Systems and Infrastructure Lifecycle Management

Introduction

- Unit 1 -

1

Agenda

- Introduction
- IT Auditor's role in System and Infrastructure lifecycle
- Project and Visio
- IT Auditor's responsibilities and roles
- Information system development – a brief history
- Type of business information systems
- Quiz

2

Introduction

Introduce yourself briefly to your instructor and the class

- Name You Wish To Be Called
- Education
- Work history
- Experience with Information Systems Development
- Fun Fact

3

Introduction

This course will introduce you to the methods used as organizations builds an enterprise information system architecture within an environment of internal control

Topics include:

- Information system planning, development, acquisition and maintenance of these technologies and their impact on the organization's business processes

4

Introduction

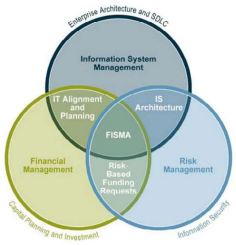
Course objective - help you understand what it means to:

- Evaluate the business case for the proposed investments in information systems
- Evaluate the project management framework and controls to determine whether business requirements are achieved in a cost-effective manner while managing risks to the organization
- Conduct reviews to determine whether a project is progressing in accordance with project plans, is adequately supported by documentation, and has timely and accurate status reporting
- Evaluate controls for information systems during the requirements, acquisition, development and testing standards, procedures and applicable external requirements
- Evaluate the readiness of information systems for implementation and migration into production to determine whether project deliverables, controls and the requirements are met
- Conduct post-implementation reviews of systems to determine whether project deliverables, controls and the requirements are met

5

IT Auditors' responsibilities

- Providing assurance that enterprise objectives are being met by information systems and infrastructure management practices
- Identifying which elements may represent greatest risk, and which controls are most effective at mitigating this risk



NIST SP 800-64r2 Security Considerations in the System Development Life Cycle

6

IT Auditors are responsible for

- Understanding which methodologies are in use for:
  - Systems development, acquisition and maintenance
  - Identifying potential vulnerabilities and points requiring control
- Advising project team and senior management of deficiencies and best practices within each of these processes

NIST SP 800-64r2 Security Considerations in the System Development Life Cycle

7

IT Auditor's Role in Information System Development

Two alternative approaches

1. Review end-stage deliverables throughout the development process, without becoming part of the process
   - Auditor reviews each stage's deliverables to ensure:
     i. What was planned from the previous stage has been accomplished and the planning of the next stage has been refined appropriately
     ii. Planning of the next stage has been refined appropriately
2. Internal control consultant, becoming part of the systems development process
   - Auditor provides ongoing proactive recommendations by participating in selected project-management meetings including: risk-assessment, systems-design, development, and systems delivery meetings
   - Auditor's independence may be compromised, but this is mitigated by another auditor who should find a system with well-designed controls incorporated

8

IT Auditor's Role in Information System Development

Produce and provide formal audit reports to the appropriate business managers including:

1. Overall assessment of the controlled progress of the project
2. Areas requiring improvement to complete the project, as specified, within budget and at an appropriate level of quality

Requires an in-depth understanding of both:

1. The overall information systems development processes adopted
2. The business processes being computerized

9

Information Systems Development – a brief history

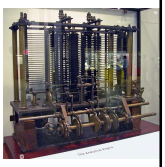
- **Prior 1946** - Before "stored-program" digital computers
  - Devices were pure hardware and had no software - their computing powers were directly tied to their specific form and engineering
  - Computing as a concept goes back to ancient times
    - Beginning with devices such as the **abacus**
      - Calculating tool used in China, Europe, and Russia centuries before adoption of written Hindu-Arabic numeral system we use today
    - Continuing on through early examples of computing such as the **Antikythera** mechanism
      - Ancient Greek analog computer used as a calendar to predict eclipses and astronomical positions decades in advance

Wikipedia – History of Software

10

Information Systems Development – a brief history

- **Prior 1946** - Before "stored-program" digital computers
  - 1837 – **The Analytical Engine**
    - First design for a general-purpose computer
    - Designed by English mathematician Charles Babbage
    - Incorporated:
      - Integrated memory
      - Arithmetic logic unit
      - Control flow in the form of conditional branching and loops
    - Logical structure essentially the same as the computer design that dominates in today's electronic era
    - First known computer program was written by Ada Lovelace to implement Luigi Menabrea's equations for generating a Bernoulli number sequence of rational numbers
      - The Analytical Engine predated the techniques of electrical engineering needed to run it



