CHAPTER II

THEORIES AND LETERATURE REVIEW

2.1 Information management

Every business needs to deal with data and information. Data is raw material includes facts, tables, figures, etc. which is meaningless. Information is data that processed by arranging, formatting, analysing, etc. which is used for work by people in the company. Some information can be transformed to knowledge that provide understanding about something or situations which is very important in strategic level.

In order to grow and gain competitive advantages in high competitive environment, the better management approaches of data, information, and knowledge are required to make the company survives.

2.1.1 The role of information in organisations

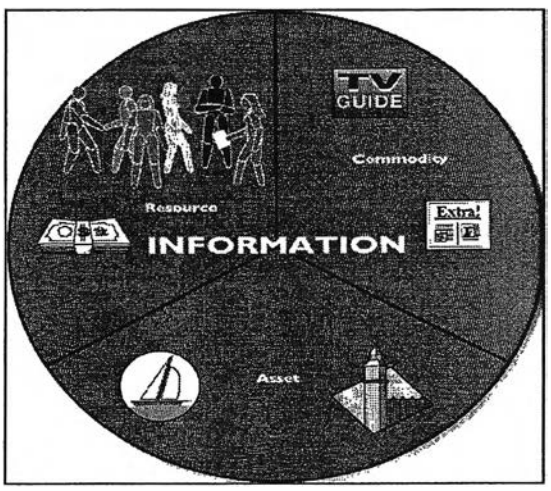


Figure 2.1: Roles of information
(Gordon and Gordon, 1999: 8)

From the figure, information in the organisation is used for 3 major types.

- **Resources**

Information is used as input of processes that produce outputs. It is information about budget, labour, cost, time which are deal directly with productivity of the company. It usually is used for planning.

- **Asset**

  "Information can serve as an asset, which is the property of a person or an organisation that contributes to a company's output." (Gordon and Gordon, 1999: 9)

  So the management should view some information as asset and invest on them to gain competitive advantages over other companies.

- **Commodity**

  Some information can be sold for gaining profit.

2.1.2 The role of Information Technology

"The widespread availability of affordable computer technology has dramatically changed the way people acquire, process, store, retrieve, transmit, communicate, and use information." (Gordon and Gordon, 1999: 9)

Information technology is a useful tool for managing information which covers individual, group, and enterprise level of work. The general components of Information Technology include hardware, software, database, and data communication which can encourage information management by;

- **Hardware**
  - Input data from data sources such as key-in by keyboard, scan from scanner, read symbols from bar cord reader, QR code reader, and RFID.
  - Process and transform data by calculating, rearranging, formatting, etc.
  - Store data or information for shared access and retrieve for processing.
  - Output data or information from printers, display screen, etc.
- **Software**
  - Platforms to control all hardware equipments and process data.
  - Specific software that can access data directly and make some changes.
- **Database**; encourage easier add / edit / delete / access data in database
- **Data communication**; techniques that mainly based on networking to transfer data or information from place to place.

2.1.3 Managing Information with Information Systems

"An information system (IS) combines information technology with data, procedures for processing data, and people who collect and use the data." (Gordon and Gordon, 1999: 11)

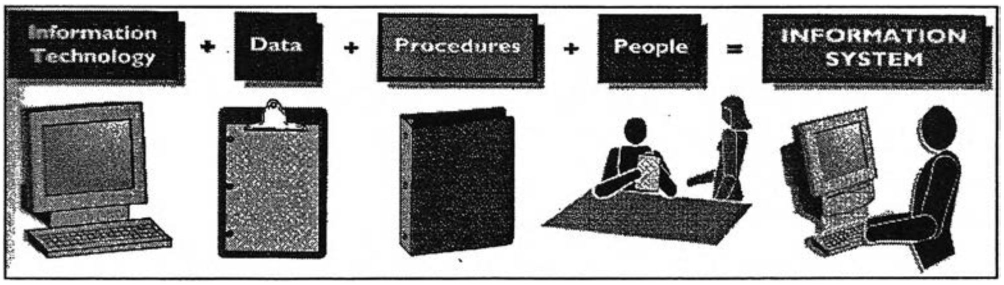


Figure 2.2: Components of Information System
(Gordon and Gordon, 1999: 11)

From the figure, the most important parts of the information system are procedures that control all work in the flows and people who use and deal with the system. Professional people and good procedures can bring about efficient information system that can fully support the company work. In order to develop an efficient information system, developers have to make sure that they have 'good enough' technology, data, and especially procedures and people.

2.1.4 Information management in this era

In this high competitive environment, information management and information system can not only used for facilitate routine works.

- **Globalisation**

  The internet and World Wide Web make the world narrower, so the company can perform real time communication with other company in another side of the world. The information system can be plugged-in with the network platform to support globalisation. However, the wider that the system can cover, the more reliability and stability of the system are needed.

- **Quality and performance improvement**

  The number of competitors is increased rapidly. Competing on price is not such an effective way to do. Total Quality Management (TQM) is introduced to improve performance of the company products and services.

"TQM programs emphasize responding to customers' needs as a top priority. By giving workers more responsibility for making decisions, these programs change the way employees perform work. TQM programs foster continuous improvement in both an organization's product and the processes for creating it." (Gordon and Gordon, 1999: 16)

- **Business process redesign**

"Reengineering refers to rethinking, and redesigning one or more of an organisation's business systems – such as accounts receivable, purchasing, or product development – and its related jobs. Business process redesign identifies core processes and looks for better ways to do them" (Gordon and Gordon, 1999: 17)

Business process reengineering and redesign are performed for improving performance of the current work processes by eliminating improper processes, shorten lead-time, increase quality of outputs, etc. the method of reengineering always deal with computers and information systems which can provide automatic work flows for the company.

- **Improve individual capabilities and productivity**

Knowledge workers who do special tasks that required specific background of knowledge and skills such as engineers, accountants, programmers are organised as company assets that needed to be kept in the company. Information that required for doing tasks has to available as they need to encourage them to function more effectively. So the information system is used for support those people to manage and access information.

2.2 SDLC (Software/System Development Life Cycle)

"The SDLC describes activities and functions that all systems developers perform, regardless of which approach they use." (Shelly, Cashman, and Rosenblatt, 2003: 23)

The SDLC is divided into 5 phases include;

2.2.1 Systems planning

"The systems planning phase usually begins with a formal request to the IT department, called a **systems request** that describes problems or desire changes in an information system or a business process." (Shelly et al., 2003: 24)

The systems planning for IT projects development sometimes covers business planning because strategies of the company and IT infrastructure must match together. In order to do this, IT people must communicate with top management for develop IT plan that get along with the strategies, and adjust strategies to open for IT implementation. There are many sources of system requirements such as top management, planning teams, head of departments, and IT themselves.

Major requests generally lead to new infrastructure or big changes in current systems. And minor requests are just add-ons features.

