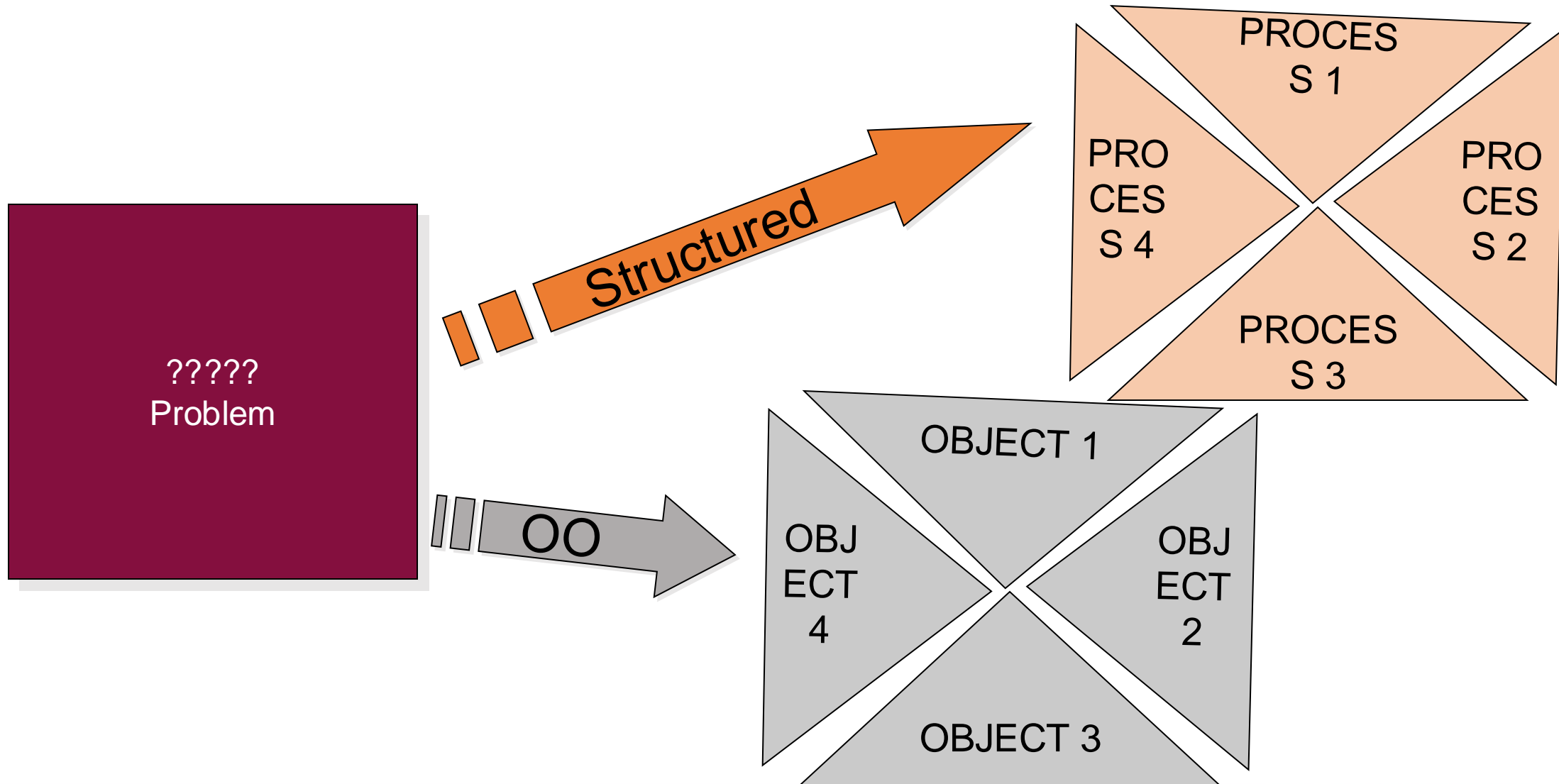
Structured Methodologies

- 1. STRADIS**
- 2. SSADM**
- 3. JSD**
- 4. YSM**
- 5. More...**

O-Oriented Methodologies

- 1. Booch**
- 2. OMT**
- 3. OOSE**
- 4. Unified Approach**
- 5. Rational Unified Process**
- 6. More...**

