

Chapter 6

Systems Development

Phases, Tools, and Techniques

Prof. Anita Beecroft, Kwantlen Polytechnic University (2009)
Prof. Tim Richardson, University of Toronto (2011)

Management Information Systems
>>> for the Information Age

Fourth Canadian Edition

Saving Lives through Systems Development and Integration

- Centers for Disease Control (CDC) track a wealth of information such as:
 - Antimicrobial-resistant infections in hospitals
 - Influenza outbreaks
 - Terrorist biochemical attacks
 - Bacteria counts in rivers and stagnant ponds.
- Unfortunately, most of that information is stored in separate IT systems that do not communicate with each other.

Saving Lives through Systems Development and Integration

- The CDC is using a service-oriented architecture (SoA) to integrate all those systems and information.
- It wants to bring all of the information together within a service-oriented architecture (SoA).
- Every component of each IT system (database file, server, CRM software solution, etc.) can be viewed as a building block.
- Within an SoA, those building blocks can be “plugged and played” so that everything works together in an integrated fashion.

Saving Lives through Systems Development and Integration

1. All computers use a common binary base language. That being true, why is it so difficult to get computer systems to easily communicate with each other?
2. In systems development, prototyping is used to build a model of a proposed system. How have you used prototyping in your personal life to build something?
3. Outsourcing – going to another company for systems development – is big business. Why would the CDC *not want* to pursue outsourcing?

Introduction

Many companies are dependent on information technology to:

- Provide services
- Develop products
- Administer daily activities
- Perform short-term and long-term management functions.

There are many factors that need to be addressed when developing a successful information system.

Introduction

A new information system can be built using:

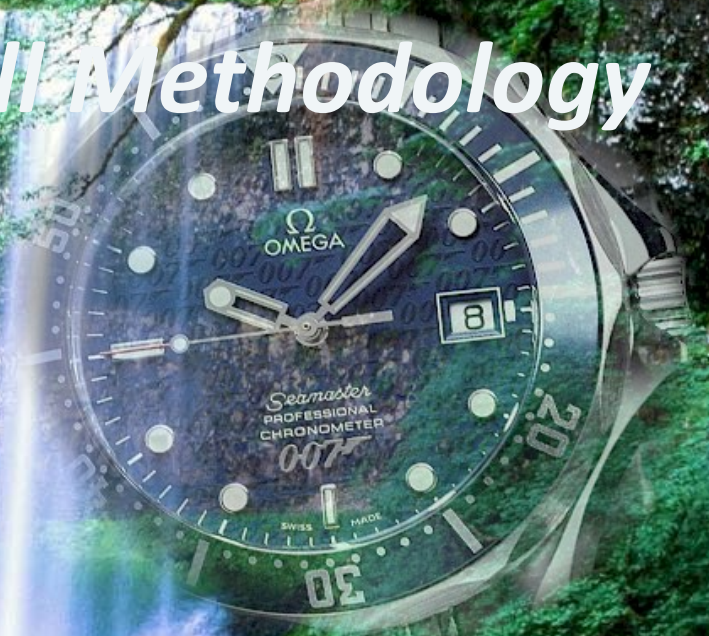
- 1. *Insourcing*** with IT specialists who are employees
- 2. *Selfsourcing (end-user development)*** where end users take with little or no help from IT specialists
- 3. *Outsourcing*** with a third-party which is given a specified cost, time frame and level of service to adhere to.

Insourcing and the SDLC

- The *systems development life cycle (SDLC)* is a structured step-by-step approach for developing information systems. It has seven key phases with many activities in each phase.
- This type of SDLC is called a ***waterfall methodology*** because each phase of the SDLC is followed by another, from planning through to implementation.

SDLC as a Waterfall Methodology

1. Planning
2. Analysis
3. Design
4. Development
5. Testing
6. Implementation
7. Maintenance



SDLC Phases & Major Activities

■ Figure 6.2

SDLC Phase	Activities
1. Planning	<ul style="list-style-type: none">• Define the system to be developed• Set the project scope• Develop the project plan including tasks, resources, and timeframes
2. Analysis	<ul style="list-style-type: none">• Gather the business requirements for the system• Prioritize the business requirements
3. Design	<ul style="list-style-type: none">• Design the technical architecture required to support the system• Design system models
4. Development	<ul style="list-style-type: none">• Build the technical architecture• Build the database and programs
5. Testing	<ul style="list-style-type: none">• Write the test conditions• Perform the testing of the system
6. Implementation	<ul style="list-style-type: none">• Write detailed user documentation• Provide training for the system users
7. Maintenance	<ul style="list-style-type: none">• Build a help desk to support the system users• Provide an environment to support system changes

The Systems Development Life Cycle (SDLC), Phases and Activities, and the Waterfall Methodology

Phase 1: Planning

The ***planning phase*** is comprised of three primary planning activities:

1. Define the system to be developed.

It isn't possible to fulfill every proposal for a new system choices are based on the organization's priorities or ***critical success factors*** (things that are critical to the success of the organization).

Phase 1: Planning

2. Set the scope of the project.

- ***Project scope*** is a high-level overview of a project.
It can usually be summed up in a paragraph called the ***project scope document***.
- Defining the project scope helps address the tendency to allow:
 - ***scope creep*** (scope of the project gets bigger than originally intended)
 - ***feature creep*** (extra features added).

