FOX | ITACS
Master of IT Auditing & Cyber Security

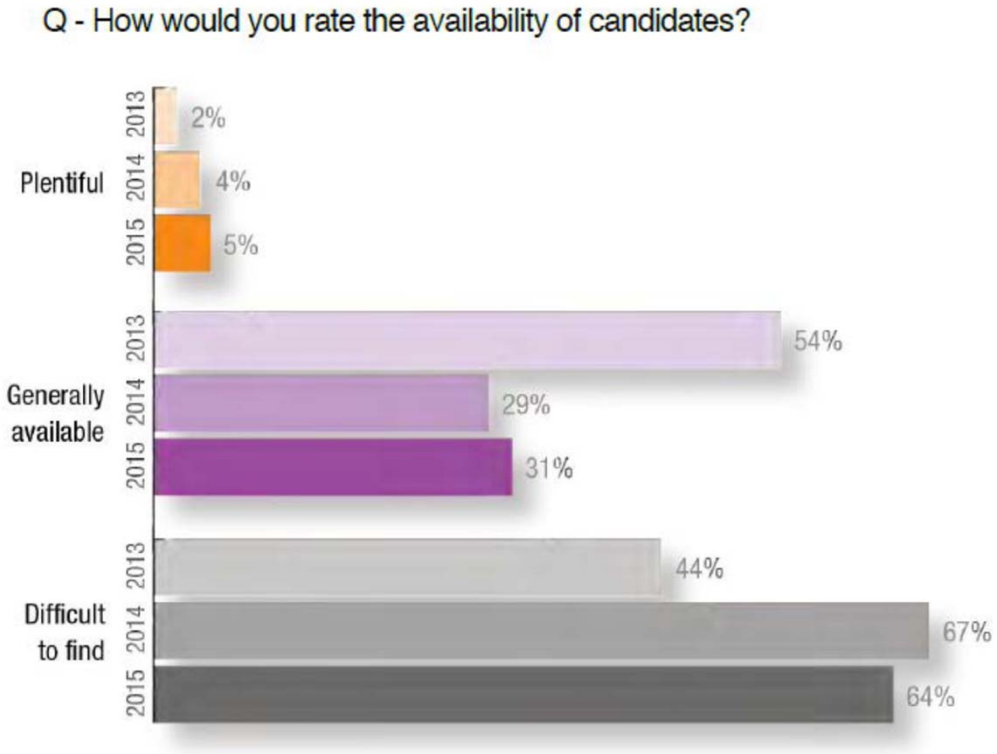
Systems and Infrastructure Lifecycle Management

Prof. Mike Romeu

January 18, 2018

State of the Market*

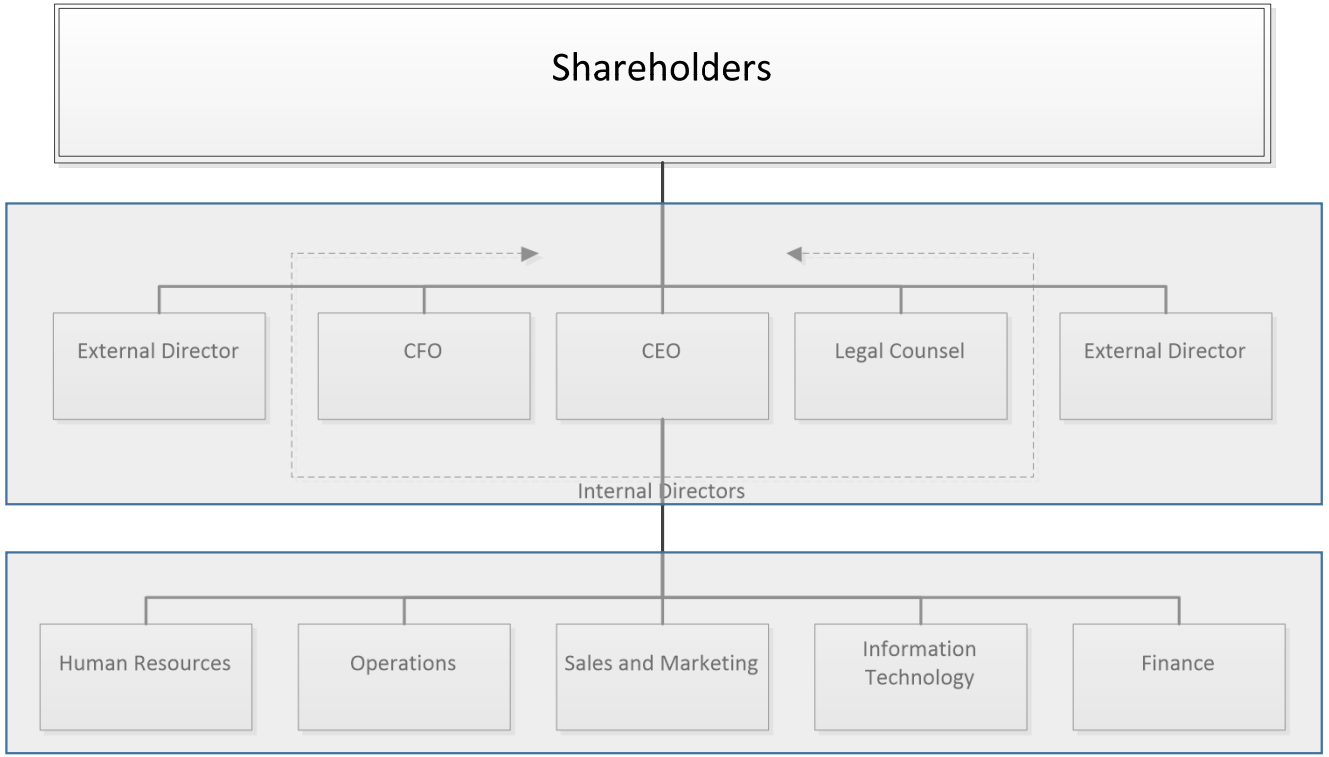
- Only 47% believe internal audit department is “sufficiently resourced”
- 51% report that technical skills is the greatest challenge for sourcing
- 64% report internal auditors difficult to find; 71% report it is difficult to recruit



*Barclay Simpson Governance Recruitment Market Report 2016: Internal Audit.

Corporate Structures

Governance Level:
- Board of Directors



Responsible for:
- Evaluate
- Direct
- Monitor

Management Level :
- Management

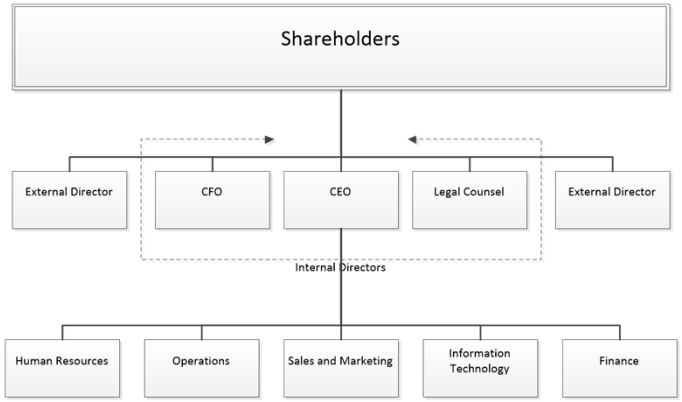
Responsible for:
- Plan
- Build
- Run
- Monitor

Corporate Governance

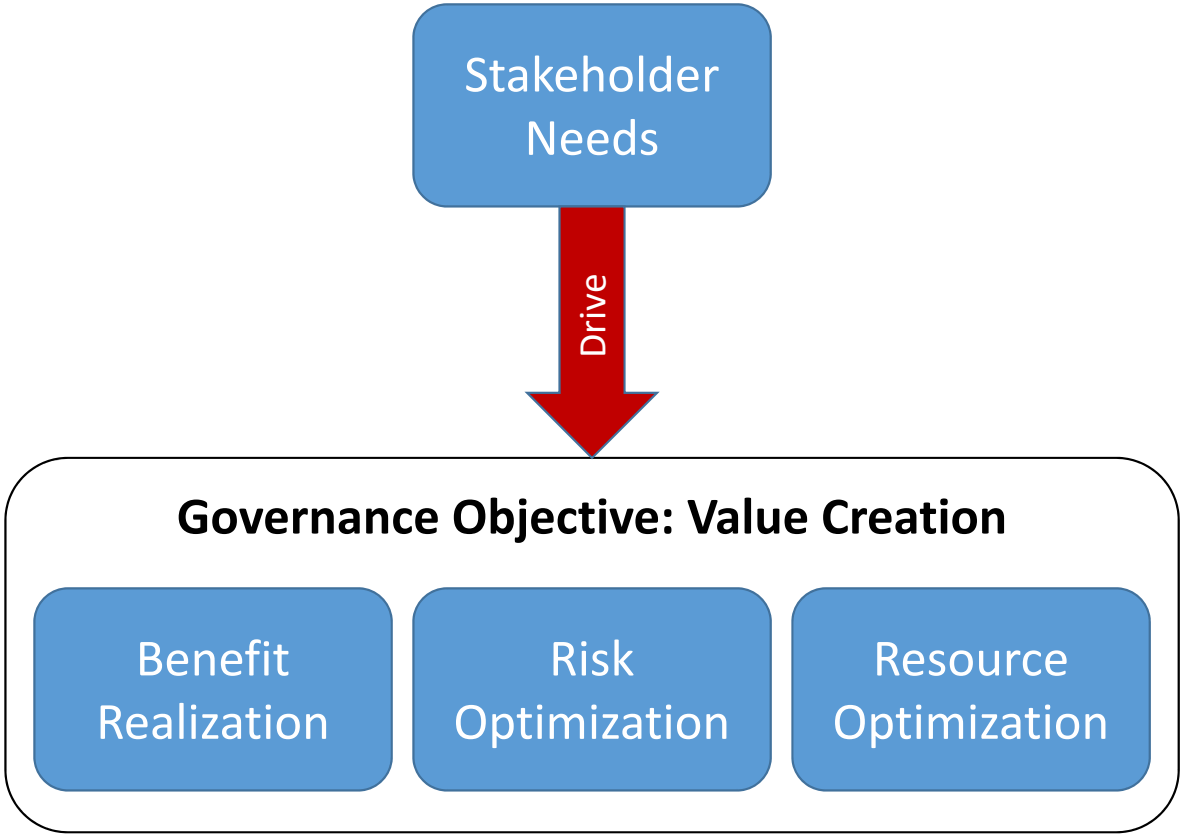
"Leadership, organizational structures and processes that ensure the enterprise sustains and extends strategies and objectives." (ISACA Glossary).