"The purpose of this phase is to perform a preliminary investigation to identify nature and scope of the business opportunity and problem. The preliminary investigation is a critical step because the outcome will affect the entire development process." (Shelly et al., 2003: 24)

The sample techniques of the Preliminary Investigation include;

- **Defining problems and opportunities** by studying SWOT Analysis, Vision / Mission of the company.
- **Defining the project scope and constrains**
- **Perform fact-finding** by studying the company's organisation chart, cross-department linkages, internal and external documents, and interview.
- **Evaluate feasibility** in many fields such as operational, technical, economy, and schedule.
- **Estimate project development time and cost**

2.2.2 Systems analysis

"The purpose of the systems analysis phase is to build a logical model of the new system. The first step is requirements modelling, where document that contains what the new system must do" (Shelly et al., 2003: 24)

Major activity in this phase is continuing investigation which had started since previous phase – system planning. The fact-finding techniques in this phase such as interview, survey, observation, etc. mainly lead to clearer view of the current system. The outputs of analysis include enterprise models, data / process models, object models.

2.2.3 Systems design

"The purpose of the systems design phase is to create a blueprint that will satisfy all documented requirements for the system. At this stage, you design the user interface and identify all necessary outputs, inputs, and processes." (Shelly et al., 2003: 24)

In this phase, resources are enough for determining application architecture which will specify format of designs and programming code. The **systems design specification** is the most important documents that managers and users must study it carefully to make sure that the design will match with their work behaviours.

2.2.4 Systems implementation

System implementation means the new system is constructed. The system will be written, testes, and documented, and installed. The system is not only ready to use, but also provide training and systems evaluation to confirm that the system operates properly within expectations.

2.2.5 Systems operation and support

"During the systems operation and support phase, the IT staff maintains and enhances the system. Maintenance changes correct errors and adapt to changes in the environment such as new tax rates. Enhancements provide new features and benefits." (Shelly et al., 2003: 25)

A well-designed system must be reliable, easy to maintain, scalable, and able to expand to meet new business requirements.

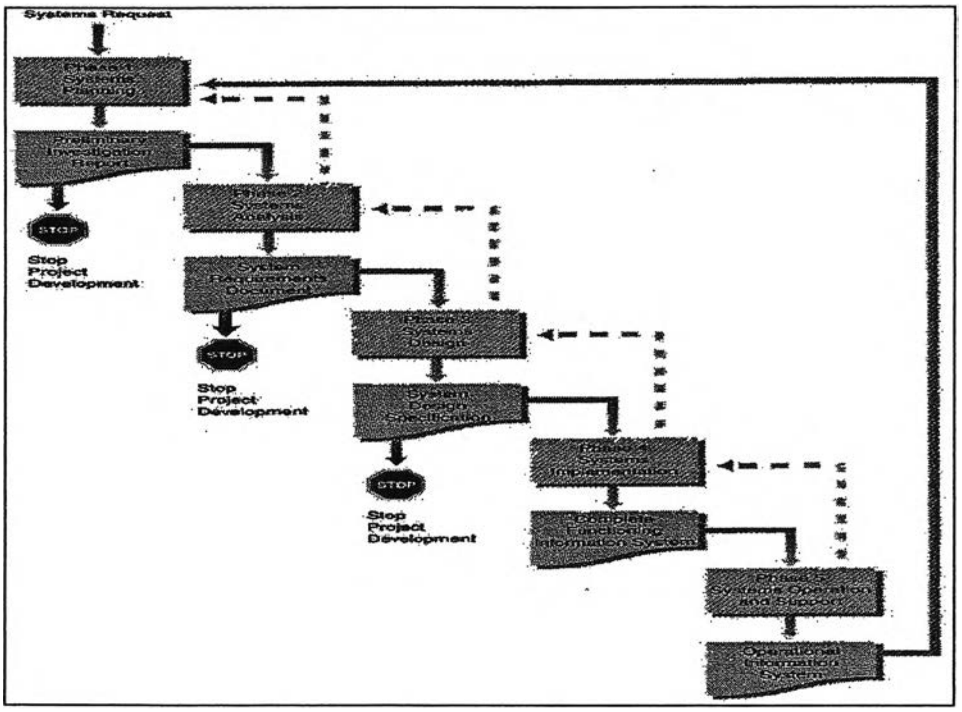


Figure 2.3: The waterfall model SDLC
(Shelly et al., 2003: 23)

2.3 System Analysis and Design

"Systems analysis and design, as performed by systems analysts, seeks to analyze systematically the data input, data flow, and information output within the context of a particular business." (Kendall and Kendall, 1992: 3)

Systems analysis and design is used for analysing, designing, developing, and implementing the improvements of businesses' functions. It is an efficient approach for information system development.

2.3.1 The need for System Analysis and Design

"If a system is installed without proper planning, it leads to great dissatisfaction with the system, and the system frequently falls into disuse. Systems analysis and design lends structure to the costly endeavour of analyzing and designing information systems, which would otherwise be done in a haphazard way. Systems analysis and design is a series of processes systematically undertaken to improve a business through the use of computerized information systems." (Kendall and Kendall, 1992: 3)

2.3.2 Systems Development Methodologies

"There are many different systems development methodologies, and each one is unique because of its emphasis on processes versus data and the order and focus it places on each SDLC phase." (Dennis and Wixom, 2003: 8)

The methodologies that categorized by sequencing of the SDLC phases and the amount of time and effort can be divided into 3 categories.

2.3.2.1 Structured Design

"Structured design methodologies adopt a formal step-by-step approach to the SDLC that moves logically from one phase to the next." (Dennis and Wixom, 2003: 9)

- **Waterfall Development**

"The key deliverables for each phase are typically produced on paper and are presented to the project sponsor for approval as the project moves from phase to phase." (Dennis and Wixom, 2003: 9)

The name 'waterfall' comes from its behaviours of moving down from phase to phase like waterfall.

Pros – it can identify the system requirements and freeze them before programming phase starts which it minimizes changes to the requirements.

Cons – the design must be illustrated on papers which take a long time until programming begins. And paper-based process is poor communication.

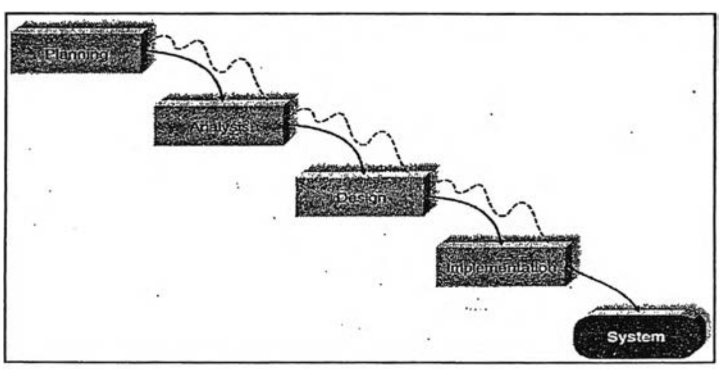


Figure 2.4: The waterfall development methodology
(Dennis and Wixom, 2003: 9)

- **Parallel Development**

"Instead of doing the design and implementation in sequence, it performs a general design for the whole system and then divides the project into a series of distinct subprojects that can be designed and implemented in parallel." (Dennis and Wixom, 2003: 10)

This approach can reduce time line of project, so the systems can be delivered faster. However, it increases chance to rework.