Phase 1: Planning

2. Set the scope of the project.

- When business people use the expression “high-level overview”
- It does not refer to status or rank
- It refers to being high (tree top level) looking down
- Being able to see all the component parts



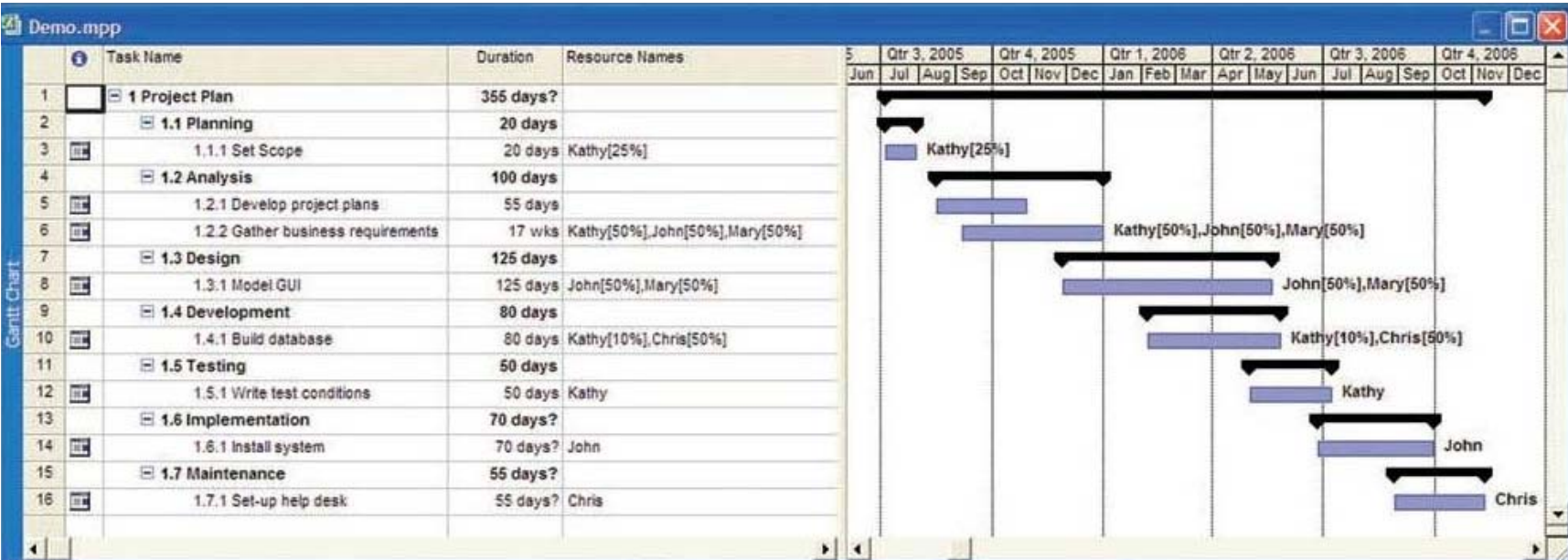
Kwantlen and Richardson

Chpt 6 slide number 12

Phase 1: Planning

3. Develop the ***project plan*** including tasks, resources, and timeframes. The project plan should answer what, when, and who questions of system development.
- ***Project milestones*** represent key dates by which a group of activities must be done.
 - The ***project manager*** is an expert in project planning and management . He/she defines and develops the project plan and tracks the plan to ensure all key project milestones are completed on time.

Phase 1: Planning



A Sample Project Plan

Phase 2: Analysis

The ***analysis phase*** involves both end users and IT specialists working together to gather, understand, and document the business requirements for the proposed system.

Phase 2: Analysis

There are two main activities during analysis:

1. Gathering the business requirements

- ***Business requirements*** list the details of what the end-users consider essential to the new system's success.
- A useful way to gather all of the user requirements is to hold a ***joint application development (JAD)*** session. This session may last for several days during which users and IT specialists make a concentrated effort to record all of the business requirements for the new system.

Phase 2: Analysis

2. Prioritize the requirements. At the end of this phase, a formal comprehensive document called the *requirements definition document* is prepared. Users sign off on this document.

A word of advice: Take time during analysis to get the business requirements correct. The cost to fix an error in the early stages of the SDLC is relatively small when compared to the cost of correcting something during the later stages.

Phase 3: Design

The ***design phase*** builds a technical blueprint of how the proposed system will work. IT specialists do most of the activities. User involvement begins to decrease.

The two primary activities include design of the:

- 1. technical architecture*** (hardware, software, and telecommunications equipment required to run the system).
- 2. system models*** (sample screens, reports, software, databases, etc.)

Phase 4: Development

The ***development phase*** takes the details of the design documents and changes them into an actual working system. This phase is usually done by IT specialists. The two primary development activities are building the:

1. Technical architecture
2. Database and programs.

Phase 5: Testing

The *testing phase* verifies that the system works and meets all of the business requirements that were in the requirements definition document.

There are two primary activities in this phase:

1. Writing the *test conditions*
(steps the system must perform along with the expected results of each step)
2. Doing the testing.

Phase 5: Testing

These types of testing need to be done:

- ***Unit testing*** – testing individual units of code
- ***System testing*** – verifying that the units of code function correctly when put together in a system
- ***Integration testing*** – verifying that separate systems work together
- ***User acceptance testing (UAT)*** – determining if the system satisfies the business requirements and enables users to perform their work correctly

Phase 5: Testing

Other types of tests that need to be done:

- **Risk-based testing** – functions/features that are most prone to failure are tested before those involving less risk
- **Volume testing** – determining if the system can handle large volumes of transactions
- **Stress (load) testing** – verifying that the system can perform adequately under unfavourable circumstances e.g. system crash
- **Regression testing** – looking for reappearance of error(s) previously corrected during the design phase

Phase 6: Implementation

During the ***implementation phase***, the system is distributed to the users who begin to use it to perform their everyday activities. The two main implementation activities are:

1. Writing detailed ***user documentation*** on how to use the new system.
2. Training users either ***on-line*** or in ***workshops***.

