COMP1010 – Week
Development Strategies and Software Quality

Dr. Raymond Chiong, Dr. Mira Park, Dr. Mark Wallis

COMP1010 – Introduction to Computing
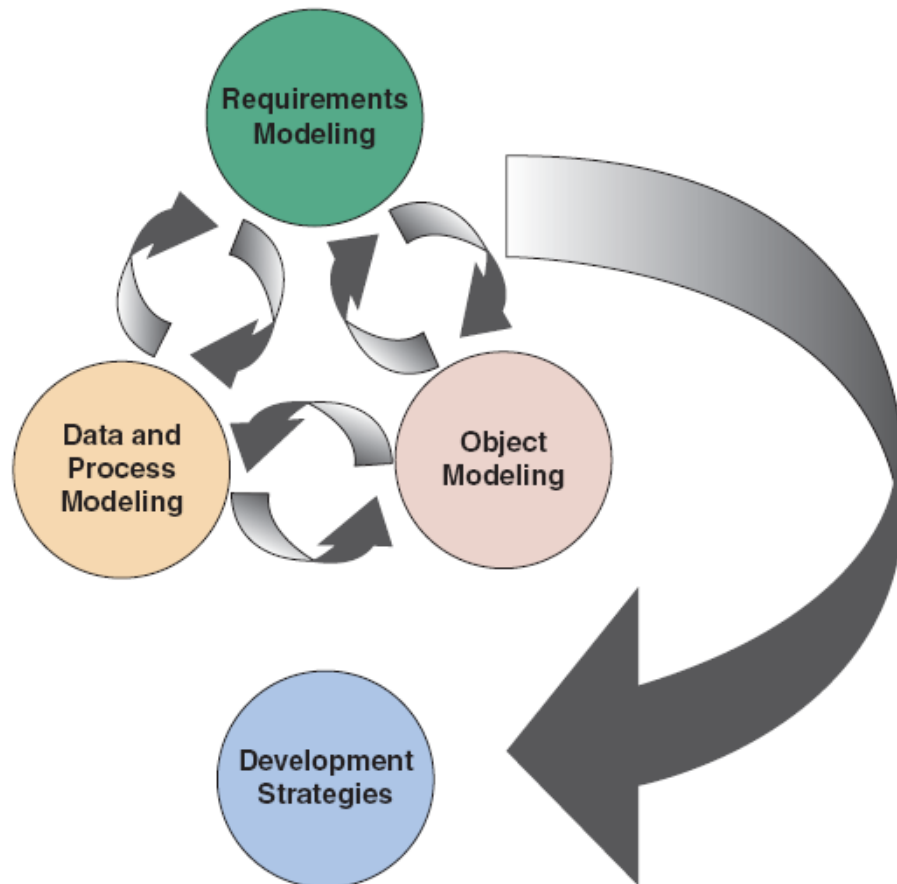
University of Newcastle

Development Strategy

THE UNIVERSITY OF
NEWCASTLE
AUSTRALIA

Systems Analysis Phase Overview

Systems Analysis Phase Tasks

- Data and process modeling
  - Graphically represents system data and processes
- Object modeling
  - Involves creation of objects to represent people, things, transactions, and events
- Development strategies
  - Include software trends, development alternatives, and outsourcing

THE UNIVERSITY OF
NEWCASTLE
AUSTRALIA

Overview

- Describe the concept of Software as a Service
- Define Web 2.0 and cloud computing
- Explain software acquisition alternatives, including traditional and Web-based software development strategies
- Describe software outsourcing options, including offshore outsourcing and the role of service providers
- Explain advantages and disadvantages of in-house software development
- Describe the system requirements document
- Explain the transition from systems analysis to systems design
- Describe the improvement of Software Quality

THE UNIVERSITY OF
NEWCASTLE
AUSTRALIA

Development Strategies Overview

- Earlier, certain work functions in the company required:
  - Development of software by in-house efforts
  - Employing the services of external entities
- Today, organizations have following choices for software acquisition
  - Application service providers
  - Web-hosted software options
  - Firms that offer enterprise-wide software solutions
- Selecting the best development path is an important decision

THE UNIVERSITY OF
NEWCASTLE
AUSTRALIA

The Impact of the Internet

- **Software as a Service (SaaS)**
  - Software deployment model that hosts an application as a service provided to customers over the Internet
  - Reduces the customer's need for software maintenance, operation, and support