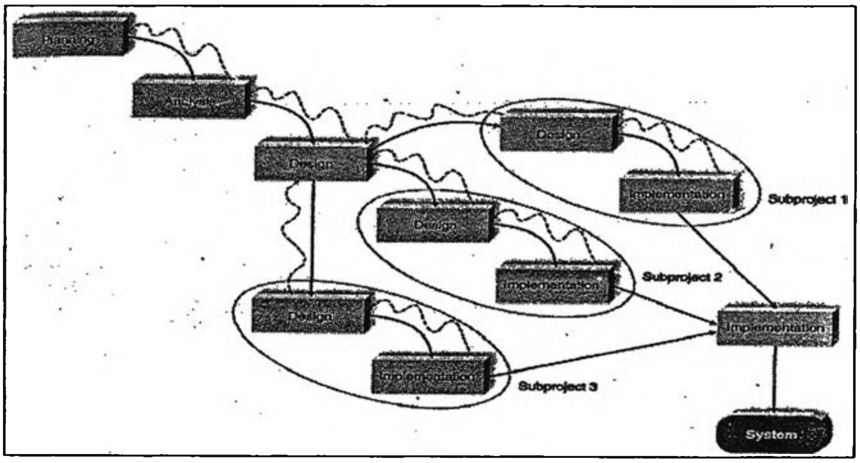


Figure 2.5: The parallel development methodology
(Dennis and Wixom, 2003: 11)

2.3.2.2 Rapid Application Development (RAD)

"RAD attempts to address both weaknesses of the structure development methodologies: long development times and the difficulty in understanding a system from paper-based description. RAD methodologies adjust the SDLC phases to get some part if the system developed quickly and into the hands of the users." (Dennis and Wixom, 2003: 10)

- **Phased development**

"The phased development methodology breaks the overall system into a series of versions that are developed sequentially. The analysis phase identifies the overall system concept, and the project team, users, and system sponsor then categorize the requirements into a series of versions." (Dennis and Wixom, 2003: 11)

This approach is mixed up from structured and parallel design.

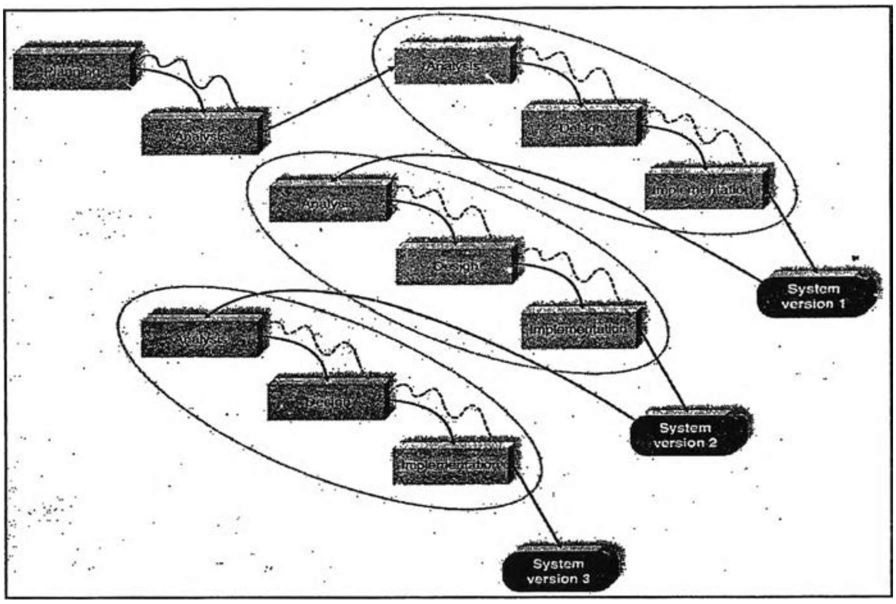


Figure 2.6: The phased development methodology
(Dennis and Wixom, 2003: 12)

• **Prototyping**

"The prototyping methodology performs the analysis, design, and implementation phases concurrently, and all three phases are performed repeatedly in a cycle until the system is completed. With this approach, the basics of analysis and design are performed, and work immediately begins on the system prototype, a 'quick-and-dirty' program that provides a minimal amount of features." (Dennis and Wixom, 2003: 12-13)

Advantage of the prototyping is it performs very quickly to provide 'quick-and -dirty' systems that allow users to interact with. It is used for confirming that the view points about the systems from both users and team are the same. It reduces time consuming and encourages refining real requirements. However, it is hard to use the prototyping for developing full systems because the errors handling is ignored while developing it.

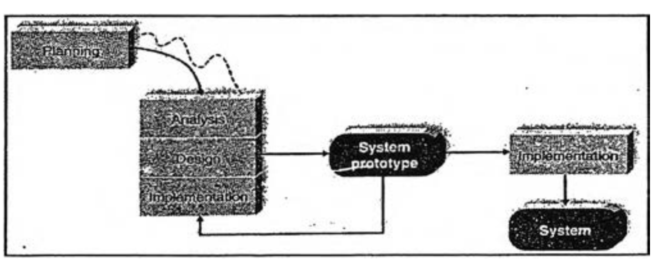


Figure 2.7: The prototyping methodology
(Dennis and Wixom, 2003: 13)

- **Throwaway Prototyping**

"Throwaway prototyping has a relatively thorough analysis phase that is used to gather information and to develop ideas for the system concept. However, many of the features suggested by the users may not be well understood, and there may be challenging technical issues to be solved." (Dennis and Wixom, 2003: 14)

Each issue is analyzing, designing, and building a design prototype. These prototypes are only a set of mock-up screens which not full systems but it contains enough details that enable people to understand the structure of the overall system – especially the focused issues that users need to know how the system fix them. When the users grab the overall solutions and agree to proceed, the project will be moved to design and implementation phase. At this point, the difference between the throwaway prototyping and the prototyping is the throwaway prototyping is actually thrown away while the prototyping is modified and used in full system development.

"Throwaway prototyping balances the benefits of well-thought-out analysis and design phases with the advantages of using prototypes to refine key issues before a system is built." (Dennis and Wixom, 2003: 14)

Compared with the prototyping, the throwaway prototyping takes longer time to deliver the final system. However, this approach generally produces more reliable systems.

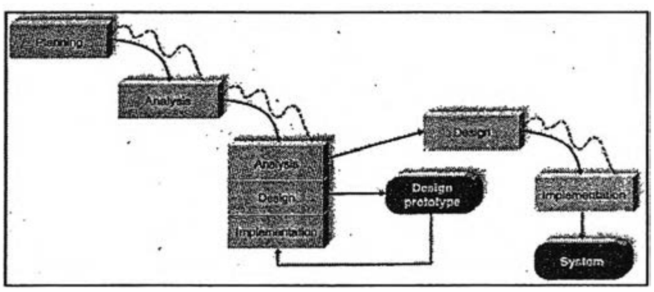


Figure 2.8: The throwaway prototyping methodology
(Dennis and Wixom, 2003: 14)

2.3.2.3 Agile Development

"These programming-centric methodologies have few rules and practices, all of which are fairly easy to follow. They focus on streamlining the SDLC by eliminating much of the modelling and documentation overhead, and the time spent on those tasks. Instead, projects emphasize simple, iterative application development." (Dennis and Wixom, 2003: 15)

- **Extreme Programming (XP)**

"XP uses continuous testing, simple coding performed by pairs of developers, and close interactions with end users to build systems very quickly. After a superficial planning process, projects perform analysis, design, and implementation phases iteratively. The system functionality grows over time." (Dennis and Wixom, 2003: 15)

The XP project starts from is users talk about their requirements for the new system. Then, programmers develop some small modules and let them test whether the systems meet their needs or not. In this approach, users must be available to clarify all issues.