■ STRUCTURED vs OBJECT ORIENTED

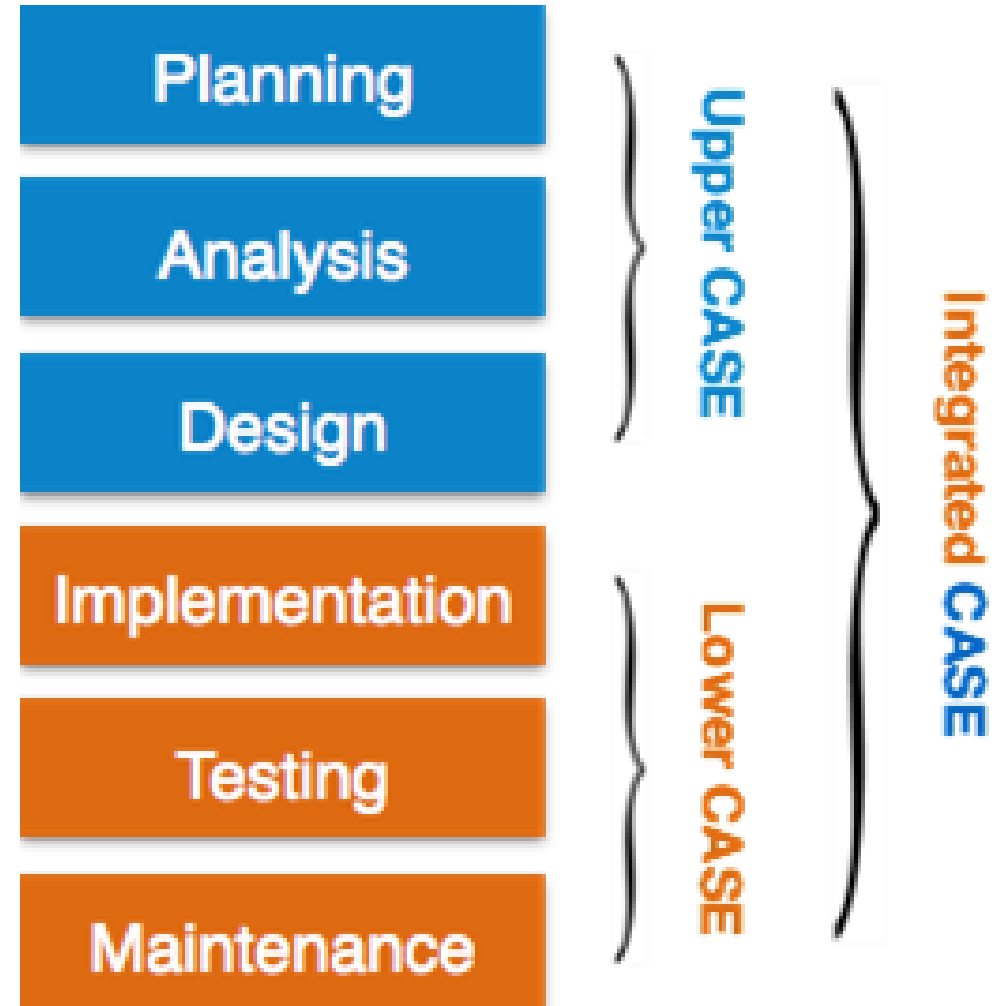


■ CASE TOOLS

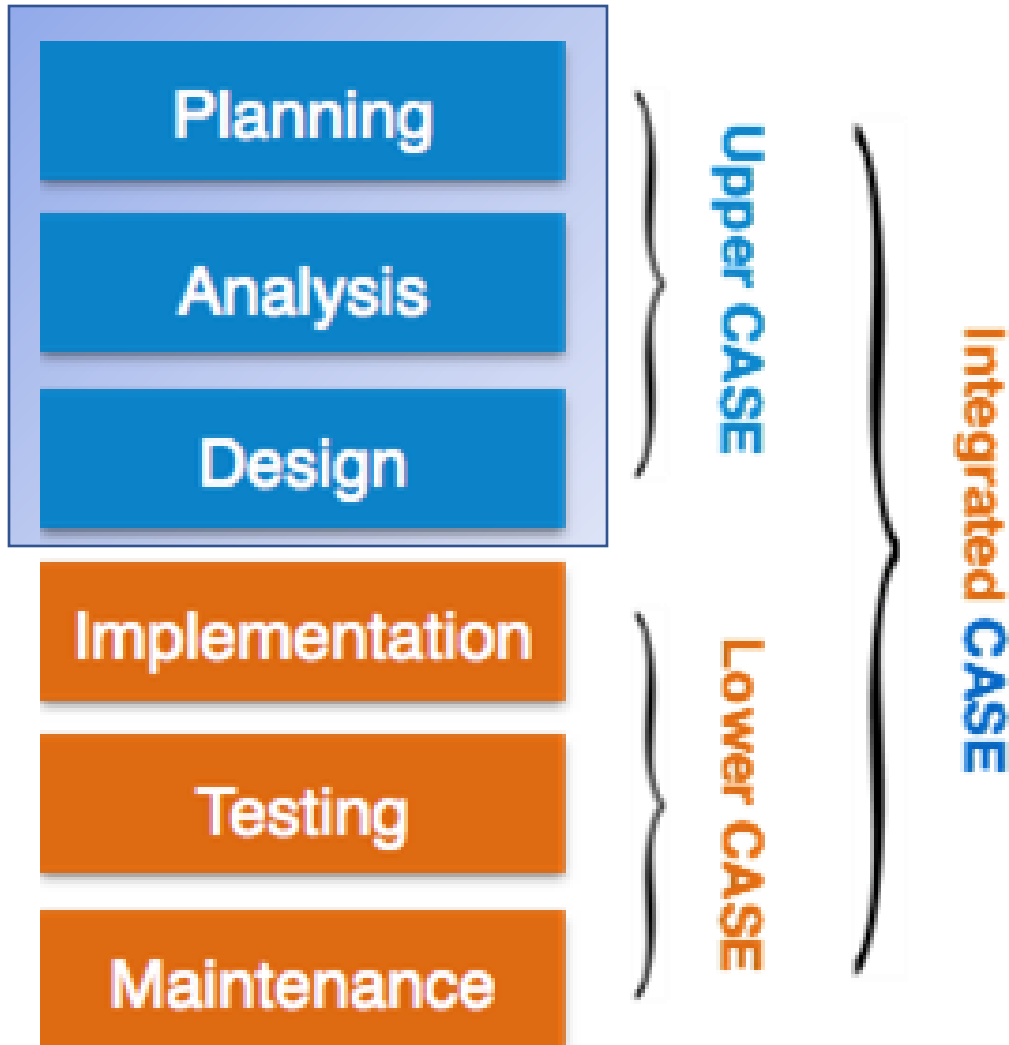
- CASE tools are productivity tools for systems analysts that have been created explicitly to improve their routine work through the use of automated support
- Reasons for using CASE tools
 - Increasing Analyst Productivity
 - Improving Analyst-User Communication
 - Integrating Life Cycle Activities
 - Accurately Assessing Maintenance Changes

■ CASE TOOLS CLASSIFICATION

- Upper CASE tools perform analysis and design (mainly for analyst & designers)
- Lower CASE tools generate programs from CASE design (use by programmers)
- Integrated CASE tools perform both upper and lower CASE functions