Corporate governance helps to:

- Set Strategic Direction to ensures goals and objectives are achievable
- Risks are properly addressed
- Resources (People, Time, Monies) are properly utilized



What is the Objective of Governance?



Balanced Scorecard

A strategic planning and management system used extensively to

- align business activities to the vision and strategy of the organization
- improve internal and external communications,
- monitor performance against strategic goals.” balancedscorecard.org

Enterprise Goals

COBIT 5 Enterprise Goals	
BSC Dimension	**Enterprise Goal**
Financial	1. Stakeholder value of business investments
	2. Portfolio of competitive products and services
	3. Managed business risk (safeguarding assets)
	4. Compliance with external laws and regulations
	5. Financial transparency
Customer	6. Customer-oriented service culture
	7. Business service continuity and availability
	8. Agie responses to changing business environment
	9. Information-based strategic decision making
	10. Optimisation of service delivery costs
Internal	11. Optimisation of business process functionality
	12. Optimisation of business process costs
	13. Managed business change programs
	14. Operational and staff productivity
	15. Compliance with internal policies
Learning and Growth	16 Skilled and motivated people
	17. Product and business innovation culture

COBIT 5 IT Goals	
BSC Dimension	**IT Goal**
Financial	1. Alignment of IT and business strategy
	2. IT Compliance and support for business compliance with external laws and regulations
	3. Commitment of executive management for making IT-related decisions
	4. Managed IT-related business risk
	5. Realised benefits from IT-enabled investments and services portfolio
	6. Transparency of IT costs, benefits and risk
Customer	7. Delivery of IT services in line with business requirements
	8. Adequate use of applications, information and technoloty solutions
Internal	9. IT Agility
	10. Security of information, processing infrastructure and applications
	11. Optimisation of IT assets, resources and capabilities
	12. Enablement and support of business processes by integrating applications and technology into business processes
	13. Delivery of programmes delivering benefits, on time, on budget, and meeting requirements and quality standards
	14. Availability of reliable and useful informatino for decision making
	15. IT compliance with internal policies
Learning and Growth	16. Competent and motivated business and IT personnel
	17. Knowledge, exprtise and initiatives for business innovation

Enterprise Goals

COBIT 5 Enterprise Goals	
BSC Dimension	**Enterprise Goal**
Financial	**1. Stakeholder value of business investments**
	2. Portfolio of competitive products and services
	3. Managed business risk (safeguarding assets)
	4. Compliance with external laws and regulations
	5. Financial transparency
Customer	6. Customer-oriented service culture
	7. Business service continuity and availability
	8. Agie responses to changing business environment
	9. Information-based strategic decision making
	10. Optimisation of service delivery costs
Internal	11. Optimisation of business process functionality
	12. Optimisation of business process costs
	13. Managed business change programs
	14. Operational and staff productivity
	15. Compliance with internal policies
Learning and Growth	16 Skilled and motivated people
	17. Product and business innovation culture

COBIT 5 Enterprise Goals	
BSC Dimension	**IT Goal**
Financial	**1. Alignment of IT and business strategy**
	2. IT Compliance and support for business compliance with external laws and regulations
	3. Commitment of executive management for making IT-related decisions
	4. Managed IT-related business risk
	5. Realised benefits from IT-enabled investments and services portfolio
	6. Transparency of IT costs, benefits and risk
Customer	7. Delivery of IT services in line with business requirements
	8. Adequate use of applications, information and technoloty solutions
Internal	9. IT Agility
	10. Security of information, processing infrastructure and applications
	11. Optimisation of IT assets, resources and capabilities
	12. Enablement and support of business processes by integrating applications and technology into business processes
	13. Delivery of programmes delivering benefits, on time, on budget, and meeting requirements and quality standards
	14. Availability of reliable and useful informatino for decision making
	15. IT compliance with internal policies
Learning and Growth	16. Competent and motivated business and IT personnel
	17. Knowledge, exprtise and initiatives for business innovation

Enterprise Goals

COBIT 5 Enterprise Goals	
BSC Dimension	**Enterprise Goal**
Financial	1. Stakeholder value of business investments
	2. Portfolio of competitive products and services
	3. Managed business risk (safeguarding assets)
	4. Compliance with external laws and regulations
	5. Financial transparency
Customer	**6. Customer-oriented service culture**
	7. Business service continuity and availability
	8. Agile responses to changing business environment
	9. Information-based strategic decision making
	10. Optimisation of service delivery costs
Internal	11. Optimisation of business process functionality
	12. Optimisation of business process costs
	13. Managed business change programs
	14. Operational and staff productivity
	15. Compliance with internal policies
Learning and Growth	16 Skilled and motivated people
	17. Product and business innovation culture

COBIT 5 Enterprise Goals	
BSC Dimension	**IT Goal**
Financial	1. Alignment of IT and business strategy
	2. IT Compliance and support for business compliance with external laws and regulations
	3. Commitment of executive management for making IT-related decisions
	4. Managed IT-related business risk
	5. Realised benefits from IT-enabled investments and services portfolio
	6. Transparency of IT costs, benefits and risk
Customer	**7. Delivery of IT services in line with business requirements**
	8. Adequate use of applications, information and technology solutions
Internal	9. IT Agility
	10. Security of information, processing infrastructure and applications
	11. Optimisation of IT assets, resources and capabilities
	12. Enablement and support of business processes by integrating applications and technology into business processes
	13. Delivery of programmes delivering benefits, on time, on budget, and meeting requirements and quality standards
	14. Availability of reliable and useful informatino for decision making
	15. IT compliance with internal policies
Learning and Growth	16. Competent and motivated business and IT personnel
	17. Knowledge, exprtise and initiatives for business innovation

Enterprise Goals

COBIT 5 Enterprise Goals	
BSC Dimension	**Enterprise Goal**
Financial	1. Stakeholder value of business investments
	2. Portfolio of competitive products and services
	3. Managed business risk (safeguarding assets)
	4. Compliance with external laws and regulations
	5. Financial transparency
Customer	6. Customer-oriented service culture
	7. Business service continuity and availability
	8. Agie responses to changing business environment
	9. Information-based strategic decision making
	10. Optimisation of service delivery costs
Internal	**11. Optimisation of business process functionality**
	12. Optimisation of business process costs
	13. Managed business change programs
	14. Operational and staff productivity
	15. Compliance with internal policies
Learning and Growth	16 Skilled and motivated people
	17. Product and business innovation culture

COBIT 5 Enterprise Goals	
BSC Dimension	**IT Goal**
Financial	1. Alignment of IT and business strategy
	2. IT Compliance and support for business compliance with external laws and regulations
	3. Commitment of executive management for making IT-related decisions
	4. Managed IT-related business risk
	5. Realised benefits from IT-enabled investments and services portfolio
	6. Transparency of IT costs, benefits and risk
Customer	7. Delivery of IT services in line with business requirements
	8. Adequate use of applications, information and technoloty solutions
Internal	**9. IT Agility**
	10. Security of information, processing infrastructure and applications
	11. Optimisation of IT assets, resources and capabilities
	12. Enablement and support of business processes by integrating applications and technology into business processes
	13. Delivery of programmes delivering benefits, on time, on budget, and meeting requirements and quality standards
	14. Availability of reliable and useful informatino for decision making
	15. IT compliance with internal policies
Learning and Growth	16. Competent and motivated business and IT personnel
	17. Knowledge, exprtise and initiatives for business innovation

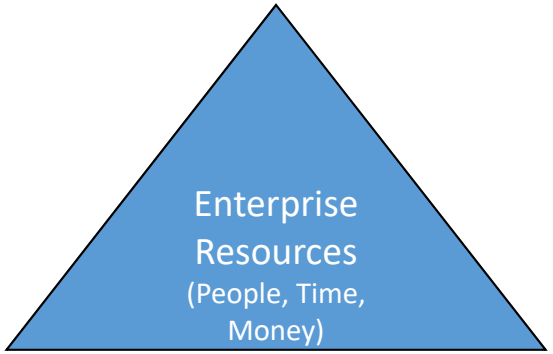
Enterprise Goals

COBIT 5 Enterprise Goals	
BSC Dimension	**Enterprise Goal**
Financial	1. Stakeholder value of business investments
	2. Portfolio of competitive products and services
	3. Managed business risk (safeguarding assets)
	4. Compliance with external laws and regulations
	5. Financial transparency
Customer	6. Customer-oriented service culture
	7. Business service continuity and availability
	8. Agie responses to changing business environment
	9. Information-based strategic decision making
	10. Optimisation of service delivery costs
Internal	11. Optimisation of business process functionality
	12. Optimisation of business process costs
	13. Managed business change programs
	14. Operational and staff productivity
	15. Compliance with internal policies
Learning and Growth	**16 Skilled and motivated people**
	17. Product and business innovation culture