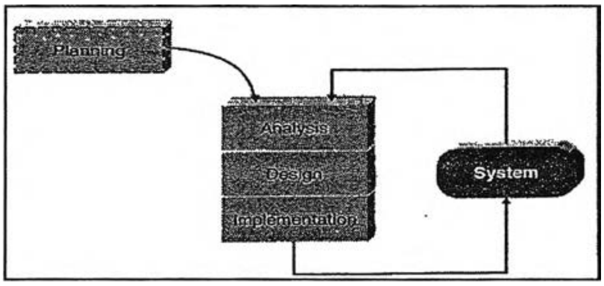


Figure 2.9: The extreme programming methodology
(Dennis and Wixom, 2003: 15)

In summary, table 2.1 is shown comparison of all approaches and criteria for selecting a methodology that can be applied for any IT projects.

Ability to Develop Systems	Structured Methodologies			RAD Methodologies		Agile Methodologies
	Waterfall	Parallel	Phased	Prototyping	Throwaway Prototyping	XP
with Unclear User Requirements	Poor	Poor	Good	Excellent	Excellent	Excellent
with Unfamiliar Technology	Poor	Poor	Good	Poor	Excellent	Poor
that are Complex	Good	Good	Good	Poor	Excellent	Poor
that are Reliable	Good	Good	Good	Poor	Excellent	Good
with a Short Time Schedule	Poor	Good	Excellent	Excellent	Good	Excellent
with Schedule Visibility	Poor	Poor	Excellent	Excellent	Good	Good

Table 2.1: Criteria for Selecting a Methodology
(Dennis and Wixom, 2003: 16)

2.3.3 Strategic Planning – A Framework for IT Systems Development

"Strategic Planning is the process of identifying long-term organisational goals, strategies and resources. Strategic planning looks beyond day-to-day activities and focuses on a horizon that is three, five, or even 10 years in the future." (Shelly et al., 2003: 48)

2.3.3.1 SWOT Analysis

"During strategic planning, top managers ask a series of questions that is called a **SWOT analysis** because it examines a company's strengths (S), weakness (W), opportunities (O), and threats (T)." (Shelly et al., 2003: 48)

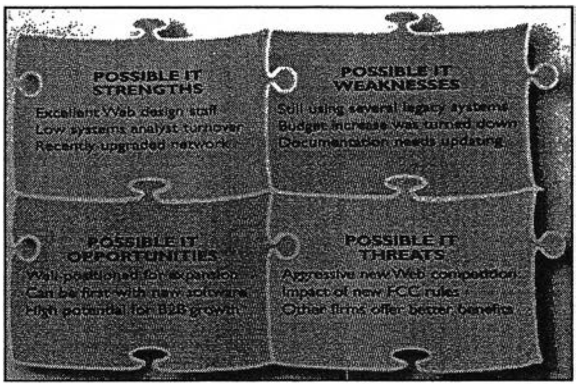


Figure 2.10: SWOT Analysis
(Shelly et al., 2003: 48)

2.3.3.2 Vision, Mission of the company

"A mission statement describes a company for its stakeholders and briefly states the company's overall purpose, products, services, and values" (Shelly et al., 2003: 49)

So outputs of this section are vision and mission statements of the company.

2.3.3.3 Reasons for Systems Projects Checklist

The checklist below is a sample options that is used for analysing the company about which benefits that they want to gain from information system.

Table 2.2: Reasons for Systems Projects

☐ Improved services
☐ Better performance
☐ More information
☐ Stronger control
☐ Reduce cost

(Summarised from Shelly et al., 2003: 53)

2.3.3.4 Internal Factors Checklist

The checklist below is a sample factors that is used for analysing about which internal factors will encourage or block information system acquisition.

Table 2.3: Internal Factors

☐ Strategic plan
☐ Top managers
☐ User requests
☐ Information Technology department
☐ Existing system

(Summarised from Shelly et al., 2003: 55)

2.3.3.5 External Factors Checklist

The checklist below is a sample factors that is used for analysing about which external factors will encourage or block information system acquisition.

Table 2.4: External Factors

☐ Technology
☐ Suppliers
☐ Clients
☐ Competitors
☐ The economy
☐ Government

(Summarised from Shelly et al., 2003: 55-57)

2.3.3.6 Evaluation of Systems Requests

When acquiring system requests, it usually results in a lot of requests which to many for the IT department to response all of them. So it must have method of prioritising requests, rearranging, grouping, and selecting the most important issues to response in this phase. The method is depended on company's culture. Generally, it is finished by voting from all staff, or executives' judgement.

2.3.3.7 Evaluate feasibility

"A feasibility study uses four main yardsticks to measure a proposal: operational feasibility, technical feasibility, economy feasibility, and schedule feasibility. Sometimes a feasibility study is quite simple and can be done in a few hours. If the request involves a new system or a major change, however, extensive fact-finding and investigation is required." (Shelly et al., 2003: 58)

- **Operational Feasibility**

"Operational feasibility means that a proposed system will be used effectively after it has been developed. If users have difficulty with the new system, it will not produce the expected benefits." (Shelly et al., 2003: 59)

Table 2.5: Sample questions for Operational Feasibility

Does management support the project?
Do users support the project?
Is the current system well liked and effectively used?
Do users see the need for change?
Will the new system result in a workforce reduction? If so, what will happen to affected employees?
Will the new system require training for users? If so, is the company prepared to provide the necessary resources for training current employees?
Will users be involved in planning the new system right from the start?
Will the new system place any new demands on users or require any operating changes? Will performance decline in any way? If so, will an overall gain to the organisation outweigh individual looses?
Will clients experience adverse effects in any way, either temporarily or permanently?

(Summarised from Shelly et al., 2003: 59-60)

- **Technical Feasibility**

"Technical feasibility refers to the technical resources needed to develop, purchase, install, or operate the system." (Shelly et al., 2003: 60)

Table 2.6: Sample questions for Technical Feasibility

Does the company have the necessary hardware, software, and network resources? If not, can those resources be acquired without difficulty?
Does the company have the needed technical expertise? If so, can it be acquired?
Does the proposed platform have sufficient capacity for future needs? If not, can it be expanded?
Will the hardware and software environment be reliable?
Will it integrate with other company information systems, both now and in the future?
Will it interface properly with external systems operated by clients and suppliers?
Will the combination of hardware and software supply adequate performance? Do clear expectations and performance specification exist?
Will the system be able to handle future transaction volume and company growth?

(Summarised from Shelly et al., 2003: 60)

- **Economic Feasibility**

  "Economy feasibility means that the projected benefits of the purposed system outweigh the estimated costs usually considered the total cost of ownership (TCO), which includes ongoing support and maintenance costs, as well as acquisition costs." (Shelly et al., 2003: 60)

- **Schedule Feasibility**

  "Schedule feasibility means that a project can be implemented in an acceptable timeframe." (Shelly et al., 2003: 61)

Table 2.7: Sample questions for Schedule Feasibility

Can the company or the IT team control the factors that affect schedule feasibility
Has management established a firm timetable for the project?
What conditions must be satisfied during the development of the system?
Will an accelerated schedule pose any risks? If so, are they risks acceptable?
Will project management techniques be available to coordinate and control the project?
Will a project manager be appointed?