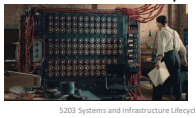
Wikipedia – History of Software

11

Information Systems Development – a brief history

Prior 1946 - Before "stored-program" digital computers

- **1935** – Alan Turing proposed the first modern theory of **software**
  - Software requires
    - A **general-purpose processor** - described as a Turing machine
    - **Computer memory**
      - In which reusable sets of routines and mathematical functions comprising programs can be stored, started, and stopped individually
  - This concept is recent in human history, led to the creation of the twin academic fields of **computer science** and **software engineering**



Wikipedia – History of Software

12

Information Systems Development – a brief history

- **1948 – 1979** Early days of computer software
  - 1948 - Claud Shannon "Father of Information Theory" wrote *A Mathematical theory of Communication* and provided an outline for how **binary logic** could be **implemented to program a computer**
    - Subsequently, the first computer programmers used binary code to instruct computers to perform various tasks
  - 1948 – ***Birth of Software*** Tim Kilburn at the University of Manchester UK wrote the first program code stored in an electronic memory to calculate the highest factor of an integer
  - 1950's – 1960's ***Development of high-level computer languages*** Fortran, LISP, COBOL and BASIC allowed programs to be specified in an abstract way, independent of the precise details of the hardware architecture of the computer

Grace Hopper developed the "self-documenting" COBOL (COmmon Business Oriented Language)

Margaret Hamilton led development of the onboard flight software for NASA's Apollo spacecraft coined the term "software engineering"

Wikipedia – History of Software

13

Information Systems Development – a brief history

- **1948 – 1979** Early days of computer software
  - 1960's – Massachusetts Institute of Technology, AT&T Bell Labs, and General Electric jointly developed an experimental **time sharing operating system** called Multics
    - Allowing multiple users to access a ***mainframe computer*** simultaneously
  - 1970's – Bell Lab's researchers left the team and implemented a **self-hosting operating system that became UNIX** on a ***minicomputer***
    - Included concepts of computer processes, device files, hierarchical file system, command-line interpreter, editor, programing shell, and assembler
    - Text editor and first text formatting and publishing program written in assembly language
    - 1971 – *UNIX Programmer's Manual written*
    - 1973 – Unix Version 3 rewritten in higher-level C language
    - Most popular variant of Unix today is macOS Mac OS X
    - Linux is closely related

Wikipedia – History of Software

14

Information Systems Development – a brief history

- **1975 -** Early days of computer software
  - 1975 – Micro Instrumentation and Telemetry Systems begins selling Altair 8800 microcomputer kit by mail order
    - Microsoft released its first product Altair BASIC Operating system
  - Before microcomputers – a successful software program sold for $50,000 - $60,000 each in units of 1,000 units; PC software sold thousands of copies for $50 - $700 each

Wikipedia – History of Software

15

Information systems Development – a brief history

Era	Hardware	Operating System	Applications
Mainframe (1970s)	Terminals connected to mainframe computer.	Time-sharing (TSO) on MVS	Custom-written MRP software
PC (mid-1980s)	IBM PC or compatible. Sometimes connected to mainframe computer via expansion card.	MS-DOS	WordPerfect, Lotus 1-2-3
Client-Server (late 80s to early 90s)	IBM PC "clone" on a Novell Network.	Windows for Workgroups	Microsoft Word, Microsoft Excel
World Wide Web (mid-90s to early 2000s)	IBM PC "clone" connected to company intranet.	Windows XP	Microsoft Office, Internet Explorer
Web 2.0 (mid-2000s to present)	Laptop connected to company Wi-Fi.	Windows 7	Microsoft Office, Firefox
Post-PC (today and beyond)	Apple iPad	iOS	Mobile-friendly websites, mobile apps

Manufacturing Resource Planning (ERP)

Enterprise Resource Planning (ERP)

https://bus206.pressbooks.com/chapter/chapter-1/

16

Information Systems Development – a brief history

- **First-generation computer programming Languages "1GLs" (1948 – 1950's)**
- **Second-generation "2GLs" (1950's - today)**
- **Third-generation "3GLs" (1950's – today)**
- **Fourth-generation "4GLs" (1970's – today)**
- **Fifth-generation "5GLs" (1980's – today)**

Citation: https://www.revolvy.com

17

Information Systems Development – a brief history

- **First-generation "1GLs" (1948 – 1950's)**
  - Binary machine languages entered through front panel switches
- **Second-generation "2GLs" (1950's - today) -** Machine-dependent assembly languages
  - Code read and written by a programmer, to run on a computer it must be converted into a machine readable form, a process called assembly
  - Assembly language is specific to a particular processor family and environment (today usually found in device drivers)
  - Considered "**low-level**" because they are designed for and executed by physical hardware without further translation required.
- **Third-generation "3GLs" (1950's – today) -** High-level programming languages, such as FORTRAN, LISP, COBOL, BASIC, Pascal, C, C++, C#, Java, and Javascript
  - Machine independent, more programmer-friendly, considered "**high-level**" because they are closer to human languages and further from machine languages, and hence require compilation or interpretation
  - Must be translated into machine language by a compiler or directly into behavior by an interpreter or compiler
  - Support structured programming, some support object-oriented programming