Phase 6: Implementation

There are 4 ways to implement a new system:

- ***Parallel implementation*** using both the old and new system simultaneously
- ***Plunge implementation*** discarding the old system and immediately starting to use the new
- ***Pilot implementation*** starting with a small group of users and gradually adding more users
- ***Phased implementation*** installing one part of the system before another parts

Phase 7: Maintenance

During the ***maintenance phase***, the new system is monitored and supported so that it continues to meet its business goals. This is ongoing. The two primary maintenance activities are:

1. Setting up a help desk so user's questions can be answered.
2. Providing an support environment so that system corrections or improvements are identified, assessed and implemented.

Component-based Development

In the traditional SDLC (Systems Development Life Cycle), computer software is usually written from scratch when it is needed for a new application. In contrast, ***component-based development (CBD)*** builds small self-contained blocks of code (components) that can be reused across a variety of applications.

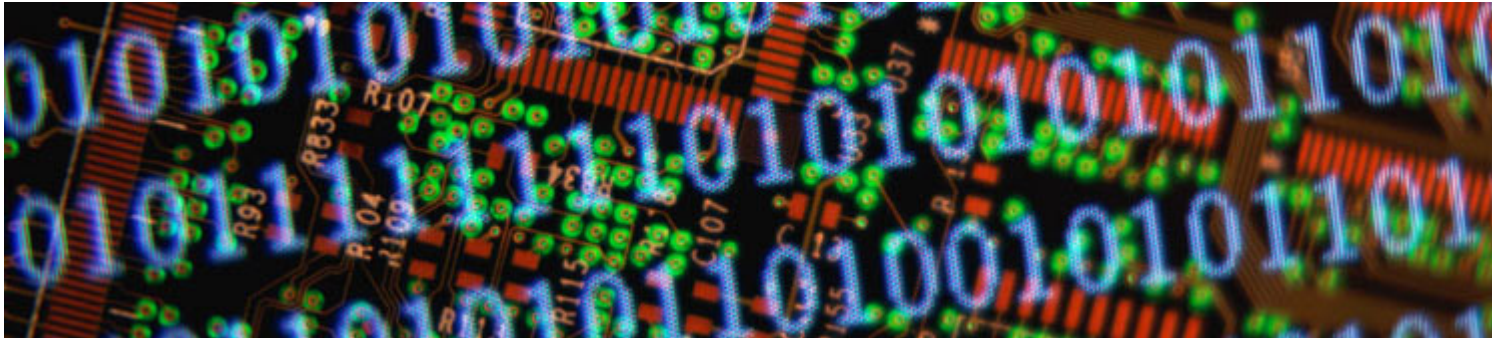
Component-based Development

In CBD, IT specialists look in the software library for code they use reuse.

If they have to build new software, they will write it so it can be used again in the future.

It will become part of the software library collection.

Component-based Development

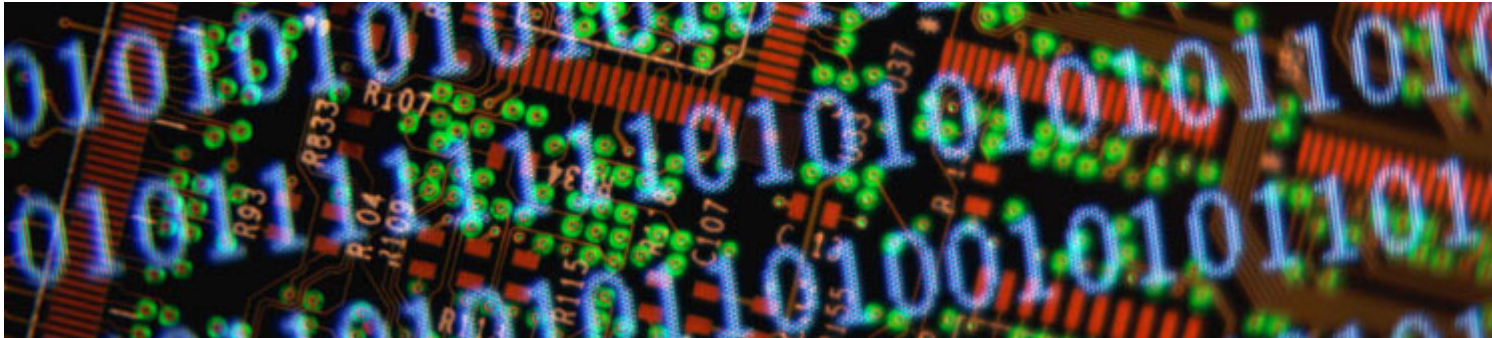


“...used again in the future”

This is a problem of Legacy Software

Programmers want to reduce programming time and “cut and paste” in older software which may not be perfectly appropriate

Component-based Development



But it gets the job done

The cost to the user is sometimes excessive time to run a program cause the software is running long strings of old code to complete a task

Component-Based Development Methodologies

- Rapid application development (RAD)
- Extreme programming (XP)
- Agile methodology

Rapid Application Development (RAD)

Rapid application development (RAD or rapid prototyping) emphasizes extensive user involvement in the rapid and evolutionary construction of working prototypes of a system. This iterative approach speeds up systems development.

If reusable software components can be adopted, the development is even quicker.