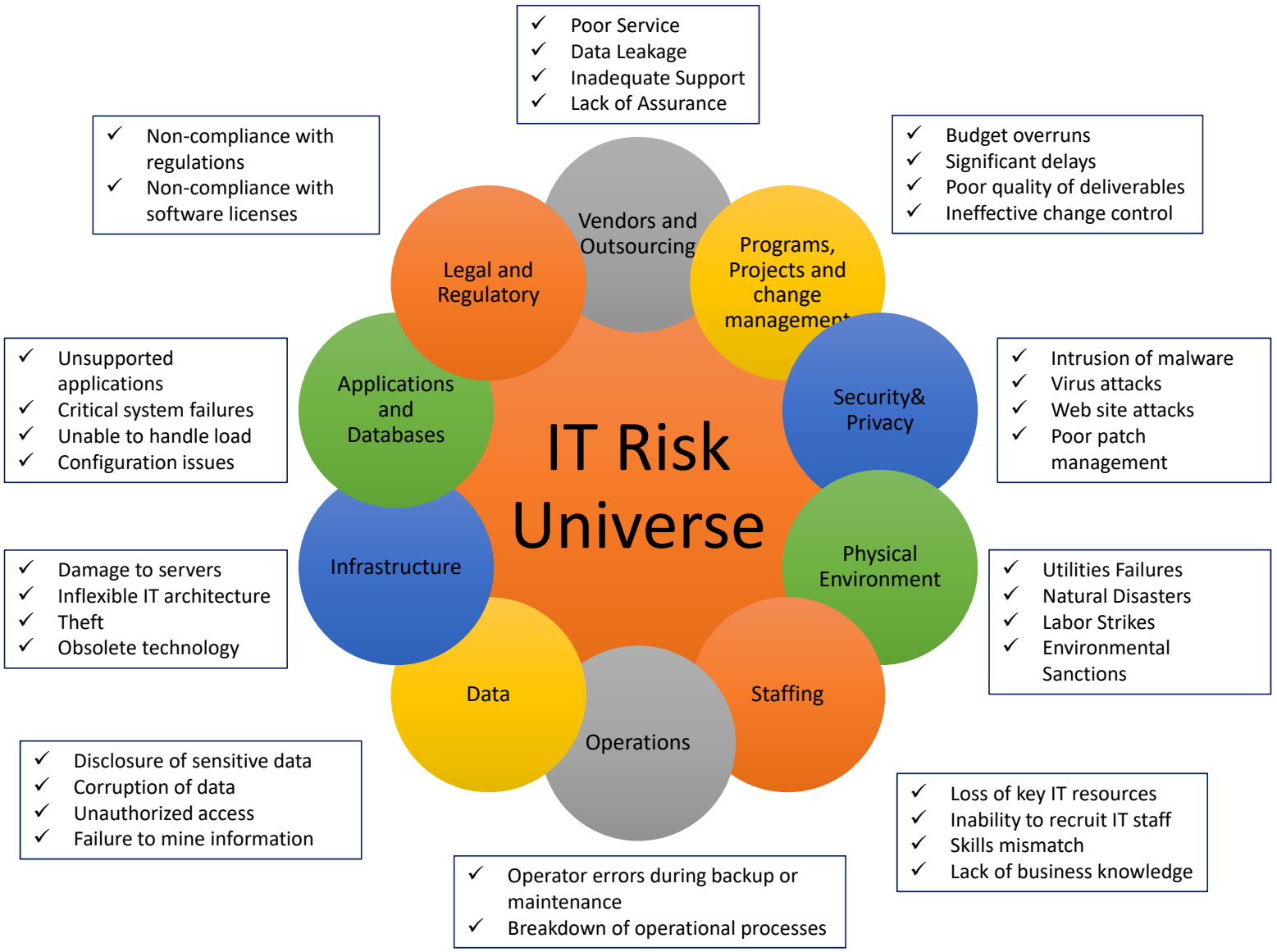
COBIT 5 Enterprise Goals	
BSC Dimension	**IT Goal**
Financial	1. Alignment of IT and business strategy
	2. IT Compliance and support for business compliance with external laws and regulations
	3. Commitment of executive management for making IT-related decisions
	4. Managed IT-related business risk
	5. Realised benefits from IT-enabled investments and services portfolio
	6. Transparency of IT costs, benefits and risk
Customer	7. Delivery of IT services in line with business requirements
	8. Adequate use of applications, information and technoloty solutions
Internal	9. IT Agility
	10. Security of information, processing infrastructure and applications
	11. Optimisation of IT assets, resources and capabilities
	12. Enablement and support of business processes by integrating applications and technology into business processes
	13. Delivery of programmes delivering benefits, on time, on budget, and meeting requirements and quality standards
	14. Availability of reliable and useful informatino for decision making
	15. IT compliance with internal policies
Learning and Growth	**16. Competent and motivated business and IT personnel**
	17. Knowledge, expertise and initiatives for business innovation

Creating Value is a Balancing Act!

- Market Risk
- Financial Risk
- Operations Risk
- Regulatory Risk



- Policies and Procedures
- Standards and Guidelines
- Laws and Regulations


IT Risk Universe
Vendors and Outsourcing
Programs, Projects and change management
Security& Privacy
Physical Environment
Staffing
Operations
Data
Infrastructure
Applications and Databases
Legal and Regulatory
✓ Poor Service
✓ Data Leakage
✓ Inadequate Support
✓ Lack of Assurance
✓ Budget overruns
✓ Significant delays
✓ Poor quality of deliverables
✓ Ineffective change control
✓ Intrusion of malware
✓ Virus attacks
✓ Web site attacks
✓ Poor patch management
✓ Utilities Failures
✓ Natural Disasters
✓ Labor Strikes
✓ Environmental Sanctions
✓ Loss of key IT resources
✓ Inability to recruit IT staff
✓ Skills mismatch
✓ Lack of business knowledge
✓ Operator errors during backup or maintenance
✓ Breakdown of operational processes
✓ Disclosure of sensitive data
✓ Corruption of data
✓ Unauthorized access
✓ Failure to mine information
✓ Damage to servers
✓ Inflexible IT architecture
✓ Theft
✓ Obsolete technology
✓ Unsupported applications
✓ Critical system failures
✓ Unable to handle load
✓ Configuration issues
✓ Non-compliance with regulations
✓ Non-compliance with software licenses

The Role of the Audit and Assurance Professional

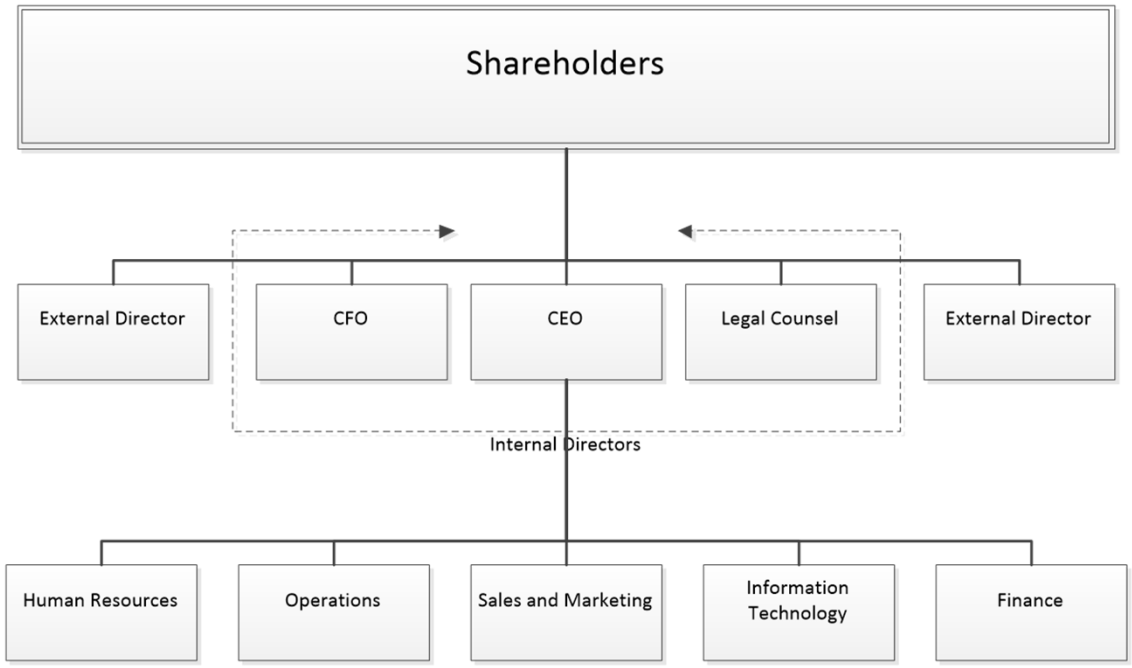
- **Audit** - Formal inspection and verification to check whether a standard or set of guidelines is being followed, records are accurate, or efficiency and effectiveness targets are being met.

- **Assurance** – pursuant to an accountability relationship between two or more parties, an IT audit and assurance professional may be engaged to issue a written communication expressing a conclusion about the subject matters to the accountable party.

- **Assurance engagements can include**:
  - support for audited financial statements,
  - reviews of controls,
  - compliance with required standards and practices, and
  - compliance with agreements, licenses, legislation and regulation.

Five Components of Audit and Assurance

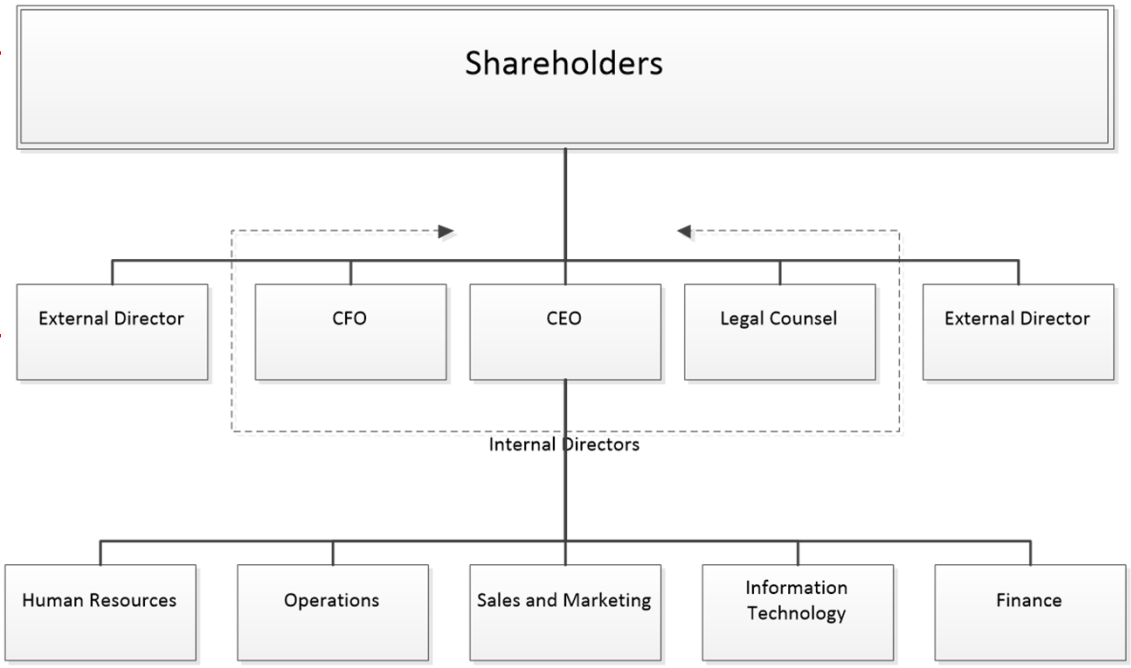
1. Three-party relationship, including:
   - The accountable party (auditee)
   - The user of the assurance report
   - Assurance professional