Citation: https://www.revolvy.com

18

Information Systems Development – a brief history

- **Fourth-generation "4GLs" (1970's – today)** - Add additional features within general purpose 3GL environments, attempt to get closer to human language and require less coding than lower-level languages, such as Python, Ruby and Perl
  - Command-line languages that may includes support for general processing (e.g. UNIX Shell, Visual DataFlex, PowerBuilder, Cognos PowerHouse 4GL, DataFlex, IBM Rational EGL, Oracle Application, Development Framework,…)
  - Database management (e.g. SQL, FOCUS, OPenROAD, NATURAL, Informix-4GL,..)
  - Report generation (e.g. RPG-II, Oracle Reports, Progress 4GL)
  - Data manipulation and mathematical analysis (e.g. SAS, SPSS, R, S, PL/SQL, Stata, MATLAZB, MathProg,…)
  - Graphical User Interface (GUI) development (XUL, Visual DataFlex, Progress 4GL, OPenRoad, 4th Dimension,…) ,
  - Web development (e.g. ActiveVFP, CFML, Wavemaker, OutSystems…)

- **Fifth-generation "5GLs" (1980's – today) -** Solve problems by providing constraints to a general inferencing program without a programmer writing additional code or developing additional algorithms
  - Mainly used in artificial intelligence languages built on LISP (e.g. OPS5, SNEPS, Mercury…)
  - In 1980s, fifth-generation languages considered "way of the future". Some predicted they would replace all other languages for system development, with exception of low-level languages
  - 1982 to 1993, Japan invested heavily in 5th generation computer systems project, hoping to design a massive computer network of general inferencing machines
  - Development of larger programs revealed flaws of the approach
    - Given a set of constraints defining a particular problem, deriving an efficient algorithm to solve it is a very difficult problem
    - This crucial step cannot yet be automated and still requires the insight of a human developer

Citation: https://www.revolvy.com

19

Information Systems Development – a brief history

Most information systems today are developed with a combination of 3GL and 4GL capabilities

Third-generation "3GLs" (1950's – today)

- High-level programming languages, such as FORTRAN, LISP, COBOL, BASIC, Pascal, C, C++, C#, Java, and Javascript

Fourth-generation "4GLs" (1970's – today)

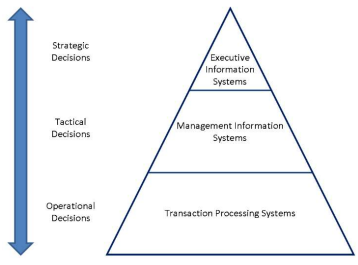
- Add additional features within general purpose 3GL environments, attempt to get closer to human language and require less coding than lower-level languages, such as Python, Ruby and Perl

Citation: https://www.revolvy.com

20

Types of Business Information Systems

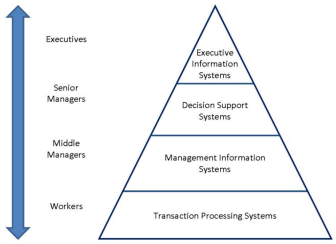
What types of information systems are used in businesses today?



21

Types of Business Information Systems

What types of information systems are used in businesses today?



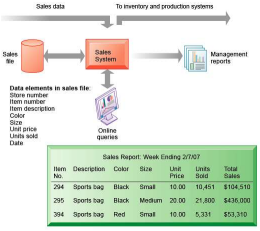
22

Some categories of Business Information Systems

Sales and marketing information systems

Help the firm with marketing business processes
- identifying customers for the firm's products or services
- developing products and services to meet their needs
- promoting products and services

and sales processes
- selling the products and services
- taking orders contacting customers
- providing customer support



Sales Report: Week Ending 2/7/07

Item No.	Description	Color	Size	Unit Price	Units Sold	Total Sales
294	Sports bag	Black	Small	10.00	10,451	$104,510
295	Sports bag	Black	Medium	20.00	21,800	$436,000
394	Sports bag	Red	Small	10.00	5,331	$53,310