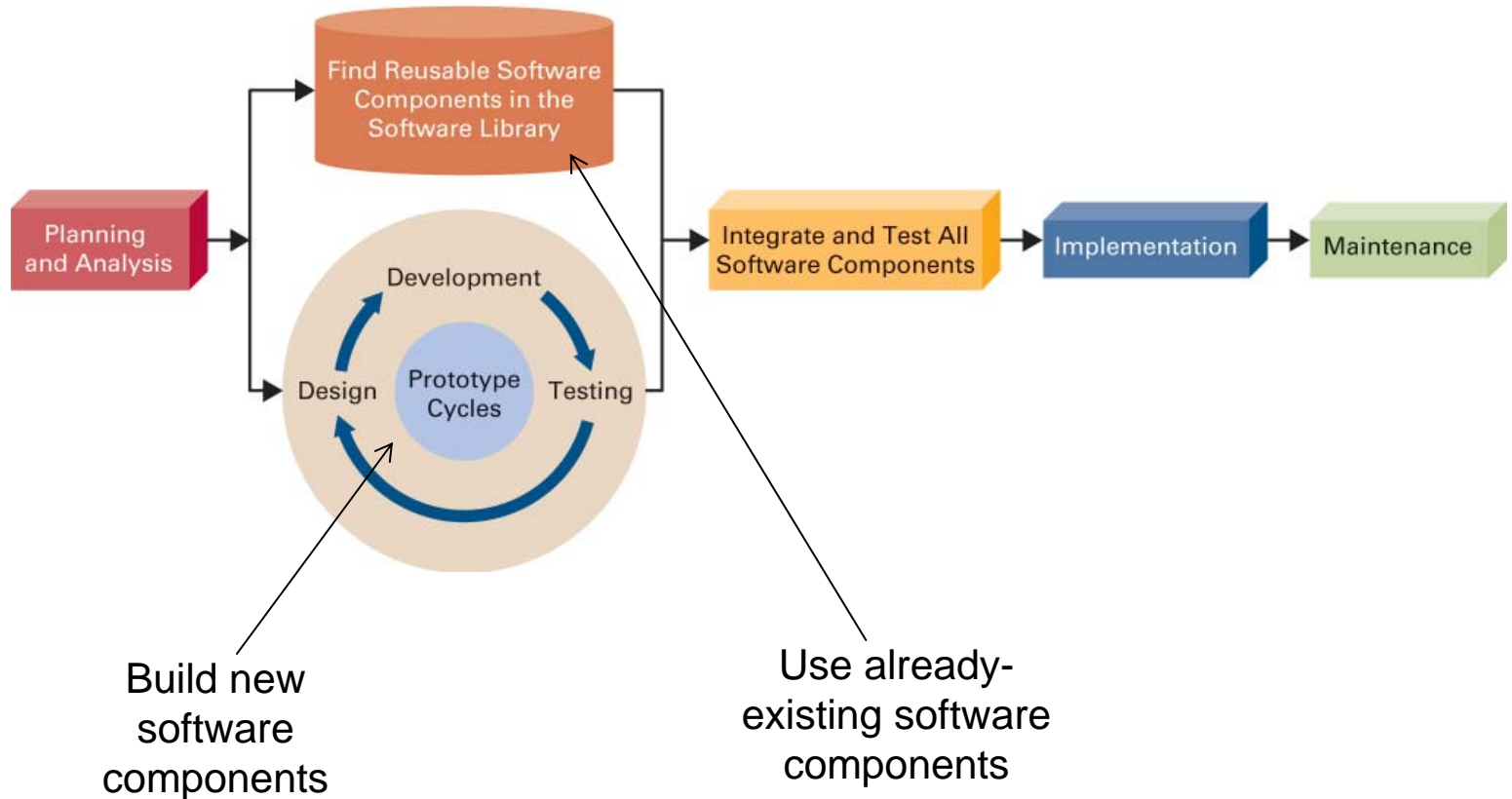
Rapid Application Development (RAD)

iterative approach

Iteration means the act of repeating a process usually with the aim of approaching a desired goal or target or result

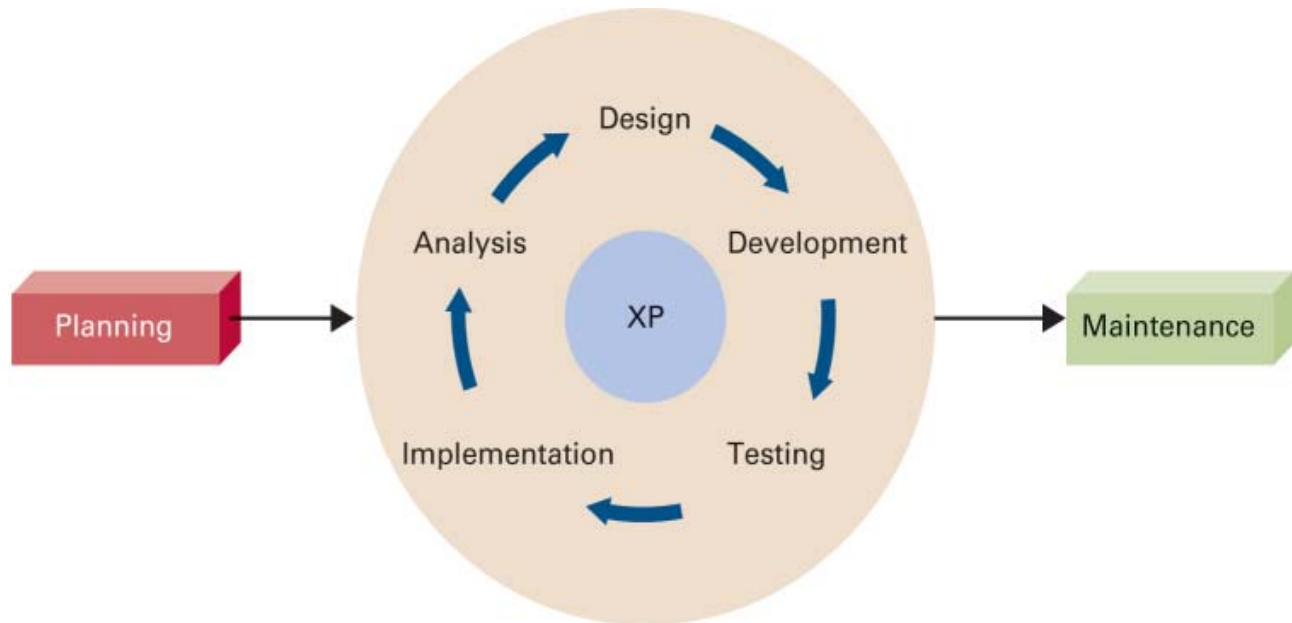
Each repetition of the process is also called an "iteration," and the results of one iteration are used as the starting point for the next iteration.