THE UNIVERSITY OF
NEWCASTLE
AUSTRALIA

The Impact of the Internet (Cont.1)

- **Traditional vs. Web-Based Systems Development**
  - **Service-oriented architecture (SOA)**
    - A way of engineering systems in which reusable business functionality is provided by services through well-defined interfaces
    - Technically, not software architecture but an architectural style

THE UNIVERSITY OF
NEWCASTLE
AUSTRALIA

The Impact of the Internet (Cont.2)

- **Traditional vs. Web-Based Systems Development**
  - Traditional Development
    - System design is influenced by compatibility issues
    - Systems are designed to run on local and wide-area networks
    - Systems often utilize Internet links and resources
    - Development typically follows one of three main paths:
      - In-house development
      - Purchase of a software package with possible modification
      - Use of outside consultants
    - Scalability is affected by network limitations and constraints

THE UNIVERSITY OF
NEWCASTLE
AUSTRALIA

The Impact of the Internet (Cont.3)

- **Traditional vs. Web-Based Systems Development (Cont.)**
  - Web-Based Development
    - Systems are developed and delivered on an Internet-based framework
    - Treats the Web as the platform rather than just a communication channel
    - Web-based systems are easily scalable and can run on multiple hardware environments
    - Used for customer relationship management, order processing, and materials management
    - Treats software applications as services that are less dependent on desktop computing power and resources

THE UNIVERSITY OF
NEWCASTLE
AUSTRALIA

The Impact of the Internet (Cont.4)

- **Traditional vs. Web-Based Systems Development (Cont.)**
  - Web-Based Development
    - Requires additional layers, called **middleware**, to communicate with existing software and legacy systems
      - **Middleware:** Connects dissimilar applications and enables them to communicate and exchange data
    - Open more complex security issues that should be addressed

THE UNIVERSITY OF
NEWCASTLE
AUSTRALIA

The Impact of the Internet (Cont.5)

- **Evolving Trends - Web 2.0, Cloud Computing, and Mobile Devices**
  - **Web 2.0**: second generation of the web that enables people to collaborate, interact, and share information much more effectively
    - Enhances interactive experiences
  - **Cloud computing**: Online software in which applications and services are accessed and used through an Internet connection
  - **Mobile devices:** Smartphones, tablets, and other computing devices that are not permanently tethered to a desk

THE UNIVERSITY OF
NEWCASTLE
AUSTRALIA

Outsourcing

- Transfer of information systems development, operation, or maintenance to an outside firm

- **The Growth of Outsourcing**

  - **Service provider:** Offers outsourcing solutions **Application service provider (ASP)**

    - Delivers a software application or access to an application by charging a usage or subscription fee

  - **Internet business services (IBS)**

    - Also called **managed hosting**

    - Provide web-based support for transactions

THE UNIVERSITY OF
NEWCASTLE
AUSTRALIA

Outsourcing (Cont.)

- **Offshore Outsourcing**
  - Called **global outsourcing**
  - Shifting IT development, support, and operations to other countries
  - Reason - Lower bottom-line costs
  - Risks and concerns
    - Impact on the economy
    - Project control
    - Security issues

THE UNIVERSITY OF
NEWCASTLE
AUSTRALIA

In-House Software Development Options

- Software development options
  - Develop own systems
  - Purchase, possibly customize, and implement a software package
- Most important consideration is the total cost of ownership (TCO)
- Companies can develop user applications based on commercial software packages

THE UNIVERSITY OF
NEWCASTLE
AUSTRALIA

In-House Software Development Options
(Cont.1)