(Summarised from Shelly et al., 2003: 61)

2.4 UML (Unified Modelling Language)

"The Unified Modelling Language (UML) is a graphical language, or a notation, for modelling systems analysis and design concepts in an object-oriented fashion." (Avison and Fitzgerald, 2003: 247)

"The UML is an official definition of a pictoral language where there are common symbols and relationships that have one common meaning. Learning the UML, therefore, is essential to being able to use pictures to cheaply, flexibility, and quickly experiment with solutions" (Kimmel, 2005: 3)

The UML is a standard language that used for illustrates some processes or some system of the company to deliver various types of models which each model has unique meaning that people understand the same. UML contains 12 types of diagrams aligned in 3 categories.

- **Static application diagrams;**
  - Class Diagrams;

    "The purpose of a class diagram is to depict the classes within a model. In an object oriented application, classes have attributes (member variables), operations (member functions) and relationships with other classes. The UML class diagram can depict all these things quite easily." (Martin, 1997: 1-2)

    "The greatest value is to show classes and their relationships from various perspectives in the way that will help convey the most useful understanding." (Kimmel, 2005: 9)

  - Object Diagrams

    "An object diagram shows this relation between the instantiated classes and the defined class, and the relation between these objects, in the logical view of the system. These are very useful to explain smaller portions of your system, when your system class diagram is very complex, and also sometimes recursive." (Chitnis, Tiwari, and Ananthamurthy, 2003a : online)

- Component Diagrams

  "UML component diagrams are great for doing this as they enable you to model the high-level software components, and more importantly the interfaces to those components. Once the interfaces are defined, and agreed to by your team, it makes it much easier to organize the development effort between sub teams." (Ambler, unknown : online)

- Deployment Diagrams

  "A deployment diagram models the run-time architecture of a system. It shows the configuration of the hardware elements (nodes) and shows how software elements and artefacts are mapped onto those nodes." (Sparx Systems Pty Ltd., unknown : online)

• **Dynamic behaviour diagrams**

- Use Case Diagrams (detail in 2.4.2)
- State Chart Diagrams (detail in 2.4.3)
- Sequence Diagrams (detail in 2.4.4)
- Activity Diagrams (detail in 2.4.5)
- Collaboration Diagrams

  Collaboration diagram is one of two kinds of interaction diagrams which show the steps of actions and messages that flow from start to end of the processes. It presents almost the same structure as sequence diagram but it is not locates the action in time line.

  "Because collaboration diagram does not indicate a time ordering visually, we number the messages to indicate the order in which they occur. Some tools will convert interaction diagrams between sequence and collaboration automatically, but it isn't necessary to create both kinds of diagrams. Generally, a sequence diagram is perceived to be easier to read and more common." (Kimmel, 2005: 9)

- **Diagrams that help to organise and manage applications.**
  - Packages
  - Subsystems
  - Models

However, it is not required to create all diagrams to for the system. Selecting some important diagrams which useful for analysis, lead to better design, and understandable by related people is enough.

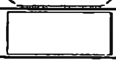
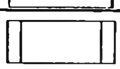
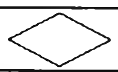
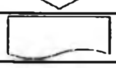
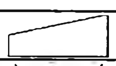
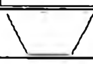
As a result, there are 4 diagrams that are selected for illustrate the business processes of the case study include Use Case diagram, Sequence diagram, Activity diagram, and State Chart diagram. It is better to start diagramming from the fundamental method, so before talking about UML diagrams the basic flowchart is introduced to render processes in the easiest way which detail in 2.4.1.

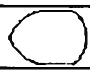
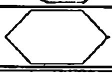
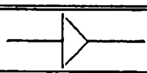
2.4.1 Basic Flowchart

"Flowcharts are diagrams that show the steps in a process. Basic flowcharts are easy to create and, because the shapes are simple and visual, they are easy to understand" (Microsoft, Inc., unknown : online)

There are many symbols available in Microsoft™ Visio basic flowchart chapes. The table below presents some of the symbols that commonly used.

Table 2.8: Basic Flowchart Symbols

	**Terminator** Use this shape for the first and last step of your process.
	**Process** This shape represents a step in your process.
	**Predefined process** Use this shape for a set of steps that combine to create a sub-process that is defined elsewhere, often on another page of the same drawing.
	**Decision** This shape indicates a point where the outcome of a decision dictates the next step. There can be multiple outcomes, but often there are just two —yes and no.
	**Document** This shape represents a step that results in a document.
	**Data** This shape indicates that information is coming into the process from outside, or leaving the process. This shape can also be used to represent materials and is sometimes called an Input/Output shape.
	**Stored data** Use this shape for a step that results in information being stored.
	**On-page reference** This small circle indicates that the next (or previous) step is somewhere else on the drawing. This is particularly useful for large flowcharts where you would otherwise have to use a long connector, which can be hard to follow.
	**Manual input** This is a step where a person provides information to the process.
	**Manual operation** This is a step that must be performed by a person.
	**Direct data** This shape represents information stored so that any single record can be accessed directly. This represents how a computer hard-drive stores data.

	**Off-page reference** When you drop this shape onto your drawing page, a dialog box opens where you can create a set of hyperlinks between two pages of a flowchart or between a sub-process shape and a separate flowchart page that shows the steps in that sub-process.
	**Sequential data** This shape represents information stored in sequence, such ad data on a magnetic tape. When data is stored in sequence, it must be retrieved in sequence. For example, in order to access record 7 you would have to first go through records 1 through 6.
	**Card and Paper tape** This shape represents a physical card or paper tape. Early computer systems used a system of punch cards and paper tape to store and retrieve data and to store and run programs.
	**Display** This shape represents information that is displayed to a person, usually on a computer screen.
	**Preparation** This shape indicates where variables are initialized in preparation for a procedure.
	Parallel mode This shape shows where two different processes can operate simultaneously
	**Control transfer** This shape indicates a step that goes to a step other than the typical next step when certain conditions are met.

(Adapted from Microsoft, Inc., unknown, online)

2.4.2 Use Case Diagram

"The purpose of a use case is to describe how a system will be used – to describe its essential purpose. The purpose of use case diagrams is to capture the essential purposes visually" (Kimmel, 2005: 17)

"A use case is usually described as a high-level function of a system and in UML the use case diagram shows the associations or interactions (as lines) between actors (stick figures) in the system and the use cases (ellipses). The diagram also shows the boundary of the system as a rectangle with the use cases within." (Avison and Fitzgerald, 2003: 250)

Use case diagram generally has 3 major components.

- **Stick figure;** is called actor represents person or something that acts.
- **Lines;** describe relationship between actor and use case.
- **Ovals;** is called use case represents action or capability to do something.

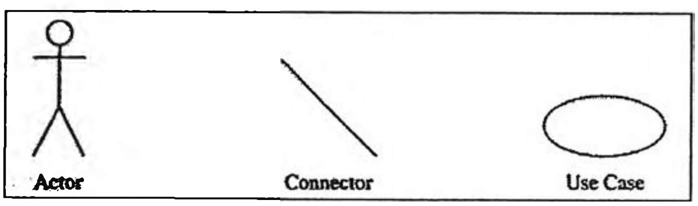


Figure 2.11: Components of use case
(Kimmel, 2005: 21)

Use case also support objected-oriented system which enable 2 special relationships as below.