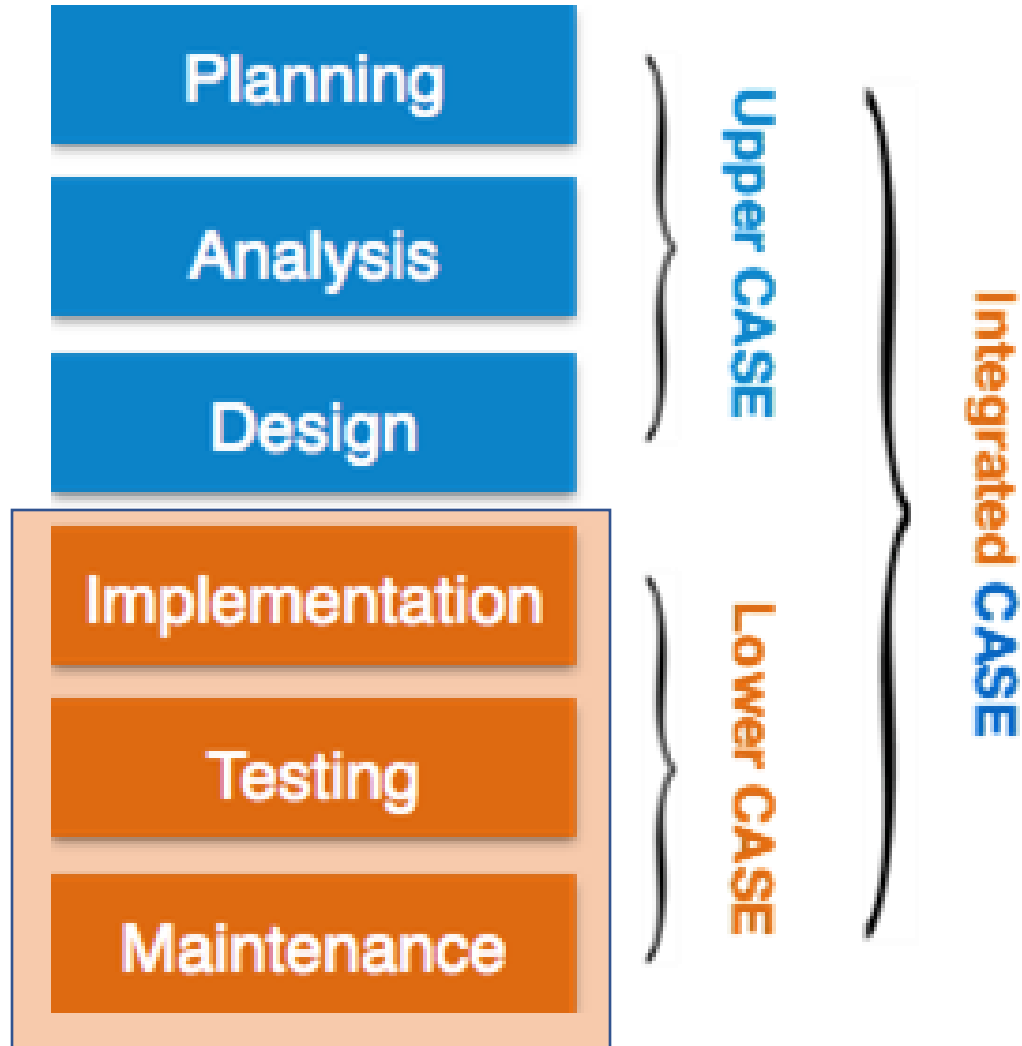


■ UPPER CASE TOOLS



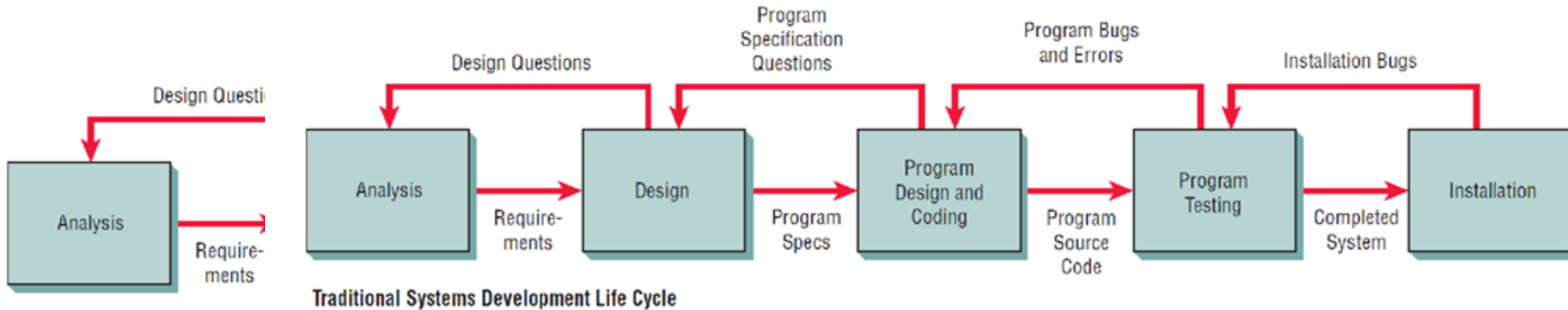
- Create and modify the system design
- Help in modeling organizational requirements and defining system boundaries
- Can also support prototyping of screen and report designs

■ LOWER CASE TOOLS

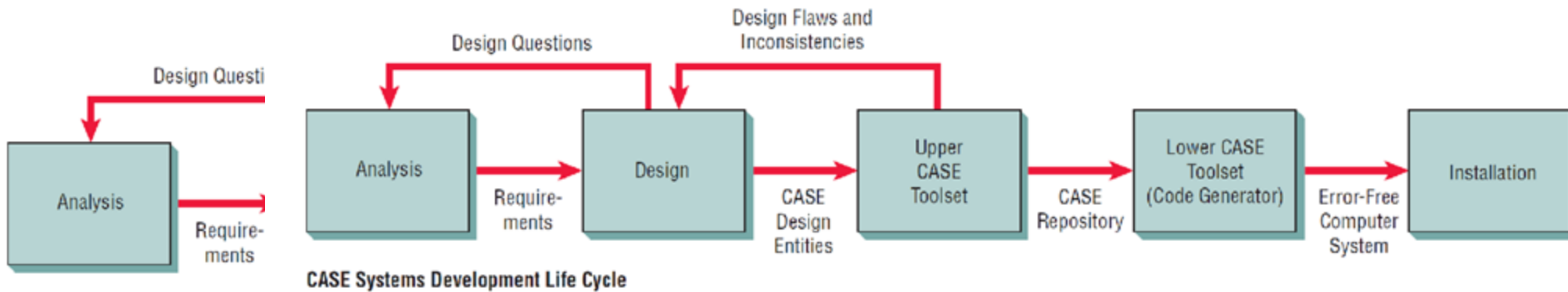


- Lower CASE tools generate computer source code from the CASE design
- Source code is usually generated in several languages