Rapid Application Development (RAD)



Extreme Programming (XP)

Extreme programming (XP) breaks a project into tiny phases. Developers cannot go on to the next phase until the previous phase is complete.



Agile Methodology

Agile methodology is a form of extreme programming.

It emphasizes limiting the project scope and satisfying the user through early and continuous delivery of useful software components.

An ***agile project*** has a minimum number of requirements and quickly turns them into a deliverable product.

Service-oriented Architecture

Service-oriented architecture (SoA) is a high-level approach that looks at how the entire organization approaches its software needs.

SoA focuses on the development, use, and reuse of small self-contained blocks of code (called *services*).

If adopted, all software is developed and managed so it could be reused. RAD, XP or agile methodologies would be used for this type of development.

Service-oriented Architecture

Sometimes ***Service-oriented architecture (SoA)*** is described as having a “holistic approach”.

Holistic is a word applied to many situations

In medicine, a holistic approach would mean (for example in cancer treatment) looking at the cancer drugs

+ the patients emotional confidence

+ family support, etc.

SELFSOURCING

Selfsourcing (end-user development) is development and support of IT systems by end users with little or no help from IT specialists.

- This do-it-yourself systems development approach must align with organizational goals.
- Some support from IT specialists may be needed.
- Technical documentation must be written.
- Ongoing user support must be provided.

SELFSOURCING

The major tools for Sourcing (end-user ***development***) are

Spreadsheets

Which can be as simple as basic Excel charts customized for your business

Database Management Systems

Such as Contact Management Databases used by Real Estate Agents

SELSOURCING

The major tools for Selfsourcing (end-user ***development***) are

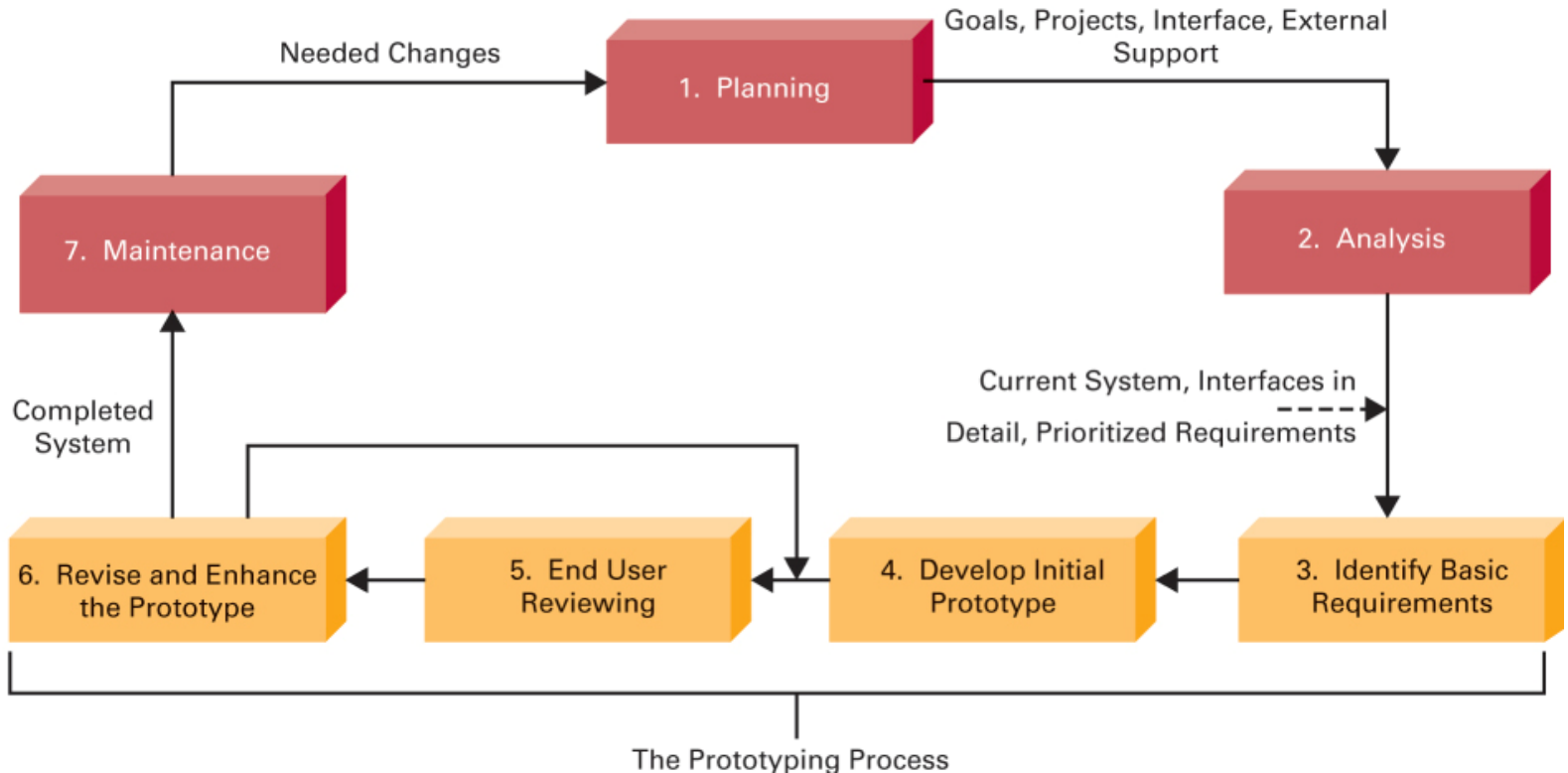
Web Development

Buying your own Domain Name

Arranging hosting

Using WYSIWYG software for creating and uploading your own content

Selfsourcing Approach



Advantages of Self-sourcing

- Improves requirements determination
- Increases end user participation and sense of ownership
- Increases speed of systems development
- Reduces ***invisible backlog*** (undertaking system development projects that have not been funded due to a lack of resources)