- **Make or Buy Decision**
  - Refers to the choice between developing and purchasing
  - A company's IT department makes, builds, and develops **in-house software**
  - A **software package** is obtained from a vendor or application service provider
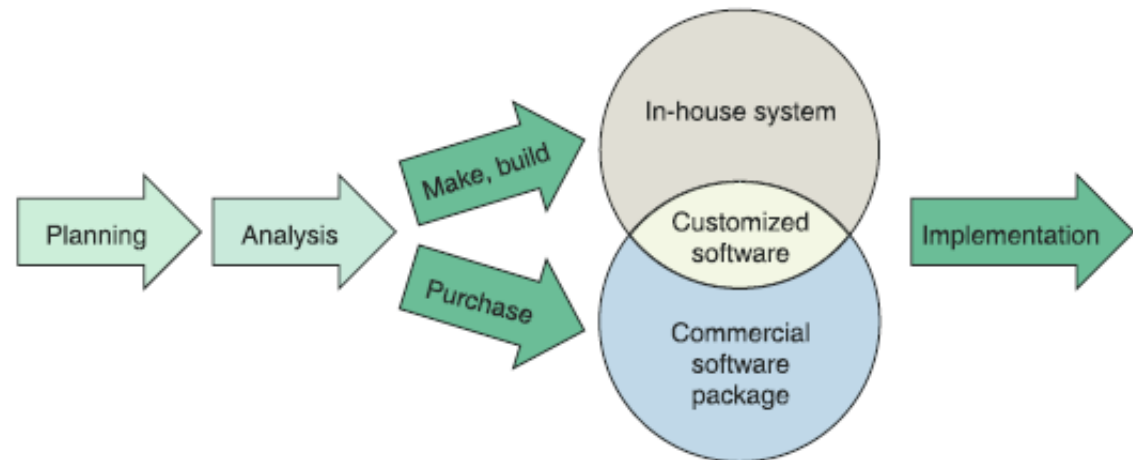
In-House Software Development Options
(Cont.2)

- **Make or Buy Decision** (cont.)
  - **Software package:** Obtained from a vendor or application service provider
  - **Software vendors**: Develop software for sale
  - **Value-added reseller (VAR):** Enhances a commercial package by adding custom features and configuring it for a particular industry
  - **Horizontal application:** Can be used by many different types of organizations
  - **Vertical application**: Developed to handle information requirements for a specific type of business

THE UNIVERSITY OF
NEWCASTLE
AUSTRALIA

In-House Software Development Options
(Cont.3)

REASONS FOR IN-HOUSE DEVELOPMENT	REASONS FOR PURCHASING A SOFTWARE PACKAGE
Satisfy unique business requirements	Lower costs
Minimize changes in business procedures and policies	Requires less time to implement
Meet constraints of existing systems	Proven reliability and performance benchmarks
Meet constraints of existing technology	Requires less technical development staff
Develop internal resources and capabilities	Future upgrades provided by the vendor
Satisfy unique security requirements	Obtain input from other companies

THE UNIVERSITY OF
NEWCASTLE
AUSTRALIA

In-House Software Development Options
(Cont.4)

- ## Customizing a Software Package
  - Purchase a basic package that vendors will customize to suit project requirements
  - Negotiate directly with the software vendor to make enhancements to meet project needs by paying for the changes
  - Purchase the package and make project-specific modifications
    - Ensure modifications are permissible under the terms of the software license

THE UNIVERSITY OF
NEWCASTLE
AUSTRALIA

In-House Software Development Options
(Cont.5)

- **Creating User Applications**
  - **User application:** Utilizes standard business software
  - **User interface:** Enables effective interaction with the application
  - **Service desk** or **information center (IC):** Provides user support
  - **Screen generators** and **report generators:** Allow users to design their own data entry forms and reports
  - Appropriate controls must be provided to ensure data security and integrity

THE UNIVERSITY OF
NEWCASTLE
AUSTRALIA

In-House Software Development Options
(Cont.6)

The Systems Analyst's Role

- Based on decisions taken by the organization in the systems development process
  - Current and future needs are considered
- Evaluation and selection of alternatives is a complicated process
  - Forecasting actual costs is difficult
- **Evaluation and selection team:** Selects hardware and software, includes systems analysts and users
  - Ensures that critical factors are not overlooked and that a sound choice is made

THE UNIVERSITY OF
NEWCASTLE
AUSTRALIA

The Software Acquisition Process

- **Step 1 - Evaluate the Information System Requirements**
    - Identify key features
    - Consider network and Web-related issues
    - Estimate volume and future growth
    - Specify hardware, software, or personnel constraints
    - Prepare a request for proposal or quotation
        - **Request for proposal (RFP)**: Describes the company, lists the IT services or products needed, and specifies the features required
        - **Request for quotation (RFQ)**: more specific than an RFP