- **Extend;** indicates inheritance or generalisation which means one use case inherit all general properties from the pointed use case.
- **Include;** indicates dependency which means one use case has to wait for result from the pointed use case before perform its action.

The easier view of use case is view it as 'to-do-list' – lists of things / actions that someone have to do. So the first step of creating use case is listing all tasks that the all people in the company must do, select and normalise major actions and group them into specific roles, then draw the use case diagram.

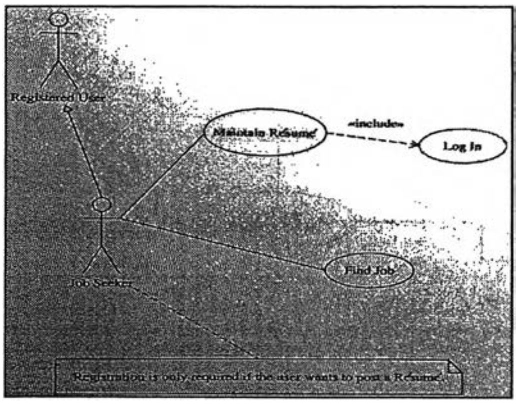


Figure 2.12: Sample Use Case Diagram
(Kimmel, 2005: 41)

The detail that explains each use case is available in standard document called 'Use Case description' table which includes;

Table 2.9: Use Case description

Use Case Name	[Use Case Name]
Participating actor	[Roles that have authorised to do action]
Entry Condition	[Specific conditions before do action]
Flow Event	Steps of work • Step1 • Step2 • Etc.
Exit Condition	[Specific conditions to finish this action]

(Adapted from Motif Technology,Inc., 2008)

2.4.3 State Chart Diagrams

"State charts (also known as state machines) are good at showing an object's state over many use cases and good at defining protocol that describe a correct orchestration of messages. State charts are ideally suited for describing the behaviour of user interfaces and device controllers for real-time systems" (Kimmel, 2005: 156)

"In state chart diagram (usually just called state diagrams) the state is a particular set of values of the attributes of an object at a particular time. When the value of an attribute changes. The object changes state. Obviously not all changes of attribute values are of interest." (Avison and Fitzgerald, 2003: 253)

State Chart diagram is used to help programmer better populate clearer view of business flows. It will illustrate conditions and behaviours that cause the objects move from nodes to nodes. The conditions that make objects change their state are very useful for programmer when they develop system.

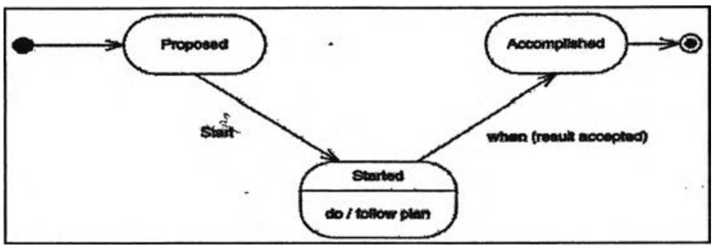


Figure 2.13: Sample State Chart Diagram
(Noran, 2004: 10)

Table 2.10: Components of State Model

Element and its Description	Symbol
Initial State: This shows the starting point or first activity of the flow. Denoted by a solid circle. This is also called as a "pseudo state," where the state has no variables describing it further and no activities.	●
State: Represents the state of object at an instant of time. In a state diagram, there will be multiple of such symbols, one for each state of the Object we are discussing. Denoted by a rectangle with rounded corners and compartments (such as a class with rounded corners to denote an Object). We will describe this symbol in detail a little later.	
Transition: An arrow indicating the Object to transition from one state to the other. The actual trigger event and action causing the transition are written beside the arrow, separated by a slash. Transitions that occur because the state completed an activity are called "triggerless" transitions. If an event has to occur after the completion of some event or action, the event or action is called the guard condition. The transition takes place after the guard condition occurs. This guard condition/event/action is depicted by square brackets around the description of the event/action (in other words, in the form of a Boolean expression).	Event / Action →
History States: A flow may require that the object go into a trance, or wait state, and on the occurrence of a certain event, go back to the state it was in when it went into a wait state—its last active state. This is shown in a State diagram with the help of a letter H enclosed within a circle.	(H)
Event and Action: A trigger that causes a transition to occur is called as an event or action. Every transition need not occur due to the occurrence of an event or action directly related to the state that transitioned from one state to another. As described above, an event/action is written above a transition that it causes.	Event / Action →
Signal: When an event causes a message/trigger to be sent to a state that causes the transition; then, that message sent by the event is called a signal. Represented as a class with the <<Signal>> icon above the action/event.	<<Signal>> Event / Action Event / Action →
Final State: The end of the state diagram is shown by a bull's eye symbol, also called a final state. A final state is another example of a pseudo state because it does not have any variable or action described.	◉

(Chitnis et al., 2003b : online)

2.4.4 Sequence Diagram

"Sequence diagrams show the classes along the top and messages sent between those classes, modelling a single flow through the objects in the system. A sequence diagram implies a time ordering by following the sequence of messages from top left to bottom right." (Kimmel, 2005: 9)

"The sequence diagrams describe sequences of messages between a set of objects. The order and timing of objects are clearly depicted. Messages in a sequence diagram may have parameters and guards. Messages may also be synchronous (must be fulfilled before anything else may be accomplished) or asynchronous (the sender does not wait for an answer before continuing)." (Noran, 2004: 11)

Sequence diagram is an effective tool for illustrate logical flow of the system. It presents step of work including conditions to pass each step based on time.

Table 2.11: Components of Sequence Diagram

Element and its description	Symbol
Object: The primary element involved in a sequence diagram is an Object —an instance of a class. A Sequence diagram consists of sequences of interaction among different objects over a period of time. An object is represented by a named rectangle. The name to the left of the ":" is the object name and to its right is the class name.	abc:ABC
Message: The interaction between different objects in a sequence diagram is represented as messages. A message is denoted by a directed arrow. Depending on the type of message, the notation differs. In a Sequence diagram, you can represent simple messages, special messages to create or destroy objects, and message responses.	testMessage() createMessage() destroyMessage() return

(Chitnis et al., 2003c : online)

"Sequences do not have to complex to be useful. Most important are the objects across the horizontal and each object's lifeline and the order and name of the messages sent between objects." (Kimmel, 2005: 91)

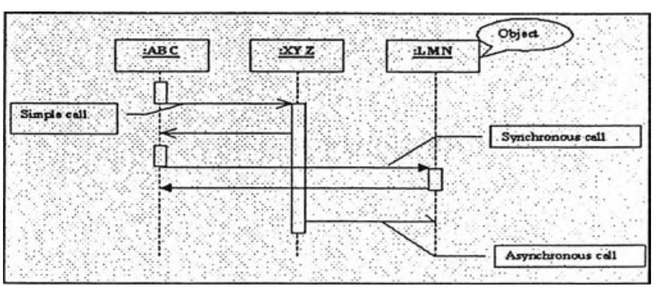


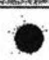
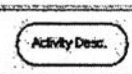
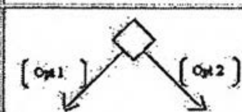
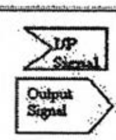
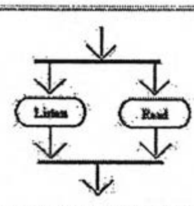
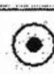
Figure 2.14: Sample Sequence Diagram
(Chitnis et al., 2003c : online)