Five Components of Audit and Assurance

1. Three-party relationship, including:
   - The accountable party (auditee)
   - The user of the assurance report
   - Assurance professional

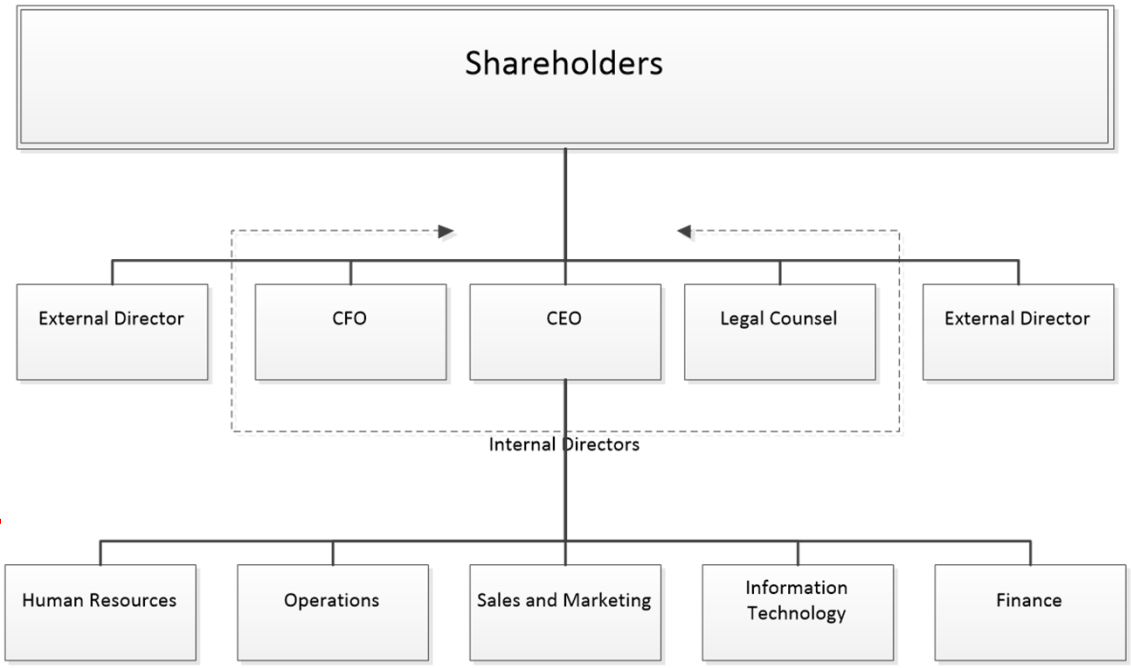
User of assurance report



Five Components of Audit and Assurance

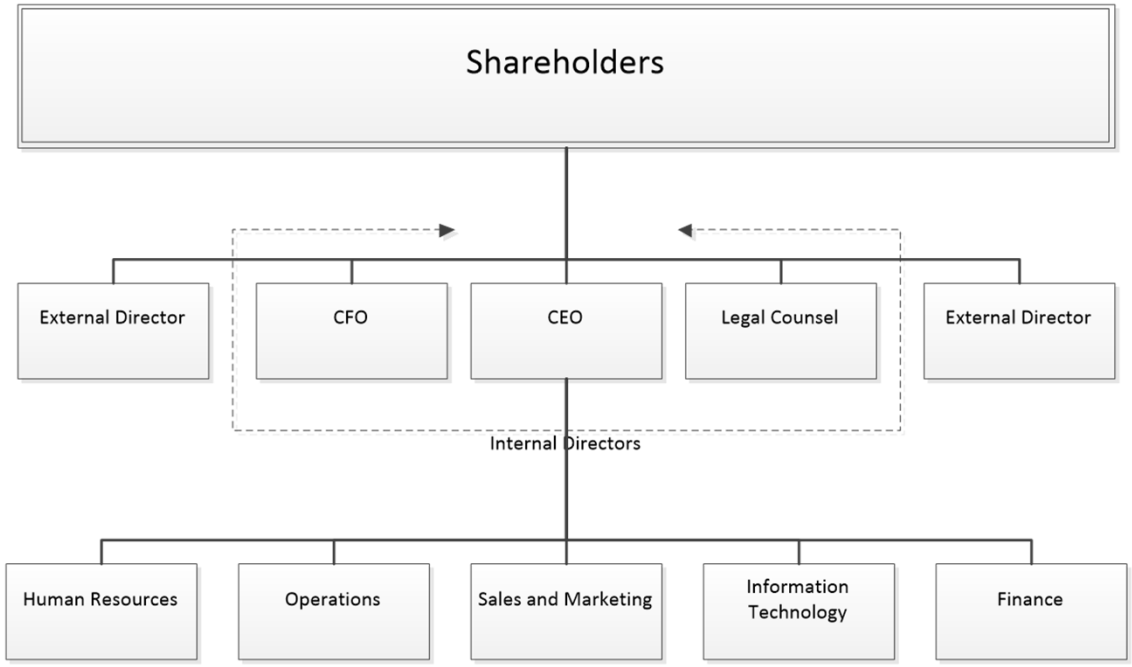
1. Three-party relationship, including:
   - The accountable party (auditee)
   - The user of the assurance report
   - Assurance professional

Accountable Party



Five Components of Audit and Assurance

1. Three-party relationship, including:
   - The accountable party (auditee)
   - The user of the assurance report
   - Assurance professional



Five Components of Audit and Assurance

1. **Three-party relationship**, including:
   - The accountable party (auditee)
   - The user of the assurance report
   - Assurance professional
2. **Subject matter** – the area within the audit universe that are under review in the assurance assignment.
3. **Suitable criteria** – reference against which the subject is evaluated
   - Usually established by Management
   - Design evaluated by assurance professional
4. **Assurance Process** –structured approach for execution of engagement
5. **Conclusions and recommendations**
   - Based on observations, facts and documentation
   - Identify control weaknesses and root causes
   - Substantiate the risks
   - Make recommendations

Fox School of Business
TEMPLE UNIVERSITY®

Information Technology Audit & Cyber Security

Introduction to Systems Development

Systems & Infrastructure
Lifecycle Management

LEARNING OBJECTIVES

- ✓Define information systems analysis and design.
- ✓Describe the information systems development life cycle (SDLC).
- ✓Explain computer-aided software engineering (CASE) tools.
- ✓Describe Agile Methodologies and eXtreme Programming.
- ✓Explain object-oriented analysis and design and the Rational Unified Process (RUP).

Introduction

IS Project

IS Development

Other Approaches

Agile Methods

OO / UML

This Course

INTRODUCTION

Information Systems Analysis and Design

- Complex organizational process
- Used to develop and maintain computer-based information systems
- Used by a team of business and systems professionals

Application Software

- Computer software designed to support organizational functions or processes

Systems Analyst

- Organizational role most responsible for analysis and design of information systems

INTRODUCTION (CONT.)

An organizational approach to systems analysis and design is driven by methodologies, techniques, and tools.

Sources: Mitarart/Fotolia, PaulPaladin/Fotolia

A MODERN APPROACH TO SYSTEMS ANALYSIS AND DESIGN

1950s: focus on efficient automation of existing processes

1960s: advent of procedural third generation languages (3GL) faster and more reliable computers

1970s: system development becomes more like an engineering discipline

1980s: major breakthrough with 4GL, CASE tools, object-oriented methods

1990s: focus on system integration, GUI applications, client/server platforms, Internet

The new century: Web application development, wireless PDAs and smart phones, component-based applications, per-use cloud-based application services.

Introduction

IS Project

IS Development

Other Approaches

Agile Methods

OO / UML

This Course

DEVELOPING INFORMATION SYSTEMS

System Development Methodology is a standard process followed in an organization to conduct all the steps necessary to analyze, design, implement, and maintain information systems.

SYSTEMS DEVELOPMENT LIFE CYCLE (SDLC)

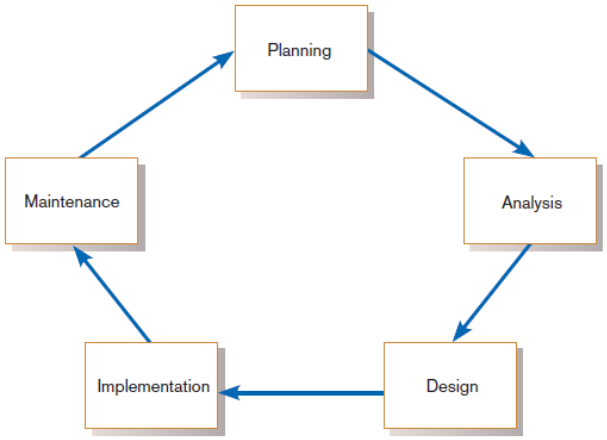
Traditional methodology used to develop, maintain, and replace information systems

Phases in SDLC:

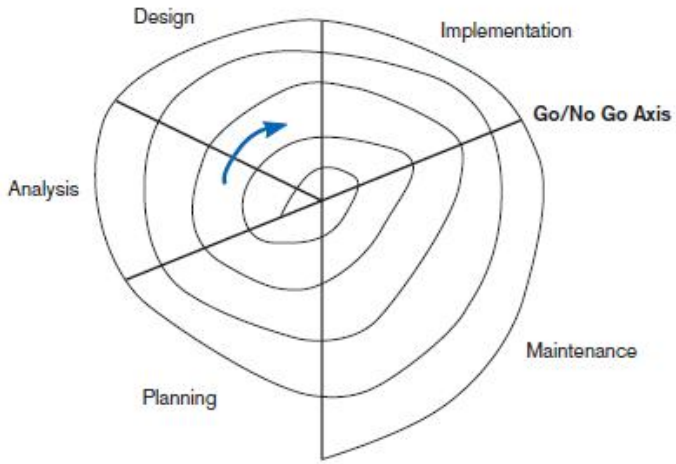
- Planning
- Analysis
- Design
- Implementation
- Maintenance

STANDARD AND EVOLUTIONARY VIEWS OF SDLC

Systems development life cycle



Evolutionary model



SYSTEMS DEVELOPMENT LIFE CYCLE (SDLC) (CONT.)