THE UNIVERSITY OF
NEWCASTLE
AUSTRALIA

The Software Acquisition Process (Cont.1)

- **Step 2 - Identify Potential Vendors or Outsourcing Options**
  - The Internet contains information on all major products and acquisition services
  - The organization can avail the services of a consulting firm that help companies select software packages
  - Online **forums or newsgroups** provide opinions and ideas
    - Google Groups
    - Yahoo Groups

THE UNIVERSITY OF
NEWCASTLE
AUSTRALIA

The Software Acquisition Process (Cont.2)

- **Step 3 - Evaluate the Alternatives**
  - Existing users
    - Provide feedback about their experiences
  - Application testing
    - Users in the organization may be able to test the product
  - Benchmarking
    - **Benchmark**: Measures the time a package takes to process a certain number of transactions
    - Each package is matched against RFP features and the choices are ranked

THE UNIVERSITY OF
NEWCASTLE
AUSTRALIA

The Software Acquisition Process (Cont.3)

- **Step 4 - Perform Cost-Benefit Analysis**
  - Identify and calculate total cost of ownership (TCO) for each option being considered
  - Study the conditions of use that come along with the **software license**
  - If a software package is purchased, consider a supplemental **maintenance agreement**

THE UNIVERSITY OF
NEWCASTLE
AUSTRALIA

The Software Acquisition Process (Cont.4)

- **Step 5 - Prepare a Recommendation**
  - Evaluate and describe alternatives along with:
    - Costs
    - Benefits
    - Advantages
    - Disadvantages
  - Submit a formal system requirements document and deliver a presentation

THE UNIVERSITY OF
NEWCASTLE
AUSTRALIA

The Software Acquisition Process (Cont.5)

- **Step 6 - Implement the Solution**
  - Implementation tasks will depend on the solution selected
  - Before the new software becomes operational, complete all implementation steps
    - Loading
    - Configuring and testing the software
    - Training users
    - Converting data files to the new system's format

THE UNIVERSITY OF
NEWCASTLE
AUSTRALIA

Completion of Systems Analysis Tasks

- **System Requirements Document**
  - Called **software requirements specification**
  - Contains the requirements for the new system
  - Describes the alternatives considered
  - Makes a specific recommendation to management
  - Similar to a contract
    - Identifies items that system developers must deliver to users
  - Format and organize the systems document
    - Easy to read and use

THE UNIVERSITY OF
NEWCASTLE
AUSTRALIA

Completion of Systems Analysis Tasks
(Cont.1)

- **Presentation to Management**
  - Helps take key decisions that affect the future development of the system
  - Suggestions for effective presentations
    - Start with a brief overview
    - Summarize the primary viable alternatives
    - Explain why the evaluation and selection team chose the recommended alternative
    - Allow time for discussion
    - Obtain a final decision from management or agree on a timetable for the next step in the process

THE UNIVERSITY OF
NEWCASTLE
AUSTRALIA

Completion of Systems Analysis Tasks
(Cont.2)

- **Presentation to Management** (Cont.)
  - Depending on management's decision, a systems analyst will do one of the following
    - Implement an outsourcing alternative
    - Develop an in-house system
    - Purchase or customize a software package
    - Perform additional systems analysis work
    - Stop all further work
  - Post presentation and management decision, the project begins a transition to the systems phase of the SDLC

THE UNIVERSITY OF
NEWCASTLE
AUSTRALIA

Transition to System Design

- **Preparing for Systems Design**
  - Systems design requires accurate documentation
    - Provide detailed specifications for output, input, data, processes, and other requirements

▸ **Logical and Physical Design**

  ○ **Logical design:** Defines what must take place

  ○ **Physical design:** Describes the actual process of entering, verifying, and storing data

  ○ Logical and physical designs are closely related
    - Accurate systems analysis is required

THE UNIVERSITY OF
NEWCASTLE
AUSTRALIA

Software Quality

THE UNIVERSITY OF
NEWCASTLE
AUSTRALIA

Software Quality

High-quality software systems

- Easy to learn and use because they perform quickly and efficiently, meet their users' needs, and operate safely and reliably so that system downtime is kept to a minimum

Software defect

- Error that, if not removed, could cause a software system to fail to meet its users' needs