23

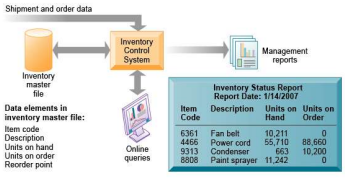
Some categories of Business Information Systems

Manufacturing and production information systems

Deal with
- planning
- development
- production

of products and services

and controlling the flow of production



Inventory Status Report
Report Date: 1/14/2007

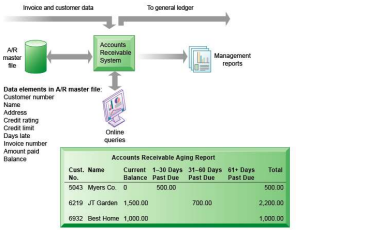
Item Code	Description	Units on Hand	Units on Order
6361	Fan belt	10,211	0
4466	Power cord	55,710	88,660
9313	Condenser	663	10,200
8808	Paint sprayer	11,242	0

24

Some categories of Business Information Systems

Finance and accounting information systems

Keep track of the firm's financial assets and fund flows



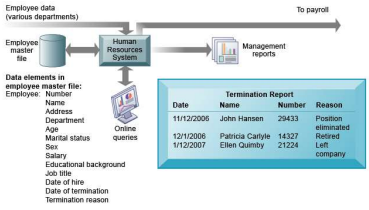
25

Some categories of Business Information Systems

Human resources information systems

Maintain employee records, tracking
- employee skills
- job performance
- Training

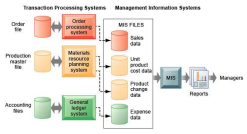
Support planning for
- employee compensation
- career development



26

Some categories of Business Information Systems

1. **Transaction processing systems (TPS)**
   Basic business systems serving operations by recording daily routine transactions for conducting business, such as payroll and sales receipts
2. **Management information systems (MIS)**
   Provide current and historical performance information aiding planning, controlling, and decision making.

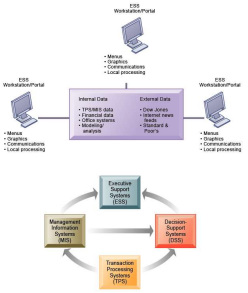


Laudon, K.C. and Laudon, J.P. Management Information Systems 10th Edition

27

Some categories of Business Information Systems

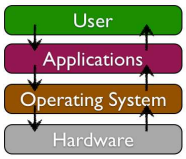
3. **Decision support systems (DSS)** a.k.a. business intelligence systems
   - Support non-routine decisions not easily specified in advance
   - Use a variety of analytic models applied to large amounts of internal and external data
4. **Executive support systems (ESS)**
   - For non-routine strategic decisions requiring judgment, evaluation, and insight
   - Visual presentations of data within an interactive interface easy for senior managers to use
   - Information often delivered via a web portal as integrated information



28

Components of Information Systems

- **Hardware**- these are the devices like the monitor, processor, printer and keyboard, all of which work together to accept, process, show data and information
- **Networks**- are a connecting system that allows diverse computers to distribute resources.
- **Software**- are the programs that allow the hardware to process the data.
  - **Databases**- are the gathering of associated files or tables containing related data.
  - **Procedures**- are the commands to the software and databases (for combining the components above to process information and produce the preferred output)



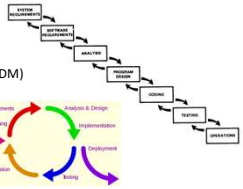
29

Information System Development Methods – Next week Unit #2

Software development lifecycle (SDLC) models are formal management processes for guiding the development of information systems

These include:

- Waterfall Models
  - Waterfall
  - Structured Systems Analysis and Design Method (SSADM)
- Spiral and Iterative Models
  - Structured Rapid Prototyping
  - Rapid Application development (RAD)
  - Agile Models
  - Rational Unified Model



If execution of the SDLC methodology is inadequate, however, the project may fail to meet business and user needs.

- ***IS Auditor is responsible for verifying that the SDLC model is appropriate for the project's goals and is properly implemented***

30

Quiz

Normally, it would be essential to involve which of the following stakeholders in the initiation stage of a project?

a) System owners
b) System users
c) System designers
d) System builders

An IS auditor has been asked to participate in project initiation meetings for a critical project. The IS auditor's MAIN concern should be that the:

a) complexity and risk associated with the project have been analyzed.
b) resources needed throughout the project have been determined.
c) technical deliverables have been identified.
d) a contract for external parties involved in the project has been completed.

31

Agenda

- Introduction
- IT Auditor's role in System and Infrastructure lifecycle
- Project and Visio
- IT Auditor's responsibilities and roles
- Information system development – a brief history
- Type of business information systems
- Quiz

32