Planning – an organization's total information system needs are identified, analyzed, prioritized, and arranged

Analysis – system requirements are studied and structured

Design – a description of the recommended solution is converted into logical and then physical system specifications

Logical design – all functional features of the system chosen for development in analysis are described independently of any computer platform

Physical design – the logical specifications of the system from logical design are transformed into the technology-specific details from which all programming and system construction can be accomplished

Implementation – the information system is coded, tested, installed and supported in the organization

Maintenance – an information system is systematically repaired and improved

Phase	Products, Outputs, or Deliverables
Planning	Priorities for systems and projects; an architecture for data, networks, and selection hardware, and information systems management are the result of associated systems
	Detailed steps, or work plan, for project
	Specification of system scope and planning and high-level system requirements or features
	Assignment of team members and other resources
	System justification or business case
Analysis	Description of current system and where problems or opportunities exist, with a general recommendation on how to fix, enhance, or replace current system
	Explanation of alternative systems and justification for chosen alternative
Design	Functional, detailed specifications of all system elements (data, processes, inputs, and outputs)
	Technical, detailed specifications of all system elements (programs, files, network, system software, etc.)
	Acquisition plan for new technology
Implementation	Code, documentation, training procedures, and support capabilities
Maintenance	New versions or releases of software with associated updates to documentation, training, and support

A SPECIALIZED SYSTEMS DEVELOPMENT LIFE CYCLE

Microsoft's Security Development Lifecycle (SDL)

Training focuses on security.

These are like traditional SDLC's analysis, design, and implementation.

Verification focuses on product quality assurance.

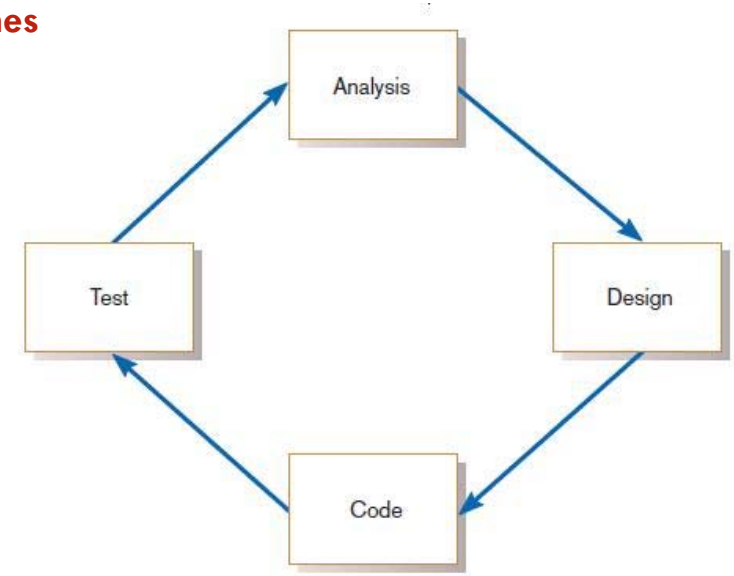
Release makes product available for general use.

Response deals with security problems that come up after product release.

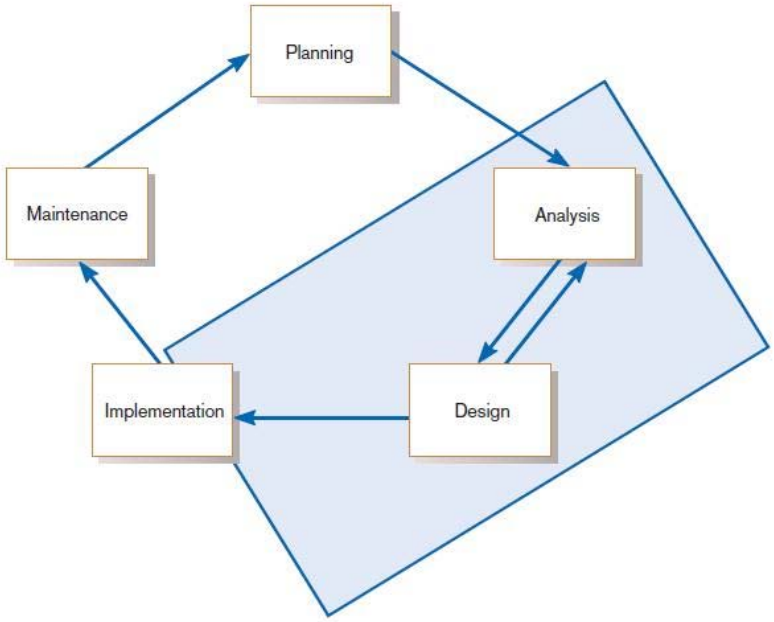
(*Source*: http://www.microsoft.com/security/sdl/default.aspx. Used by permission.)

THE HEART OF THE SYSTEMS DEVELOPMENT PROCESS

Analysis–design–code–test loop

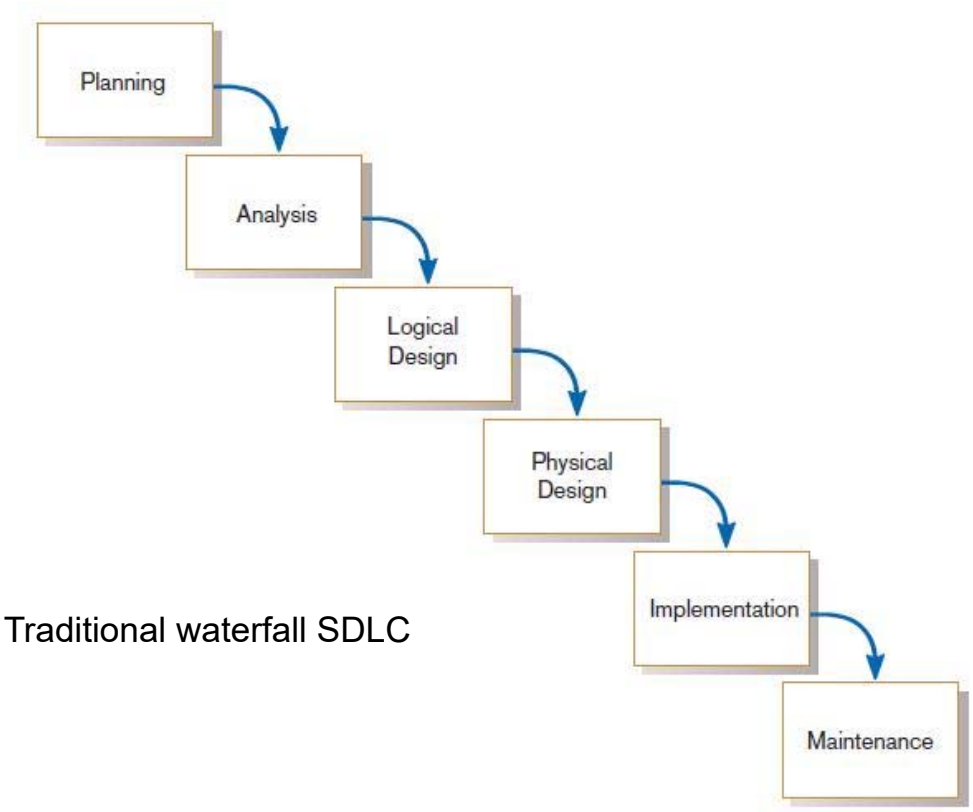


The heart of systems development



Current practice combines analysis, design, and implementation into a single iterative and parallel process of activities.

TRADITIONAL WATERFALL SDLC



Traditional waterfall SDLC

One phase begins when another completes, with little backtracking and looping.

Introduction

IS Project

IS Development

Other Approaches

Agile Methods

OO / UML

This Course

PROBLEMS WITH WATERFALL APPROACH

Feedback ignored, milestones lock in design specs even when conditions change

Limited user involvement (only in requirements phase)

Too much focus on milestone deadlines of SDLC phases to the detriment of sound development practices

Introduction

IS Project

IS Development

Other Approaches

Agile Methods

OO / UML

This Course

DIFFERENT APPROACHES TO IMPROVING DEVELOPMENT

CASE Tools

Agile Methodologies

eXtreme Programming

COMPUTER-AIDED SOFTWARE ENGINEERING (CASE) TOOLS

Diagramming tools enable graphical representation.

Computer displays and report generators help prototype how systems "look and feel".

Analysis tools automatically check for consistency in diagrams, forms, and reports.

A central repository provides integrated storage of diagrams, reports, and project management specifications.

Documentation generators standardize technical and user documentation.

Code generators enable automatic generation of programs and database code directly from design documents, diagrams, forms, and reports.

CASE TOOLS (CONT.)

SDLC Phase	Key Activities	CASE Tool Usage
Project identification and selection	Display and structure high-level organizational information	Diagramming and matrix tools to create and structure information
Project initiation and planning	Develop project scope and feasibility	Repository and documentation generators to develop project plans
Analysis	Determine and structure system requirements	Diagramming to create process, logic, and data models
Logical and physical design	Create new system designs	Form and report generators to prototype designs; analysis and documentation generators to define specifications
Implementation	Translate designs into an information system	Code generators and analysis, form and report generators to develop system; documentation generators to develop system and user documentation
Maintenance	Evolve information system	All tools are used (repeat life cycle)

Introduction

IS Project

IS Development

Other Approaches

Agile Methods

OO / UML

This Course

AGILE METHODOLOGIES

Motivated by recognition of software development as fluid, unpredictable, and dynamic

Three key principles

- Adaptive rather than predictive
- Emphasize people rather than roles
- Self-adaptive processes

Introduction

IS Project

IS Development

Other Approaches

Agile Methods

OO / UML

This Course

The Manifesto for Agile Software Development

Seventeen anarchists agree:

We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

- *Individuals and interactions* over processes and tools.
- *Working software* over comprehensive documentation.
- *Customer collaboration* over contract negotiation.
- *Responding to change* over following a plan.