■ TRADITIONAL vs CASE SDLC



Traditional Systems Developm

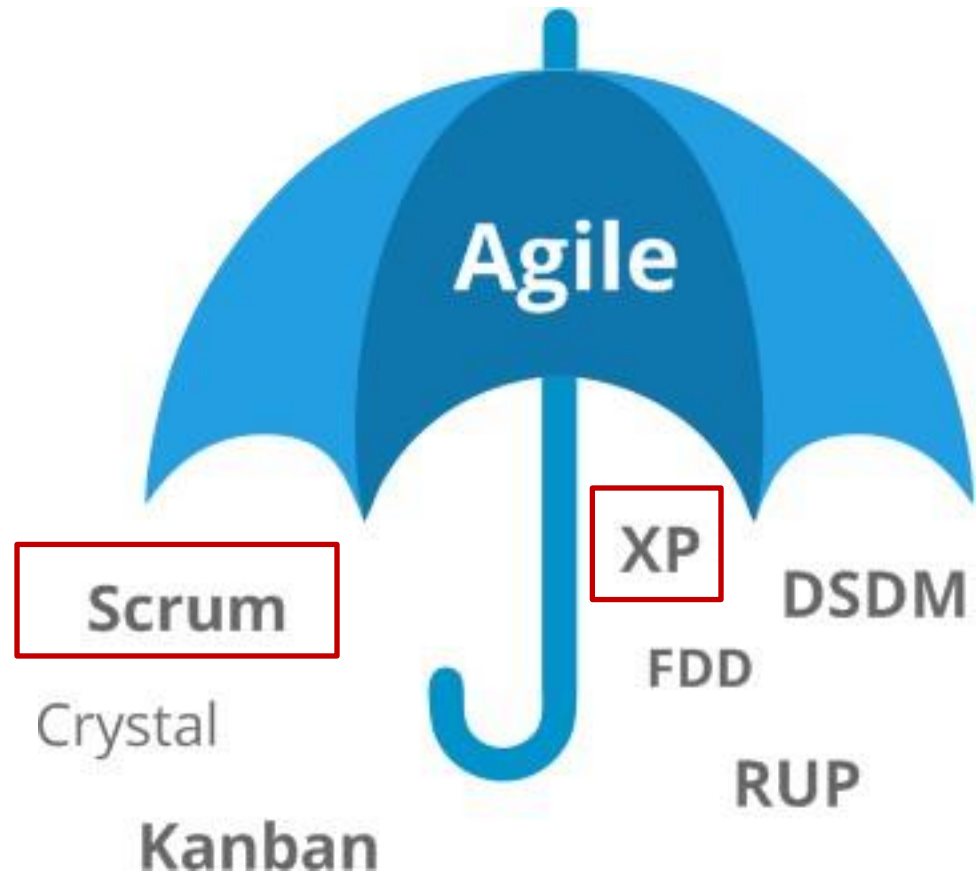


CASE Systems Development Life Cycle

ALTERNATE SDLC APPROACHES



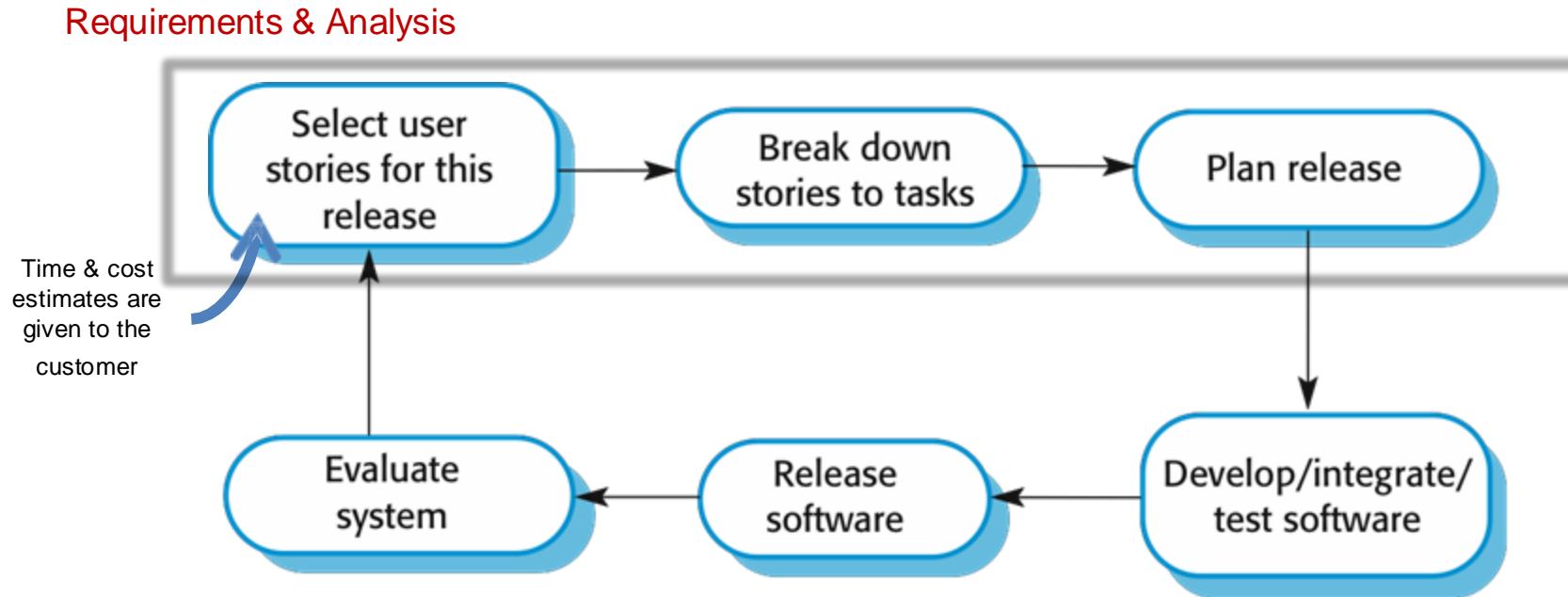
Agile Development Techniques



Extreme Programming

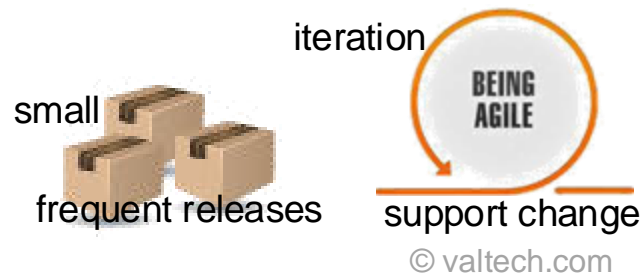
- A very influential agile method, developed in the late 1990s, that introduced a range of agile development techniques
- Extreme Programming (XP) takes an 'extreme' approach to iterative development:
 - **New versions** may be built several times per day
 - **Increments** are delivered to customers every 2 weeks
 - **All tests** must be run for every build and the build is only accepted if tests run successfully

The Extreme Programming Release Cycle



XP and Agile Principles

- **Incremental development** is supported through small, frequent system releases
- **Customer involvement** means full-time customer engagement with the team
- **People not process** through pair programming, collective ownership and a process that avoids long working hours
- **Change** supported through regular system releases
- **Maintaining simplicity** through constant refactoring of code