2.4.5 Activity Diagram

"An activity diagram is the UML version of a flowchart. Activity diagrams are used to analyze processes and, if necessary, perform process reengineering. An activity diagram is an excellent tool for analyzing problems that the system ultimately will have to solve." (Kimmel, 2005: 7-8)

"Activity diagrams represent the business and operational workflows of a system. An Activity diagram is a dynamic diagram that shows the activity and the event that causes the object to be in the particular state." (Chitnis et al, 2003d : online)

Table 2.12: Components of Activity Diagram

Element and its description	Symbol
Initial Activity: This shows the starting point or first activity of the flow. Denoted by a solid circle. This is similar to the notation used for Initial State.	
Activity: Represented by a rectangle with rounded (almost oval) edges.	 Activity Desc.
Decisions: Similar to flowcharts, a logic where a decision is to be made is depicted by a diamond, with the options written on either sides of the arrows emerging from the diamond, within box brackets.	 Opt 1, Opt 2
Signal: When an activity sends or receives a message, that activity is called a signal. Signals are of two types: Input signal (Message receiving activity) shown by a concave polygon and Output signal (Message sending activity) shown by a convex polygon.	 I/P Signal, Output Signal
Concurrent Activities: Some activities occur simultaneously or in parallel. Such activities are called concurrent activities. For example, listening to the lecturer and looking at the blackboard is a parallel activity. This is represented by a horizontal split (thick dark line) and the two concurrent activities next to each other, and the horizontal line again to show the end of the parallel activity.	 Listen, Read
Final Activity: The end of the Activity diagram is shown by a bull's eye symbol, also called as a final activity.	

(Chitnis et al, 2003d : online)

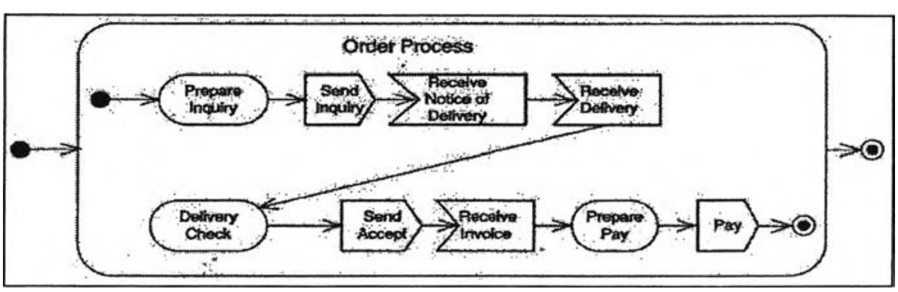


Figure 2.15: Sample Activity Diagram
(Noran, 2004: 11)

2.5 IDEF (Integration DEFinition)

"The IDEF family of languages is strongly linked to the Integrated Computer Aided Manufacturing (ICAM), which aimed to use the then emerging (1970's) computer technology in order to improve the manufacturing productivity. These languages cover a large area (IDEFØ up to IDEF14), from function modelling to information, simulation, objected-oriented analysis and design and knowledge acquisition" (Noran, 2004: 2).

"IDEF0 may be used first to define the requirements and specify the functions, and then to design an implementation that meets the requirements and performs the functions. For existing systems, IDEF0 can be used to analyze the functions the system performs and to record the mechanisms (means) by which these are done. The result of applying IDEF0 to a system is a model that consists of a hierarchical series of diagrams, text, and glossary cross-referenced to each other. The two primary modelling components are functions (represented on a diagram by boxes) and the data and objects that inter-relate those functions (represented by arrows)." (Draft Federal Information Processing Standards Publication 183, 1993: vii.)

"IDEF0 allows the user to depict a view of the process including the inputs, outputsm controls and mechanisms (which are referred to generally as ICOMs):

- **Inputs;** are resources consumed or transformed (refined) by the process;
- **Outputs;** are the things created through the consumption / transformation of the inputs by the process;
- **Controls;** are the things guiding the process: policies, guidelines, standards, laws;
- **Mechanisms;** are the agents that accomplish the actions (activities) contained by the process. Examples: people or manual and automated tools;" (Noran, 2004: 17)

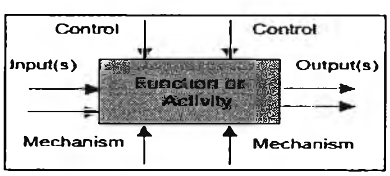


Figure 2.16: IDEFØ
(Noran, 2004: 17)

2.6 BPI (Business Process Improvement)

"The 'width' scope of a BPR project is concerned with the flow of products, information and other resources. Its prime objective is to identify the enablers that speed up the flow." (Botha and Eloff, 2001: 127)

Focusing on workflow, it is aligned in business process re-engineering (BPR) which required process definition, and IT application to connect between workflow services and user interactions as the figure below.

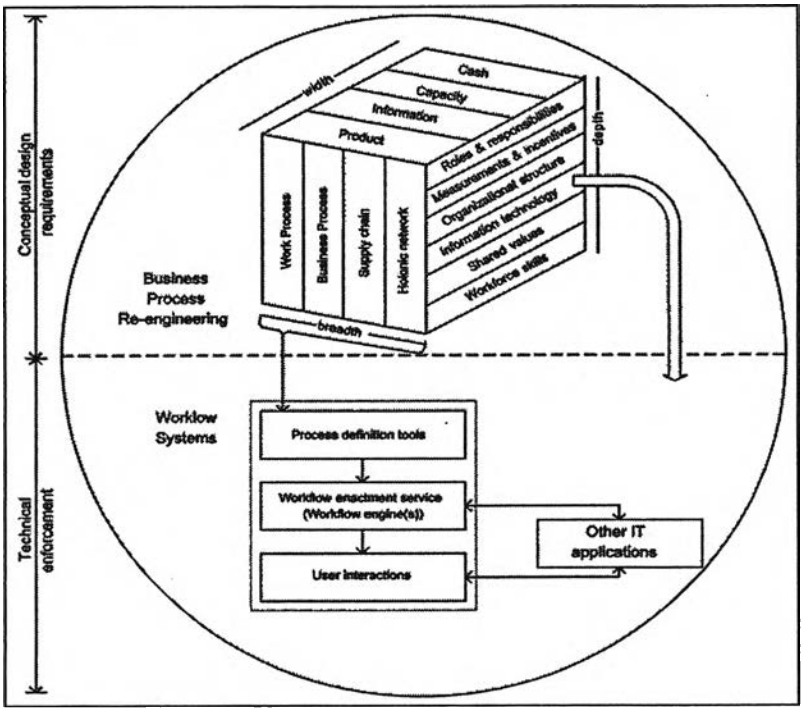


Figure 2.17: The workflow sphere
(Botha and Eloff, 2001: 128)

"By automating the flow of documents and procedures throughout an organisation, they increase the accuracy of clerical tasks – and can make quite startling efficiency improvements" (Brunwin, 1994: 27).