That is, while we value the items on the right, we value the items on the left more. We follow the following principles:

- Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
- Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
- Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
- Businesspeople and developers work together daily throughout the project.
- Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
- The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
- Working software is the primary measure of progress.
- Continuous attention to technical excellence and good design enhances agility.
- Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
- Simplicity—the art of maximizing the amount of work not done—is essential.
- The best architectures, requirements, and designs emerge from self-organizing teams.
- At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

—Kent Beck, Mike Beedle, Arie van Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler, James Grenning, Jim Highsmith, Andrew Hunt, Ron Jeffries, Jon Kern, Brian Marick, Robert C. Martin, Steve Mellor, Ken Schwaber, Jeff Sutherland, Dave Thomas (*www.agileAlliance.org*)

The Agile Methodologies group argues that software development methodologies adapted from engineering generally do not fit with real-world software development.

Introduction

IS Project

IS Development

Other Approaches

Agile Methods

OO / UML

This Course

WHEN TO USE AGILE METHODOLOGIES

If your project involves:

- Unpredictable or dynamic requirements
- Responsible and motivated developers
- Customers who understand the process and will get involved

Introduction

IS Project

IS Development

Other Approaches

Agile Methods

OO / UML

This Course

Factor	Agile Methods	Traditional Methods
Size	Well matched to small products and teams. Reliance on tacit knowledge limits scalability.	Methods evolved to handle large products and teams. Hard to tailor down to small projects.
Criticality	Untested on safety-critical products. Potential difficulties with simple design and lack of documentation.	Methods evolved to handle highly critical products. Hard to tailor down to products that are not critical.
Dynamism	Simple design and continuous refactoring are excellent for highly dynamic environments but a source of potentially expensive rework for highly stable environments.	Detailed plans and Big Design Up Front, excellent for highly stable environment but a source of expensive rework for highly dynamic environments.
Personnel	Requires continuous presence of a critical mass of scarce experts. Risky to use non-agile people.	Needs a critical mass of scarce experts during project definition but can work with fewer later in the project, unless the environment is highly dynamic.
Culture	Thrives in a culture where people feel comfortable and empowered by having many degrees of freedom (thriving on chaos).	Thrives in a culture where people feel comfortable and empowered by having their roles defined by clear practices and procedures (thriving on order).

(*Source:* Boehm, Barry; Turner, Richard, *Balancing Agility and Discipline: A Guide for the Perplexed,* 1st Ed., © 2004. Reprinted and electronically reproduced by permission of Pearson Education, Inc.)

Introduction

IS Project

IS Development

Other Approaches

Agile Methods

OO / UML

This Course

EXTREME PROGRAMMING

Short, incremental development cycles

Automated tests

Two-person programming teams

Coding, testing, listening, designing

Coding and testing operate together

Advantages:

- Communication between developers
- High level of productivity
- High-quality code

Introduction

IS Project

IS Development

Other Approaches

Agile Methods

OO / UML

This Course

OBJECT-ORIENTED ANALYSIS AND DESIGN (OOAD)

Based on objects rather than data or processes

Object: a structure encapsulating attributes and behaviors of a real-world entity

OBJECT-ORIENTED ANALYSIS AND DESIGN (OOAD) (CONT.)

Object class: a logical grouping of objects sharing the same attributes and behaviors

Inheritance: hierarchical arrangement of classes enable subclasses to inherit properties of superclasses

RATIONAL UNIFIED PROCESS (RUP)

An object-oriented systems development methodology

Establishes four phase of development: inception, elaboration, construction, and transition

- Each phase is organized into a number of separate iterations.

Introduction

IS Project

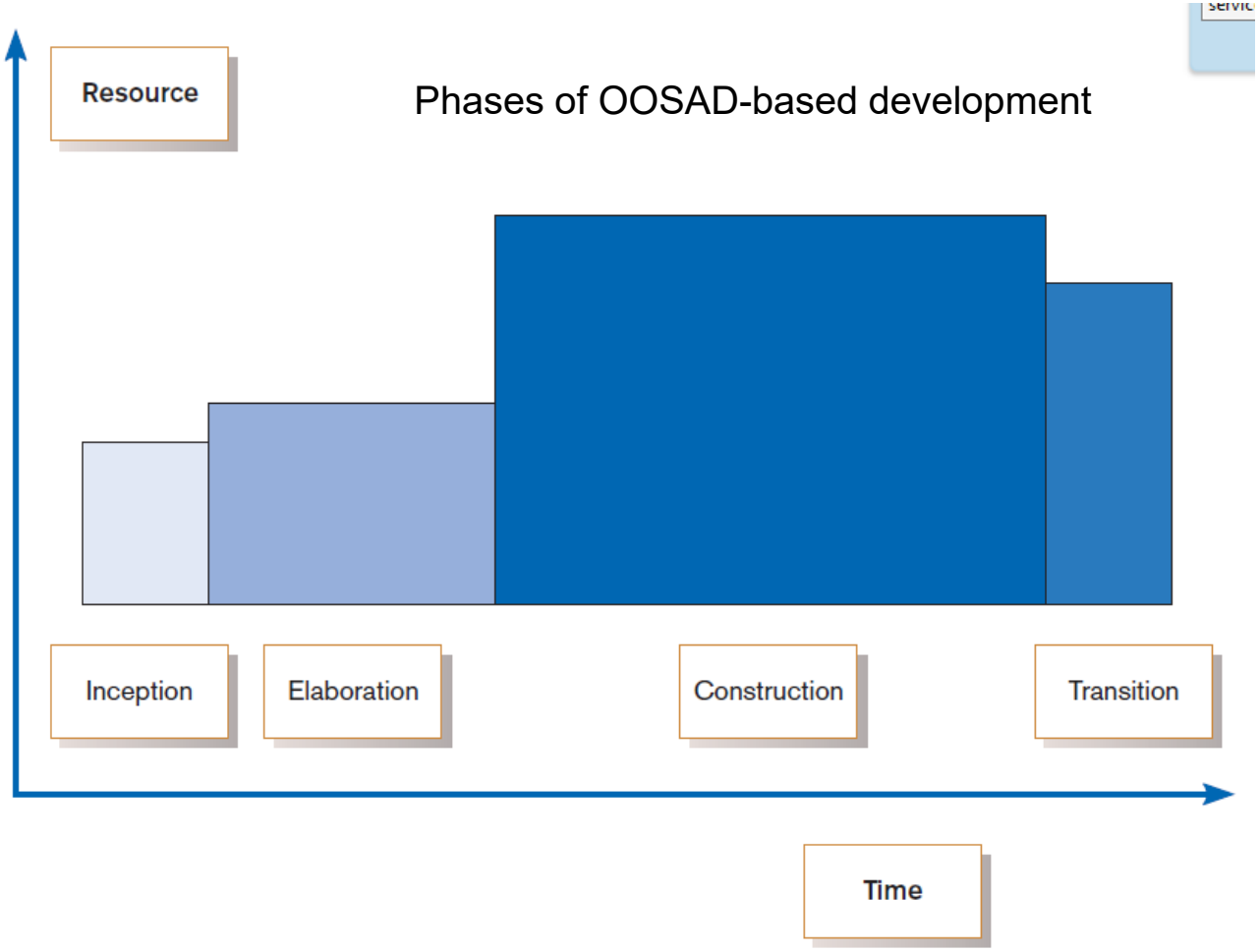
IS Development

Other Approaches

Agile Methods

OO / UML

This Course



Introduction

IS Project

IS Development

Other Approaches

Agile Methods

OO / UML

This Course

OUR APPROACH TO SYSTEMS DEVELOPMENT

Criticisms of SDLC

- Forcing timed phases on intangible processes (analysis and design) is doomed to fail
- Too much formal process and documentation slows things down
- Cycles are not necessarily waterfalls

And yet the concept of a cycle is in all methodologies. So, SDLC is a valuable model that has many variations.

SUMMARY

In this presentation you learned how to:

- ✓Define information systems analysis and design.
- ✓Describe the information systems development life cycle (SDLC).
- ✓Explain computer-aided software engineering (CASE) tools.
- ✓Describe Agile Methodologies and eXtreme Programming.
- ✓Explain object-oriented analysis and design and the Rational Unified Process (RUP).

Fox School of Business
TEMPLE UNIVERSITY®

Information Technology Audit & Cyber Security

The Origins of Software

Systems & Infrastructure Lifecycle Management

Introduction

Sources of Software

Outsourcing

Packaged Software

Vendors

Cloud

Open Source

In-House

Evaluation

Resuse

LEARNING OBJECTIVES

- ✓Explain outsourcing.
- ✓Describe six different sources of software.
- ✓Discuss how to evaluate off-the-shelf software.
- ✓Explain reuse and its role in software development.

Introduction

Sources of Software

Outsourcing

Packaged Software

Vendors

Cloud

Open Source

In-House

Evaluation

Resuse

INTRODUCTION

Historically, software development for a corporate information systems department was done primarily in-house.

Now it involves use of components from external sources.

Much in-house application coding involves making the components work together.

Six sources of software:

- Information technology service firms
- Packaged software providers
- Vendors of enterprise-wide solution software
- Cloud computing
- Open-source software
- In-house development

There are ways to evaluate software from sources

Introduction

Sources of Software

Outsourcing

Packaged Software

Vendors

Cloud

Open Source

In-House

Evaluation

Resuse

SOURCES OF SOFTWARE

Information technology services firm

Packaged software producers

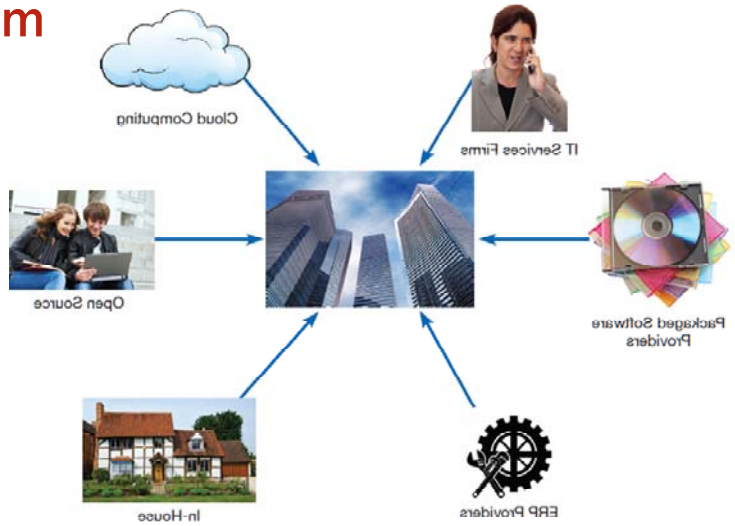
Enterprise solutions software

- Enterprise Resource Planning (ERP)

Cloud computing

Open source software

In-house developers



Sources of Application Software

Introduction

Sources of Software

Outsourcing

Packaged Software

Vendors

Cloud

Open Source

In-House

Evaluation

Resuse

SYSTEMS ACQUISITION: OUTSOURCING

Outsourcing: The practice of turning over responsibility of some or all of an organization's information systems applications and operations to an outside firm

Introduction
Sources of Software
Outsourcing
Packaged Software
Vendors
Cloud
Open Source
In-House
Evaluation
Resuse

SYSTEMS ACQUISITION: OUTSOURCING (CONT.)