2021

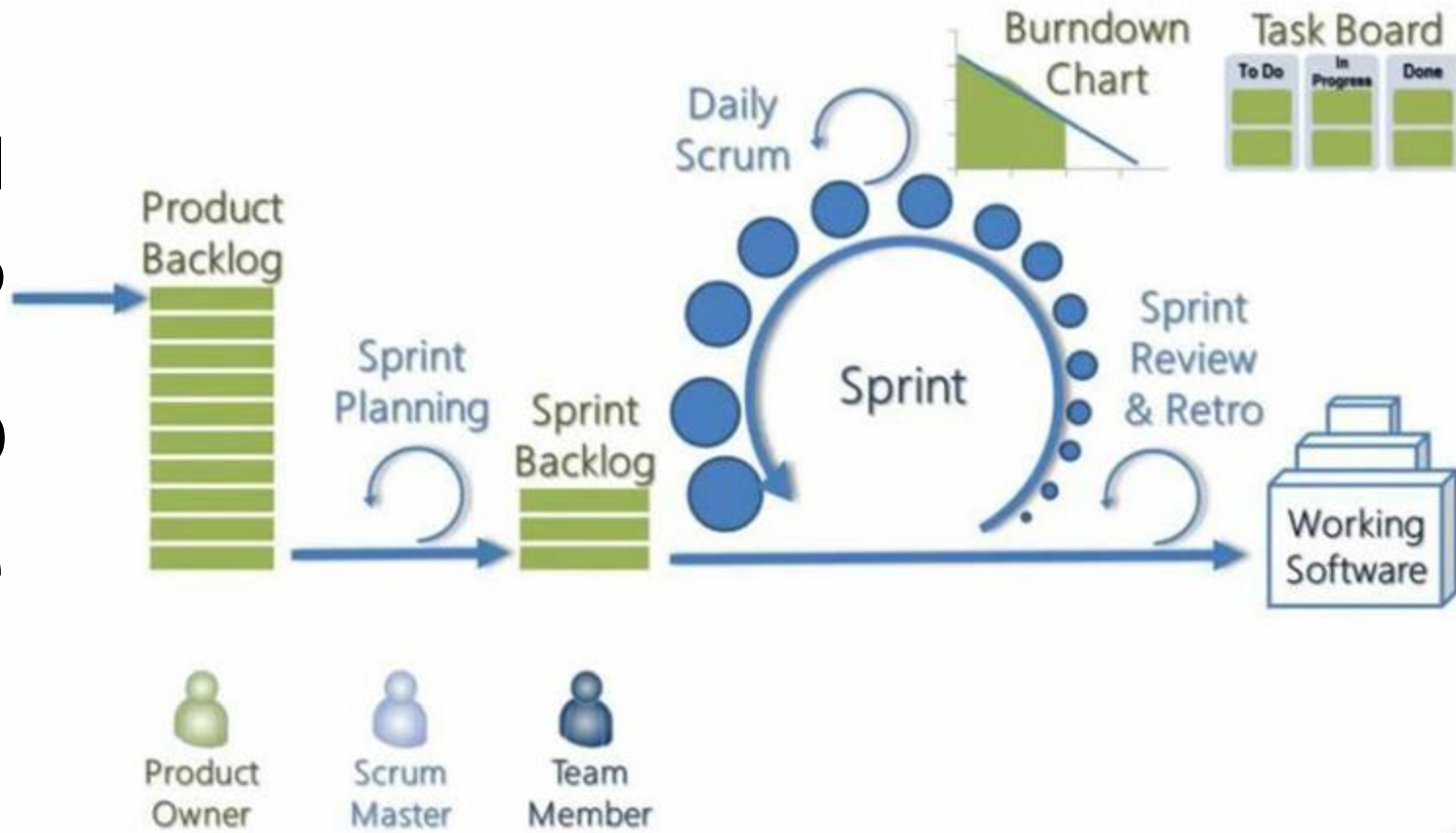
people
NOT
process



Scrum

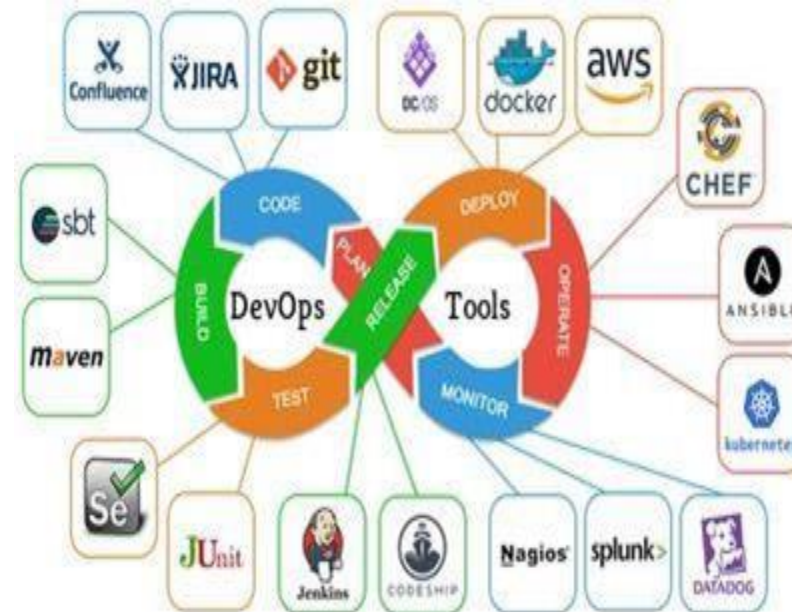
- Scrum is an agile method that focuses on **managing iterative development** rather than specific agile practices
- There are **three phases** in Scrum:
 - The **initial phase** is an outline planning phase where the team establishes the general objectives for the project and design the software architecture
 - This is followed by **a series of sprint cycles**, where each cycle develops an increment of the system
 - The project closure phase **wraps up the project**, completes required documentation such as system help frames and user manuals and assesses the lessons learned from the project