Pitfalls and Risks of Self-sourcing

- Inadequate end user expertise leads to inadequately developed systems.
- Lack of organizational focus creates “privatized” IT systems.
- Insufficient analysis of design alternatives leads to IT systems which are below par.
- Lack of documentation and external support leads to short-lived systems.
- Lack of consideration for security leads to potential gaps in protection.

Prototyping

A ***prototype*** is a model of a proposed product, service, or system. For example, architects make prototypes to show a building's design. During systems development, prototypes are used to:

- gather system requirements

- help clarify requirements for a new system

- show that a system is technically feasible

- (proof-of-concept prototype)***

- convince people of the worth of a proposed system ***(selling prototype)***.

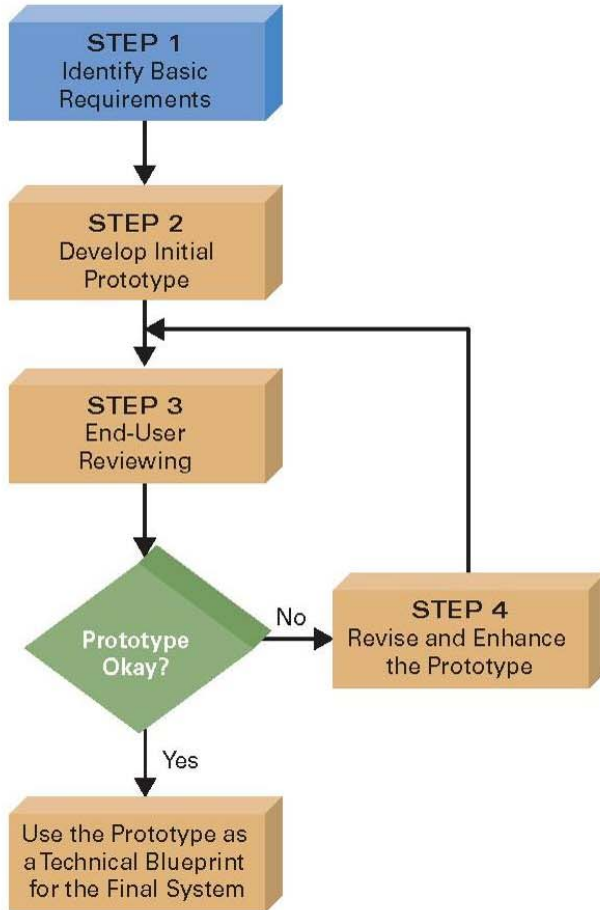
The Prototyping Process

The prototyping process always involves four steps:

1. Identify basic requirements
2. Develop initial prototype
3. Let the end-user review it
4. Revise and enhance the prototype

The Prototyping Process

INSOURCING PROTOTYPING



SELF-SOURCING PROTOTYPING

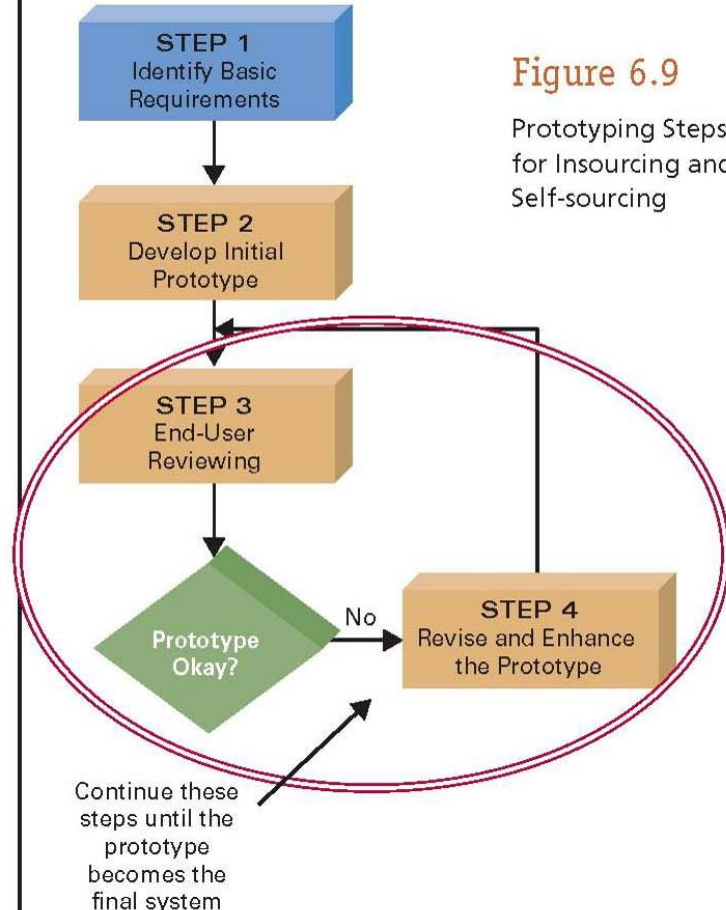


Figure 6.9

Prototyping Steps for Insourcing and Self-sourcing

Insourcing Prototyping is different from Self-Sourcing Prototyping

- When insourcing, IT specialists may be using the traditional SDLC.
- The finished prototype is used as a blueprint to illustrate the technical design .
- When self-sourcing, the prototype eventually becomes the final system.