Outsourcing Example

- Shell Oil outsource spending: $3.2 billion (2008)
- Shell's outsourcing vendors (2008-2011): EDS, T-Systems, AT&T, IBM, Logica, Wipro, Accenture

Introduction

Sources of Software

Outsourcing

Packaged Software

Vendors

Cloud

Open Source

In-House

Evaluation

Resuse

OUTSOURCING (CONT.)

Reasons to outsource

- Cost-effectiveness
- Take advantage of economies of scale
- Make up for lack of in-house knowledge
- Free up internal resources
- Reduce time to market
- Increase process efficiencies
- System development is a non-core activity for the organization
- Political reasons (e.g. labor disputes)

Introduction

Sources of Software

Outsourcing

Packaged Software

Vendors

Cloud

Open Source

In-House

Evaluation

Resuse

GLOBAL OUTSOURCING

Top outsourcing countries: India, China, Malaysia (A.T. Kearny report 2014)

Top 10 are in Asia, Latin America, Europe, and Africa

Some U.S. firms are switching to nearshoring (same time zone, low labor costs)

Introduction

Sources of Software

Outsourcing

Packaged Software Vendors

Cloud

Open Source

In-House

Evaluation

Resuse

SOURCES OF SOFTWARE (CONT.)

Specialization	Example Firms or Websites
IT Services	Accenture Computer Sciences Corporation (CSC) IBM HP
Packaged Software Providers	Intuit Microsoft Oracle SAP AG Symantec
Enterprise Software Solutions	Oracle SAP AG
Cloud Computing	Amazon.com Google IBM Microsoft Salesforce.com
Open Source	SourceForge.net

Introduction

Sources of Software

Outsourcing

Packaged Software

Vendors

Cloud

Open Source

In-House

Evaluation

Resuse

INFORMATION TECHNOLOGY (IT) SERVICES FIRMS

Help companies develop custom information systems for internal use

Develop, host, and run applications for customers

Provide other services (management, accounting, auditing, financial)

Introduction

Sources of Software

Outsourcing

Packaged Software

Vendors

Cloud

Open Source

In-House

Evaluation

Resuse

PACKAGED SOFTWARE PRODUCERS

Serve many market segments

Provide software ranging from broad-based packages (i.e. general ledger) to niche packages (i.e. day care management)

Pre-packaged, off-the-shelf software

Introduction

Sources of Software

Outsourcing

Packaged Software

Vendors

Cloud

Open Source

In-House

Evaluation

Resuse

PACKAGED SOFTWARE PRODUCERS (CONT.)

Software runs on all size computers, from microcomputers to large mainframes.

Prepackaged software is off-the-shelf, turnkey software (i.e. not customizable).

Off-the-shelf software, at best, meets 70% of organizations' needs.

Introduction

Sources of Software

Outsourcing

Packaged Software

Vendors

Cloud

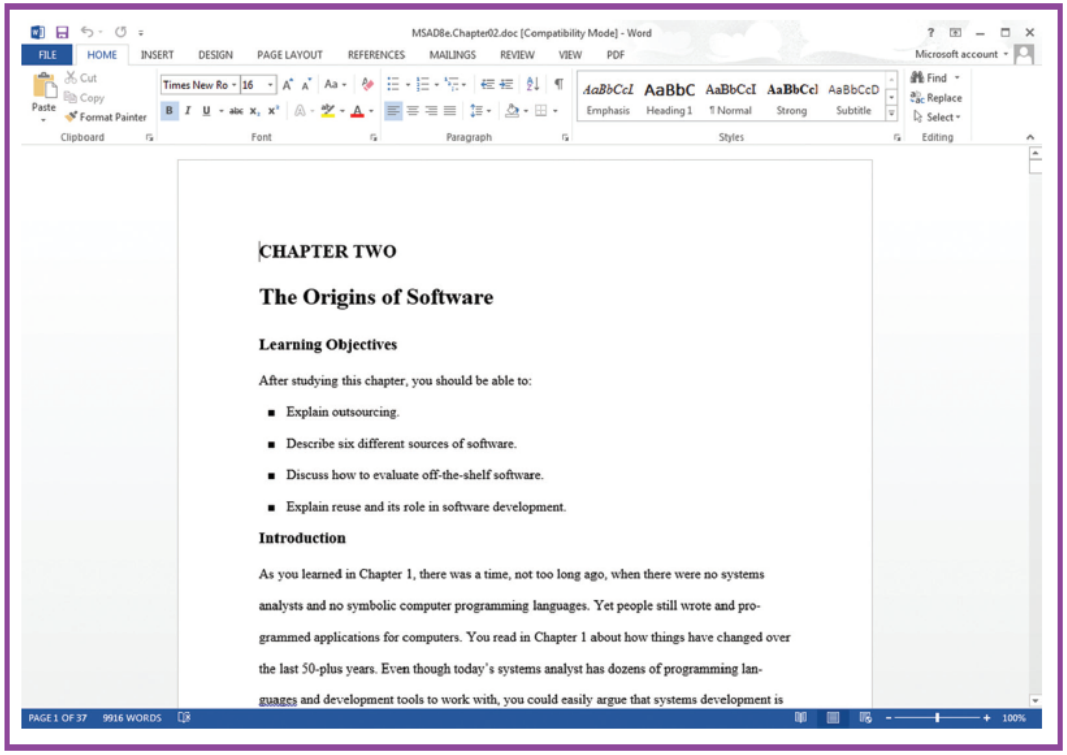
Open Source

In-House

Evaluation

Resuse

PREPACKAGED SOFTWARE



A document created in Microsoft's Word
(*Source:* Microsoft Corporation.)

Introduction

Sources of Software

Outsourcing

Packaged Software

Vendors

Cloud

Open Source

In-House

Evaluation

Resuse

ENTERPRISE SOLUTIONS SOFTWARE

Enterprise Resource Planning (ERP) systems integrate individual traditional business functions into modules enabling a single seamless transaction to cut across functional boundaries.

SAP AG is the leading vendor of ERP systems.

Introduction

Sources of Software

Outsourcing

Packaged Software

Vendors

Cloud

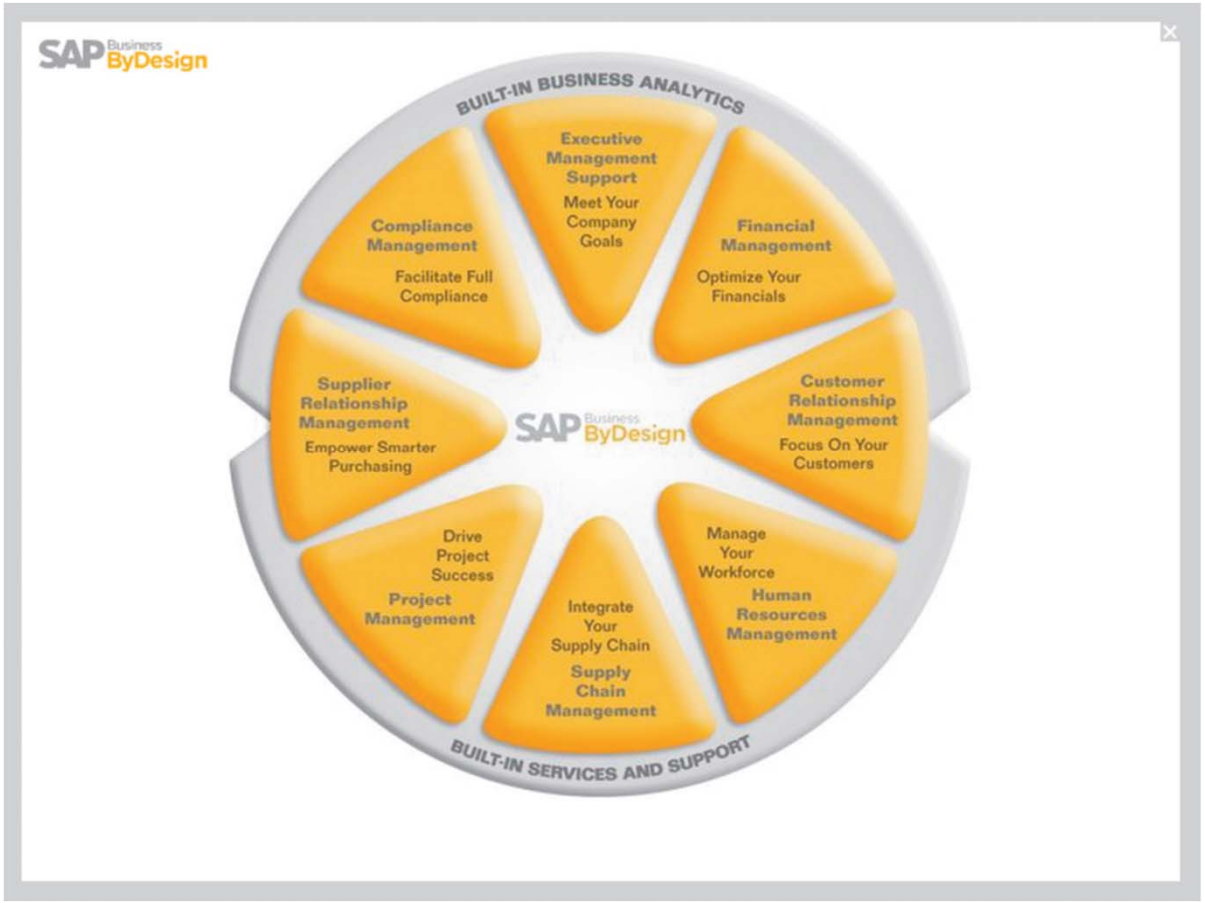
Open Source

In-House

Evaluation

Resuse

ENTERPRISE SOLUTIONS SOFTWARE (CONT.)