The Scrum Process

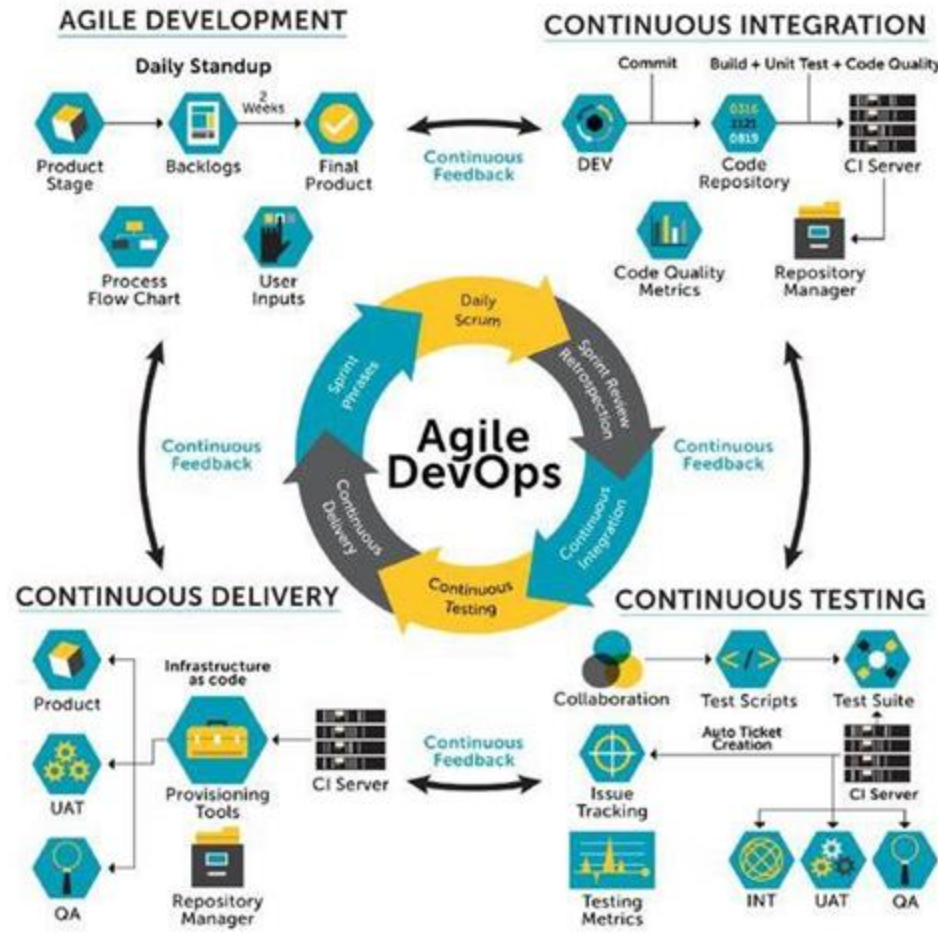


DevOps and Tools

- DevOps is a set of practices that emphasizes the **collaboration and communication** of both software developers and other information technology (IT) professionals, while **automating the process** of software delivery and infrastructure changes, its implementation can include the definition of the **series of tools** used at various stages of the lifecycle
- There is no one product that can be considered a single DevOps tool



Agile DevOps...



<https://www.educba.com/agile-devops/>

- Agile along with DevOps has a **collaborative working style**, irrespective of the method implemented
- Both the methodologies rely on **continual feedback and routine updates** about the work progress from internal and external stakeholders
- Both Agile and DevOps focus on **developing the product at a fast pace**, by keeping smaller teams and using a risk-free approach
- Both methods **adapt to the business requirements** and continually improve the products to fulfill customer expectations

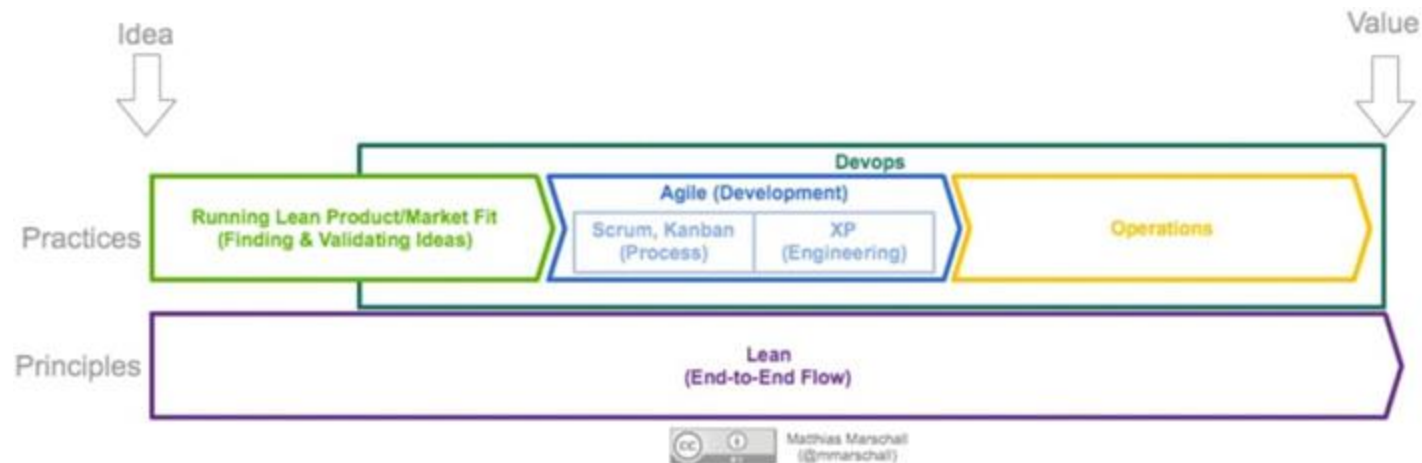
Agile DevOps

- Agile DevOps is an **extended agile methodology** for product development where DevOps is a succession to agile and not a replacement
- In a nutshell, Agile works towards software development by making **continuous alterations**, adapting and developing products as per customer expectations
- While DevOps through **automated processes** and **bug detection** at an early stage of software development focusses primarily on deploying products of high quality

Source: <https://www.educba.com/agile-devops/>

DevOps: Agile & Lean to Operations Work

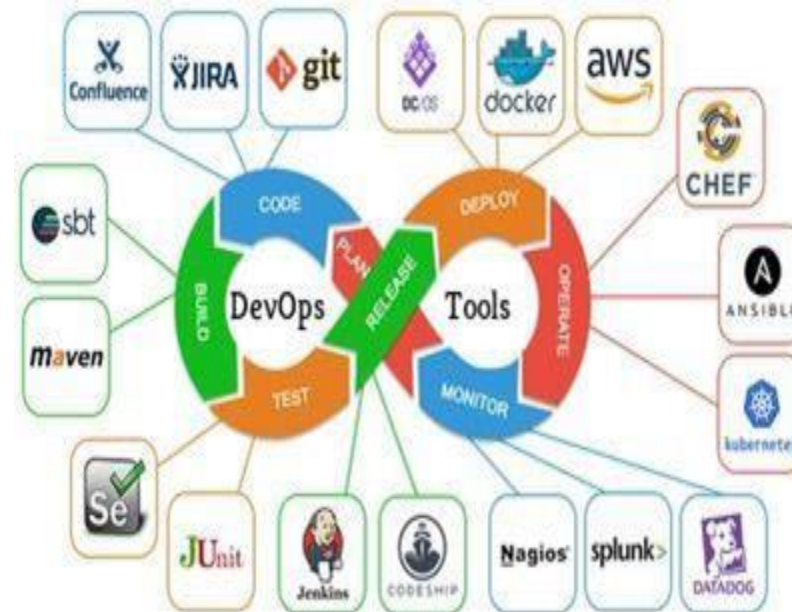
- DevOps: a practice of **operations** and **developments** engineers participating together in the entire service lifecycle from design through development process to production support