Advantages of Prototyping

- Encourages end-user participation
- Helps resolve discrepancies among end-users
- Gives end-users a feel for the final system
- Helps determine technical feasibility
- Helps sell the proposed system

Disadvantages of Prototyping

- Leads people to believe that the new system will soon be complete
- Gives no indication of how the system will perform under real operational conditions
- Leads the project team to forgo proper testing and documentation

Outsourcing

Outsourcing is the delegation of specified work to a third party for a specified length of time, at a specified cost, and at a specified level of service.

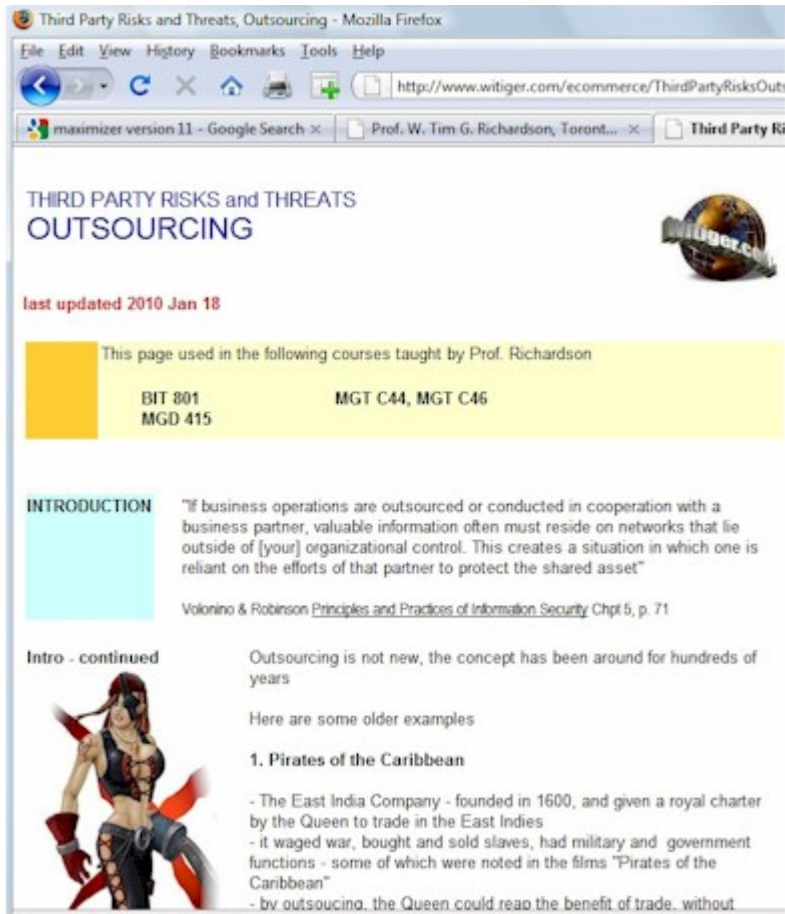
Outsourcing has become popular because of:

- Globalization
- The Internet (Technological Environment)
- Changes in the economy (Economic Environment)
- Technological advances
- Deregulation (Political Environment)

Outsourcing

Outsourcing is not new

<http://www.witiger.com/ecommerce/ThirdPartyRisksOutsourcing.htm>




Third Party Risks and Threats, Outsourcing - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://www.witiger.com/ecommerce/ThirdPartyRisksOutsourcing.htm

maximizer version 11 - Google Search Prof. W. Tim G. Richardson, Toronto... Third Party Ri

THIRD PARTY RISKS and THREATS OUTSOURCING



last updated 2010 Jan 18

This page used in the following courses taught by Prof. Richardson

BIT 801 MGD 415	MGT C44, MGT C46
--------------------	------------------

INTRODUCTION

"If business operations are outsourced or conducted in cooperation with a business partner, valuable information often must reside on networks that lie outside of [your] organizational control. This creates a situation in which one is reliant on the efforts of that partner to protect the shared asset"


Volonino & Robinson *Principles and Practices of Information Security* Chpt 5, p. 71

Intro - continued

Outsourcing is not new, the concept has been around for hundreds of years

Here are some older examples

- 1. Pirates of the Caribbean**
 - The East India Company - founded in 1600, and given a royal charter by the Queen to trade in the East Indies
 - it waged war, bought and sold slaves, had military and government functions - some of which were noted in the films "Pirates of the Caribbean"
 - by outsourcing, the Queen could reap the benefit of trade without



Outsourcing Options

IT outsourcing for software development can take one of four forms:

1. Purchase existing software.
2. Purchase existing software and pay the publisher to make certain modifications.
3. Purchase existing software and pay the publisher for the right to make modifications yourself.
4. Outsource the development of an entirely new and unique system for which no software exists.

Outsourcing Options

1. Purchase existing software.

2. Purchase existing software and pay publisher to make certain modifications.



4. Outsource the development of an entirely new and unique system for which no software exists.

3. Purchase software and pay publisher for the right to make changes yourself.

Outsourcing Process

Before outsourcing, the business must perform the analysis phase and determine the business requirements for the new system.

A request for proposal must be prepared.

This document describes the detailed requirements of the new system and invites companies to bid on the development work.

Bids are then evaluated and one vendor is chosen.