Software quality

- Degree to which a software product meets the needs of its users

Quality management

- Defining, measuring, and refining the quality of the development process and the products developed
  - **Deliverables**: Products of quality management

THE UNIVERSITY OF
NEWCASTLE
AUSTRALIA

Causes of Poor Software Quality

- Developers must define and follow a set of software engineering principles
    - Be committed to learning from past mistakes
- Developers must understand the environment in which their systems will operate
    - Design systems that are immune to human error
- Extreme pressure that software companies feel to reduce the time to market for their products
    - Resources needed to ensure quality are cut under the pressure to ship a new product

THE UNIVERSITY OF
NEWCASTLE
AUSTRALIA

Importance of Software Quality

- **Business information system**: Set of interrelated components that collects and processes data and disseminates the output
  - **Decision support system (DSS)**
- Controls industrial processes and the operation of industrial and consumer products
- Mismanaged software can be fatal to a business
  - Miss product deadlines, increased product development costs, and delivery of low quality products

THE UNIVERSITY OF
NEWCASTLE
AUSTRALIA

Dynamic Testing

- **Dynamic testing**: Entering test data and comparing the results with the expected results in a process
  - **Black-box testing**: Viewing the software unit as a device that has expected input and output behaviors but whose internal workings are unknown
    - If the unit demonstrates the expected behaviors for all the input data in the test suite, it passes the test
  - **White-box testing**: Treats the software unit as a device that has expected input and output behaviors but whose internal workings are known

THE UNIVERSITY OF
NEWCASTLE
AUSTRALIA

Types of Software Testing

Unit testing

Integration testing

System testing

User acceptance testing

THE UNIVERSITY OF
NEWCASTLE
AUSTRALIA

Unit testing

- A software development process in which the smallest testable parts of an application

- It aims at testing each of the components that a system are built upon.

- Components that are erroneous can be detected earlier

Integration testing

- After successful unit testing, the software units are combined into an integrated subsystem that undergoes rigorous testing to ensure that the linkages among the various subsystems work successfully

THE UNIVERSITY OF
NEWCASTLE
AUSTRALIA

System testing

- After successful integration testing, the various subsystem are combined to test the entire system as a complete entity.

User acceptance testing

- Independent testing is performed by trained end user to ensure that the system operates as they expect

THE UNIVERSITY OF
NEWCASTLE
AUSTRALIA

Reliability and Safety in Safety-Critical Systems

- **Reliability**: Measure of the rate of failure in a system that would render it unusable over its expected lifetime
  - Capability of the system to continue to perform
- Safety - Ability of the system to perform in a safe manner
- System-human interface - Important and difficult areas of safety-critical system design
  - Design of the system should not allow for erroneous judgment on the part of the operator

THE UNIVERSITY OF
NEWCASTLE
AUSTRALIA

Quality Management Standards

- **ISO 9001 family of standards**
  - Guide to quality products, services, and management
  - Organization must submit to an examination by an external assessor to obtain the certificate

- **Failure mode and effects analysis (FMEA)**
  - Used to develop ISO 9001-compliant quality systems
  - By evaluating reliability and determining the effects of system and equipment failures
  - **Failure mode**: Describes how a product or process could fail to perform the desired functions described by the customer

THE UNIVERSITY OF
NEWCASTLE
AUSTRALIA

Summary

- A new trend views Software as a Service (SaaS), rather than a product

- Traditional systems must:
  - Function in various hardware and software environments
  - Be compatible with legacy systems
  - Operate within the constraints of company networks and desktop computing capability

- Companies that choose to handle their own software development needs can:
  - Create in-house systems
  - Commercially purchase software packages

THE UNIVERSITY OF
NEWCASTLE
AUSTRALIA

Summary (Cont. 1)

- The systems analyst's role in the software development process depends on the specific development strategy

- Acquiring software involves a series of specific steps

- The system requirements document is the deliverable, or end product, of the systems analysis phase

THE UNIVERSITY OF
NEWCASTLE
AUSTRALIA

Summary (Cont. 2)

- Demand for high-quality software is increasing

- Developers are under extreme pressure to reduce time to market of products

- ISO 9001 standard is a guide to quality products, services, and management

- Failure mode and effects analysis (FMEA) is an important technique used to develop ISO 9001-compliant quality systems

THE UNIVERSITY OF
NEWCASTLE
AUSTRALIA