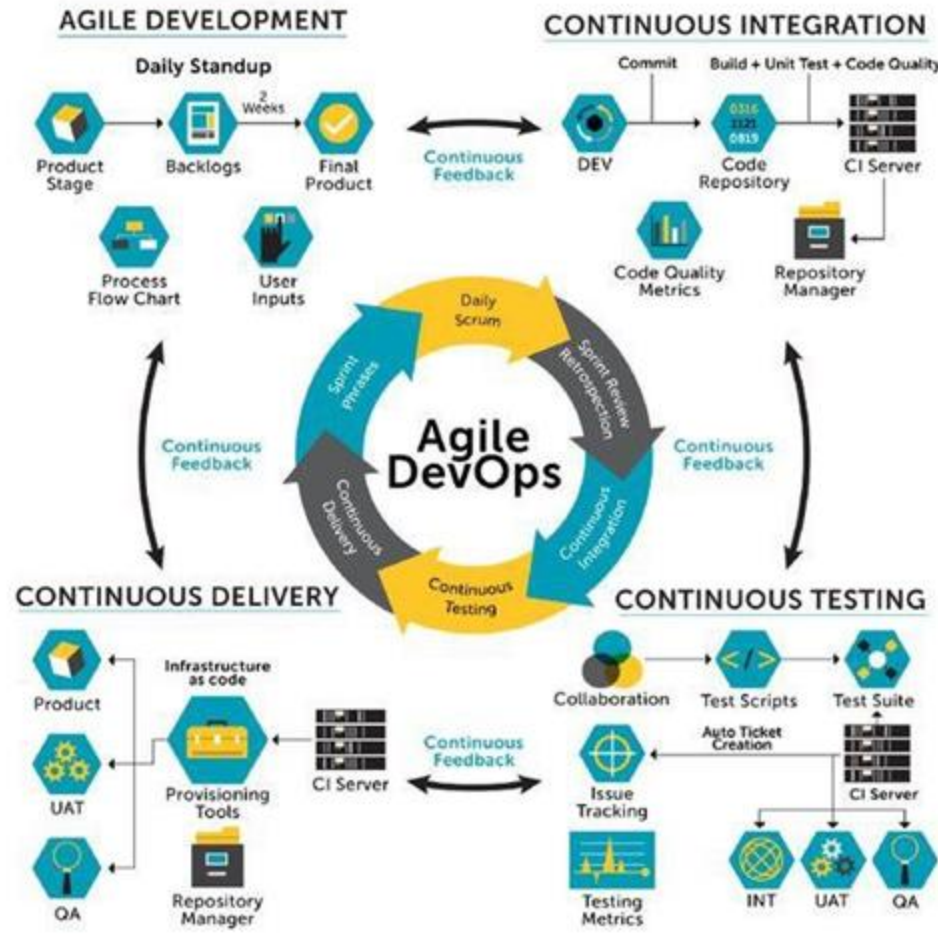
Source: <https://theagileadmin.com/what-is-devops/>

DevOps and Tools

- DevOps is a set of practices that emphasizes the **collaboration and communication** of both software developers and other information technology (IT) professionals, while **automating the process** of software delivery and infrastructure changes, its implementation can include the definition of the **series of tools** used at various stages of the lifecycle
- There is no one product that can be considered a single DevOps tool



Agile DevOps...



<https://www.educba.com/agile-devops/>

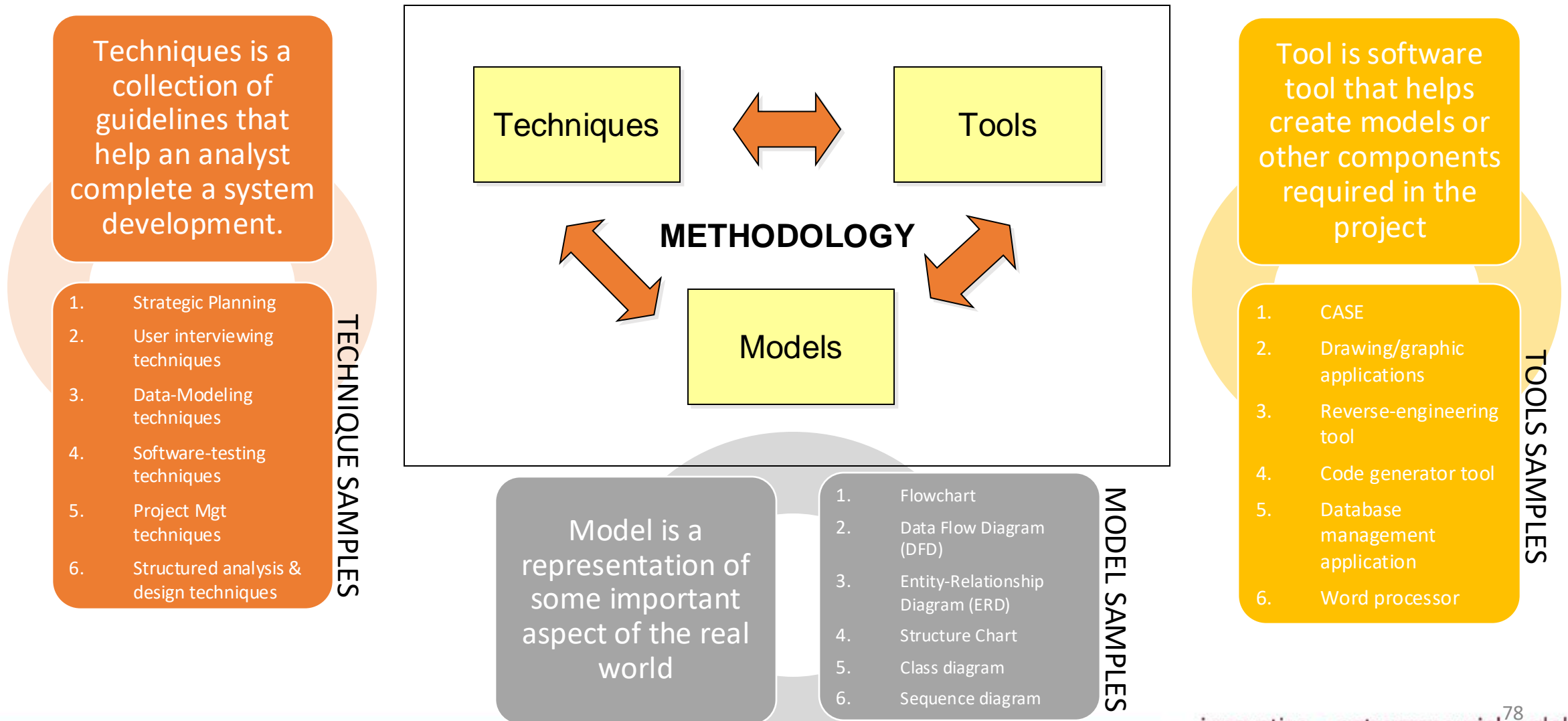
- Agile along with DevOps has a **collaborative working style**, irrespective of the method implemented
- Both the methodologies rely on **continual feedback and routine updates** about the work progress from internal and external stakeholders
- Both Agile and DevOps focus on **developing the product at a fast pace**, by keeping smaller teams and using a risk-free approach
- Both methods **adapt to the business requirements** and continually improve the products to fulfill customer expectations