SAP's Business ByDesign, a product designed for medium sized companies.

Introduction

Sources of Software

Outsourcing

Packaged Software

Vendors

Cloud

Open Source

In-House

Evaluation

Resuse

CLOUD COMPUTING

The provision of computing resources, including applications, over the Internet, so customers do not have to invest in the computing infrastructure needed to run and maintain the resources

Pay-per-use or monthly/yearly licenses

Introduction

Sources of Software

Outsourcing

Packaged Software

Vendors

Cloud

Open Source

In-House

Evaluation

Resuse

CLOUD COMPUTING (CONT.)

Examples:

- Google Apps– for sharing documents, spreadsheets, and presentations
- Salesforce.com – online customer relationship management (CRM) software
  - An example of software as a service (SaaS)
- Microsoft Azure platform
- Amazon.com cloud infrastructure and services
  - An example of hardware as a service (HaaS)

Introduction

Sources of Software

Outsourcing

Packaged Software

Vendors

Cloud

Open Source

In-House

Evaluation

Resuse

CLOUD COMPUTING (CONT.)

Heavy growth predicted

Benefits:

- Frees company of internal IT staff requirements
- Faster access to application than via internal development
- Lower cost than internal development

Concerns

- Security
- Reliability
- Regulation compliance

Introduction

Sources of Software

Outsourcing

Packaged Software

Vendors

Cloud

Open Source

In-House

Evaluation

Resuse

OPEN SOURCE SOFTWARE

Freely available including source code

Developed by a community of interested people

Performs the same functions as commercial software

Examples: Linux, mySQL, Firefox

How to make money?

- Provide maintenance/services
- Sell a more featured version of the free software

Introduction

Sources of Software

Outsourcing

Packaged Software

Vendors

Cloud

Open Source

In-House

Evaluation

Resuse

IN-HOUSE DEVELOPMENT

If sufficient system development expertise with the chosen platform exists in-house, then some or all of the system can be developed by the organization's own staff.

- In-house development usually leads to more maintenance burden than other approaches

Hybrid solutions involving some purchased and some in-house components are common.

Introduction

Sources of Software

Outsourcing

Packaged Software

Vendors

Cloud

Open Source

In-House

Evaluation

Resuse

SOURCES OF SOFTWARE COMPONENTS

Producers	When to Go to This Type of Organization for Software	Internal Staffing Requirements
IT services firms	When task requires custom support and system can't be built internally or system needs to be sourced	Internal staff may be needed, depending on application
Packaged software producers	When supported task is generic	Some IS and user staff to define requirements and evaluate packages
Enterprise-wide solutions vendors	For complete systems that cross functional boundaries	Some internal staff necessary but mostly need consultants
Cloud computing	For instant access to an application; when supported task is generic	Few; frees up staff for other IT work
Open-source software	When supported task is generic but cost is an issue	Some IS and user staff to define requirements and evaluate packages
In-house developers	When resources and staff are available and system must be built from scratch	Internal staff necessary though staff size may vary

Introduction

Sources of Software

Outsourcing

Packaged Software

Vendors

Cloud

Open Source

In-House

Evaluation

Resuse

SELECTING OFF-THE-SHELF SOFTWARE

Cost: comparing the cost of developing the same system in-house with the cost of purchasing or licensing the software package

Functionality: the tasks that the software can perform and the mandatory, essential, and desired system features

Vendor support: whether and how much support the vendor can provide and at what cost

Viability of vendor: can vendor continue to adapt/update software to changes in systems software and hardware

Flexibility: the ease with which software is customized

Documentation: understandable and up-to-date user's manual and technical documentation

Response time: how long it takes the software package to respond to the user's requests in an interactive session

Ease of installation: a measure of the difficulty of loading the software and making it operational

Introduction

Sources of Software

Outsourcing

Packaged Software

Vendors

Cloud

Open Source

In-House

Evaluation

Resuse

VALIDATING PURCHASED SOFTWARE INFORMATION

Send a **request for proposal (RFP)** to vendors.

- RFP – a document provided to vendors to ask them to propose hardware and system software that will meet the requirements of a new system

Use a variety of information sources:

- Collect information from vendor
- Software documentation
- Technical marketing literature

Introduction

Sources of Software

Outsourcing

Packaged Software

Vendors

Cloud

Open Source

In-House

Evaluation

Resuse

REQUEST FOR PROPOSAL (RFP)

Sometimes called a **Request For Quote (RFQ)**

Analyst selects best candidates based on:

- vendor bids
- a variety of information sources

Introduction

Sources of Software

Outsourcing

Packaged Software

Vendors

Cloud

Open Source

In-House

Evaluation

Resuse

INFORMATION SOURCES FOR RFP

Vendor's proposal

Running software through a series of tests

Feedback from other users of the vendor's product

Independent software testing services

Customer surveys

Articles in trade publications are sometimes biased (seeded by manufacturer)

Introduction

Sources of Software

Outsourcing

Packaged Software

Vendors

Cloud

Open Source

In-House

Evaluation

Resuse

REUSE

The use of previously written software resources, especially objects and components, in new applications

Commonly applied to two different development technologies:

- Object-oriented development
- Component-based development

Introduction

Sources of Software

Outsourcing

Packaged Software

Vendors

Cloud

Open Source

In-House

Evaluation

Resuse

REUSE (CONT.)

Object-oriented development

- Object class encapsulates data and behavior of common organizational entities (e.g. employees)

Component-based development

- Components can be as small as objects or as large as pieces of software that handle single business functions

Introduction

Sources of Software

Outsourcing

Packaged Software

Vendors

Cloud

Open Source

In-House

Evaluation

Resuse

REUSE (CONT.)

Can be effective (increased productivity, less defects, reduced rework)

Technical issues – lack of methodology for component library (creating and labeling reusable components)

Organizational issues – lack of commitment, training, and organizational support; hard to measure economic benefits; legal and contractual issues

Introduction

Sources of Software

Outsourcing

Packaged Software

Vendors

Cloud

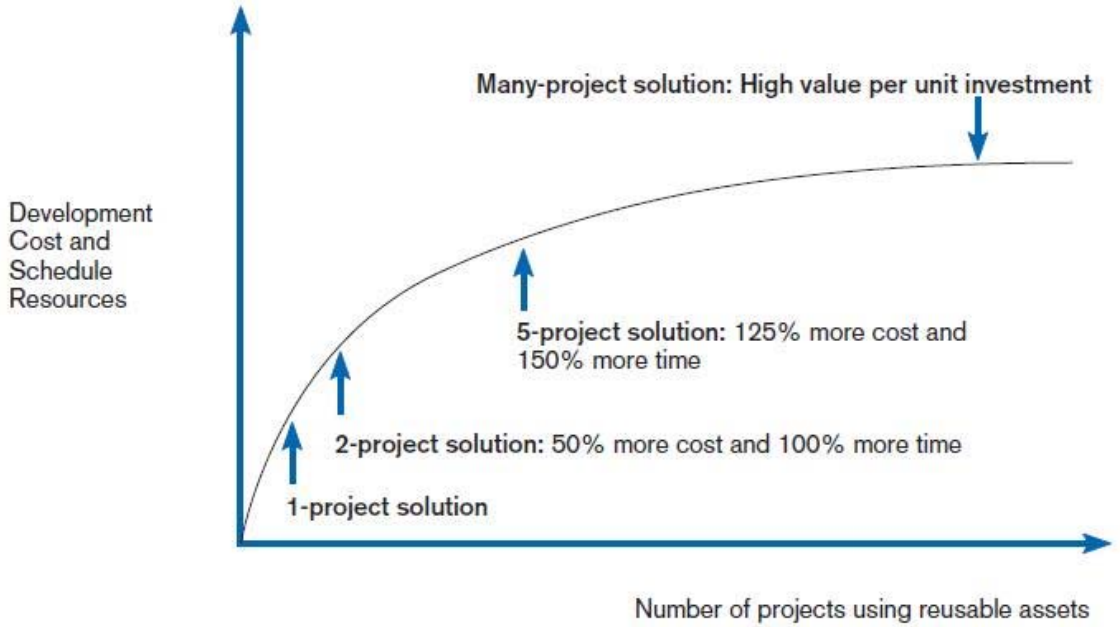
Open Source

In-House

Evaluation

Resuse

COSTS AND BENEFITS OF REUSE



Investments necessary to achieve reusable components
(*Source:* Royce, Walker, *Software Project Management: A Unified Framework,* 1st ed.,©1998.

Introduction

Sources of Software

Outsourcing

Packaged Software

Vendors

Cloud

Open Source

In-House

Evaluation

Resuse

3 STEPS OF SOFTWARE REUSE

Abstraction – design of reusable piece of software

Storage – making software assets available for others

Recontextualization – making the software understandable to developers

(Grinter, 2001)

Introduction

Sources of Software

Outsourcing

Packaged Software

Vendors

Cloud

Open Source

In-House

Evaluation

Resuse

APPROACHES TO REUSE

Ad-hoc: individuals are free to find or develop reusable assets on their own

Facilitated: developers are encouraged to practice reuse

Managed: the development, sharing, and adoption of reusable assets is mandated

Designed: assets mandated for reuse as they are being designed for specific applications

(Griss 2003)

Introduction

Sources of Software

Outsourcing

Packaged Software

Vendors

Cloud

Open Source

In-House

Evaluation

Resuse

APPROACHES TO REUSE (CONT.)

Approach	Reuse Level	Cost	Policies and Procedures
Ad hoc	None to low	Low	None.
Facilitated	Low	Low	Developers are encouraged to reuse but are not required to do so.
Managed	Moderate	Moderate	Development, sharing, and adoption of reusable assets are mandated; organizational policies are established for documentation, packaging, and certification.
Designed	High	High	Reuse is mandated; policies are put in place so that reuse effectiveness can be measured; code must be designed for reuse during initial development, regardless of the application it is originally designed for; there may be a corporate office for reuse.

(*Source:* Based on Flashline, Inc. and Griss, 2003.)

SUMMARY

In this slide deck we discussed:

- ✓Explain outsourcing.
- ✓Describe six different sources of software.
- ✓Discuss how to evaluate off-the-shelf software.
- ✓Explain reuse and its role in software development.