Outsourcing Process

RFPs *Federal Government*

The screenshot shows a Mozilla Firefox browser window with the address bar displaying <http://www.rhdcc-hrsc.gc.ca/eng/cs/comm/news/2000>. The page title is "The Government of Canada issues a Request for Proposal for the administration of Canada Student Loans". The website is the Human Resources and Skills Development Canada (HRSDC) website, with the URL www.hrsdc.gc.ca. The page features a navigation menu with links for "Français", "Home", "Contact Us", "Help", "Search", and "canada.gc.ca". The main content area displays the following text:

The Government of Canada issues a Request for Proposal for the administration of Canada Student Loans

news release

The Government of Canada issues a Request for Proposal for the administration of Canada Student Loans

September 14, 2000
FOR IMMEDIATE RELEASE
00-63

OTTAWA, ONTARIO - The Government of Canada today issued a Request for Proposal (RFP) to select one or more service providers to administer and manage Canada Student Loans. In this role, service providers will be responsible for:

- maintaining contact with borrowers;
- establishing repayment schedules;
- receiving payments; and

OTTAWA AND MONTREAL

Outsourcing – SLA

- Once the vendor is selected, a ***service level agreement (SLA)*** becomes the contract between the two parties.
- Technical specifications are in a supporting document called a ***service level specification (SLS)*** or ***service level objective (SLO)***.
- The users must test the new system before accepting it.

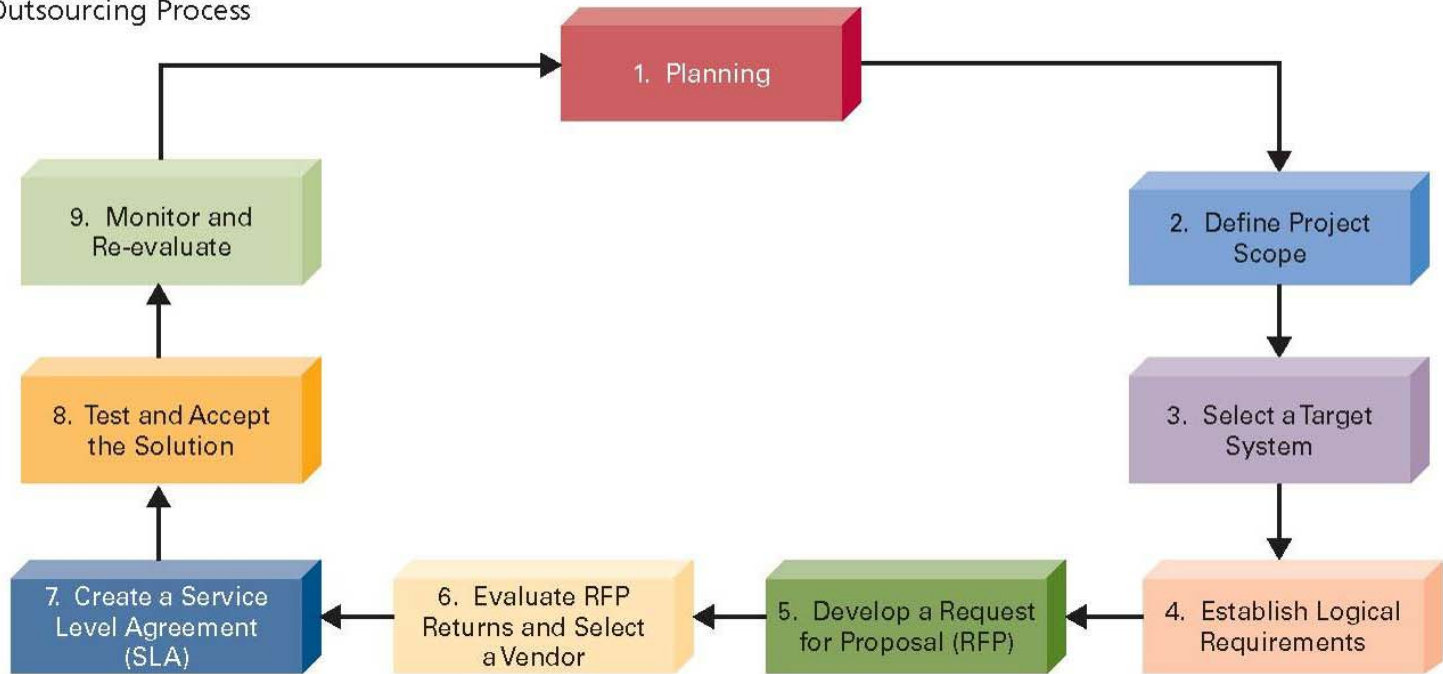
Outsourcing – SLA

- The new system must be monitored and evaluated repeatedly.
- If this is the vendor's responsibility, this must be specified in the SLA.

Outsourcing Process

Figure 6.11

The Outsourcing Process



When outsourcing, develop two vitally important documents:
a request for proposal and a service level agreement.

Geopolitical Outsourcing Options

There are three different forms of outsourcing:

- 1. Onshore outsourcing*** (within the same country e.g. Canada)
- 2. Nearshore outsourcing*** (with a company in a nearby country e.g. USA)
- 3. Offshore outsourcing*** (with a company that is geographically far away e.g. India)

Advantages of Outsourcing

An organization may benefit from outsourcing to:

- Focus on unique core competencies
- Exploit the intellect of another organization
- Better predict future costs
- Acquire leading-edge technology
- Reduce costs
- Improve performance accountability

Disadvantages of Outsourcing

Outsourcing is not a good option if it

- Reduces technical know-how for future innovation
- Reduces degree of control
- Increases vulnerability of your strategic information
- Increases dependency on other organizations.

***In this chapter, the following topics
were covered:***

- LO1 Systems development life cycle and the phases in it
- LO2 Component-based development methodologies
- LO3 Self-sourcing
- LO4 Prototypes
- LO5 Outsourcing