Agile DevOps

- Agile DevOps is an **extended agile methodology** for product development where DevOps is a succession to agile and not a replacement
- In a nutshell, Agile works towards software development by making **continuous alterations**, adapting and developing products as per customer expectations
- While DevOps through **automated processes** and **bug detection** at an early stage of software development focusses primarily on deploying products of high quality

Source: <https://www.educba.com/agile-devops/>

■ MAIN COMPONENTS OF METHODOLOGY

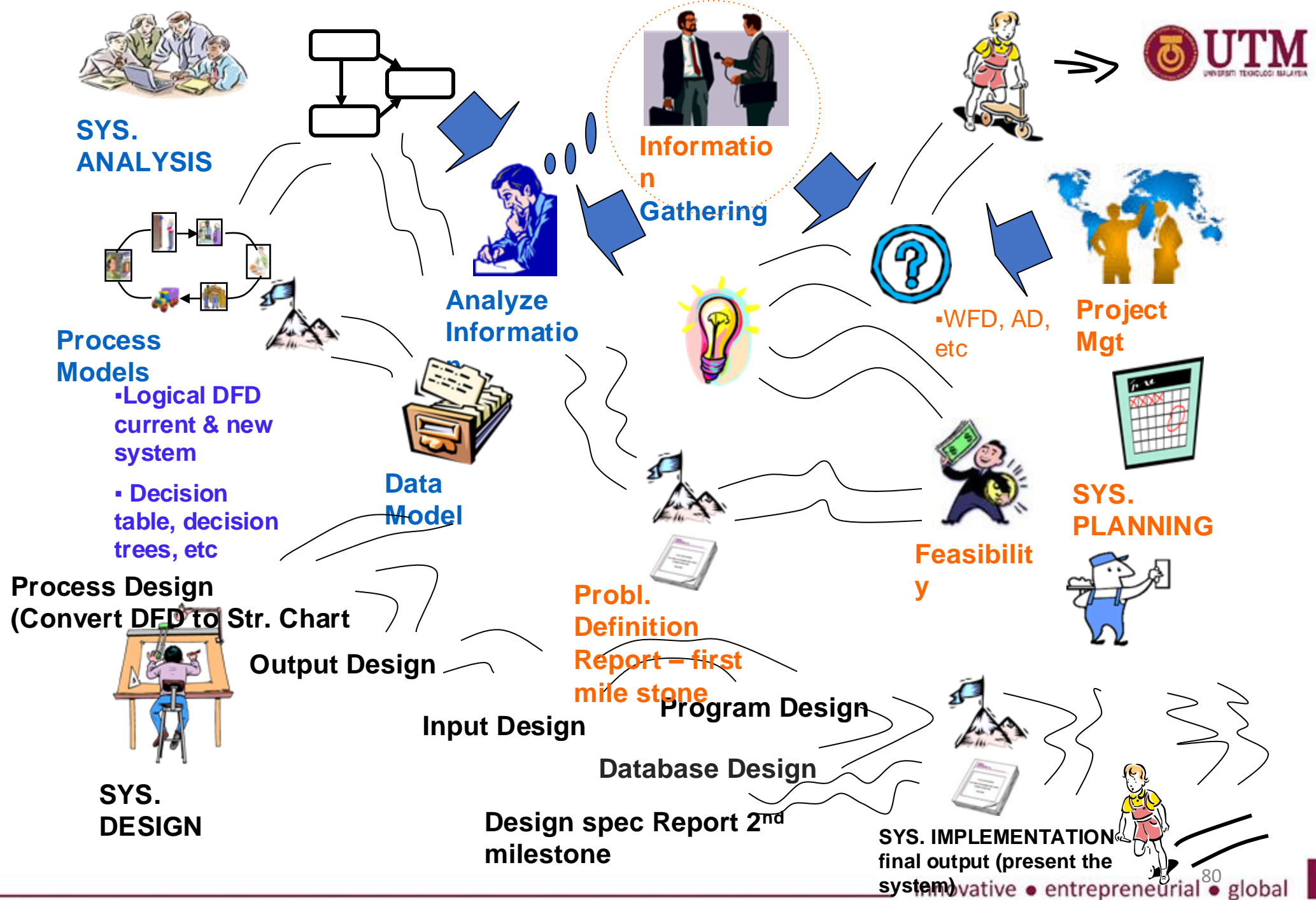


TOPIC SUMMARY



innovative • entrepreneurial • global

OVERALL PICTURE OF SYSTEM ANALYSIS & DESIGN



■ REFERENCES

- Kendall, K.E. & Kendall, J.E., 2019. *System Analysis and Design*. 10th Ed. Essex: Pearson.



UTM
UNIVERSITI TEKNOLOGI MALAYSIA



univteknologimalaysia



utm_my



utmofficial

Thank You

update: August 2019 (sharinhh)

www.utm.my

innovative • entrepreneurial • global