Brunwin also stated that the workflow can encourage BPR by illustrating overall business processes and redefining new ones. Moreover, workflow can be viewed as a bridge between technology and business. Information Technology can not represent all business needs because it has 'specific and systematic processes' which quite conflict with real business behaviours. Business needs flexibility to change. When business processes are changed, it means disposal of current IT and new acquisition.

2.7 Workflow Analysis and Management

J.L. Caro, A. Guevara and A. Aguayo talked about concept of workflow management include define current workflow, redesign, and apply it to real business.

Method of workflow modelling and how to implement CASE tools and WMFS to develop information system that support workflow is shown in figure below.

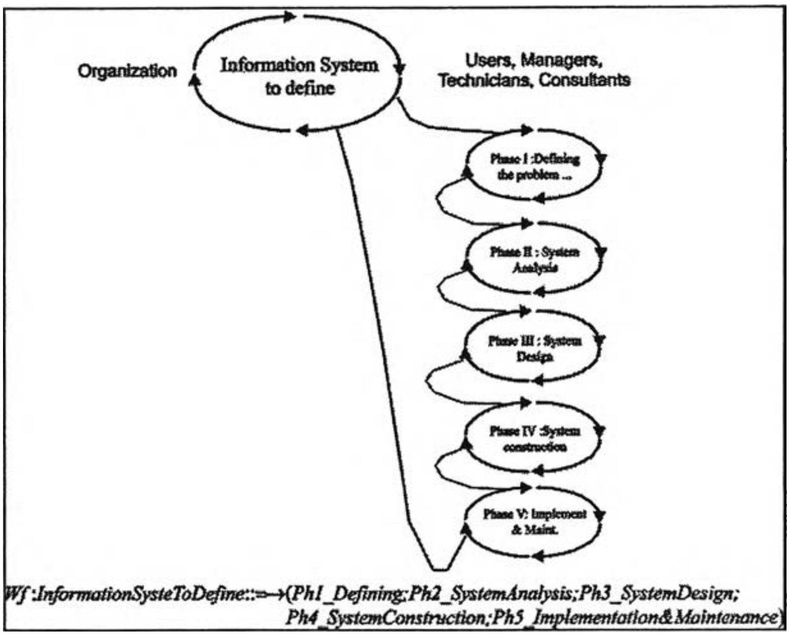


Figure 2.18: Workflow, IS Plan
(Caro, Guevara and Aguayo, 2003: 216)

"Simulation of the workflow component of a computer supported co-operation work (CSCW) system has the potential to reduce the costs of system implementation, while at the same time improving the quality of the delivered system". (Greasley, 2003: 256)

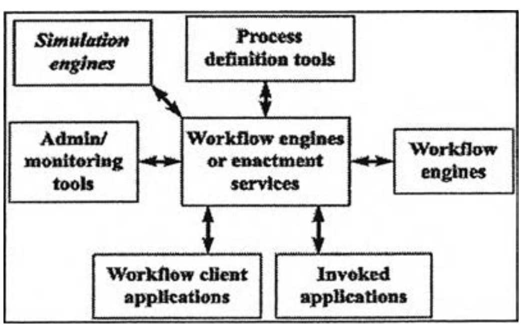


Figure 2.19: Components of a workflow management system – the reference model of the Workflow Management Coalition
(Greasley, 2003: 257)

Greasley focused on simulation processes to test the workflow using Workflow management system (WFMS) in the case study of selling properties.

N.S Ong, and W.C. Foo mensioned about 'Workflow Tracking Systems' (WFS) which is focusing on redesign to improve performance by reducing non-value added costs and activities within manufacturing. WFS used for capture current processes flow behaviors and illustrate in form of diagram that allows workflow improvement. It provides example of assembly processes as below.

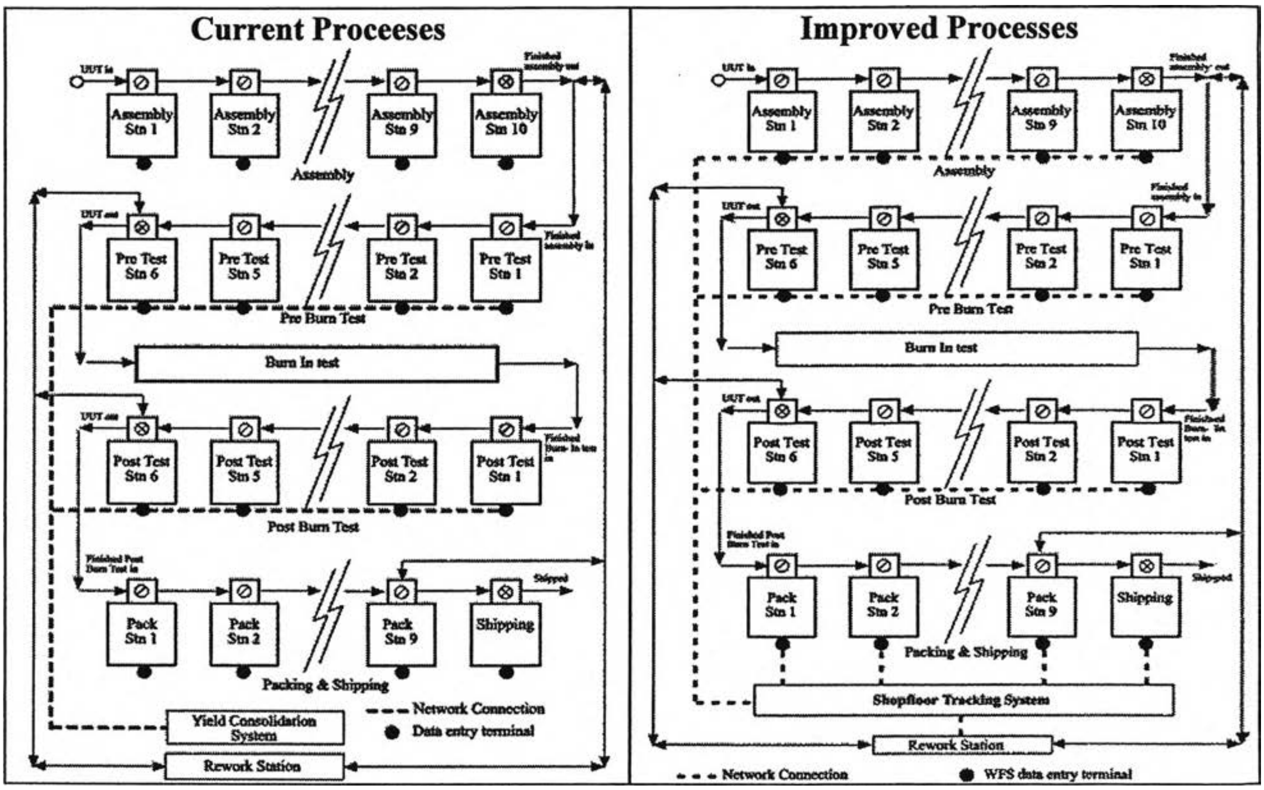


Figure 2.20: Workflow Improvement of Assembly Processes using WFS
(Ong, and Foo, 2004: 36,39).