Lecture01: Introduction

EGCI341: SOFTWARE ENGINEERING (WEEK 01)

Overview

- US News: Top 10 Best Jobs of 2023
- (Reference: https://money.usnews.com/careers/best-jobs/rankings/the-100-best-jobs)

1. Software developer

Median salary: $120,730
Projected job growth by 2031: 370,600 positions
Education requirements: Bachelor's degree

2. Nurse practitioner

Median salary: $120,680
Projected job growth by 2031: 112,700 positions
Education requirements: Master's degree

3. Medical and health services manager

Median salary: $101,340
Projected job growth by 2031: 136,200 positions
Education requirements: Bachelor's degree

4. Physician assistant

Median salary: $121,530
Projected job growth by 2031: 38,400 positions
Education requirements: Master's degree

5. Information security analyst

Median salary: $102,600
Projected job growth by 2031: 56,500 positions
Education requirements: Bachelor's degree

6. Physical therapist

Median salary: $95,620
Projected job growth by 2031: 40,400 positions
Education requirements: Doctorate degree

7. Financial manager

Median salary: $131,710
Projected job growth by 2031: 123,100 positions
Education requirements: Bachelor's degree

8. IT Manager

Median salary: $159,010
Projected job growth by 2031: 82,400 positions
Education requirements: Bachelor's degree

9. Web developer

Median salary: $77,030
Projected job growth by 2031: 28,900 positions
Education requirements: Bachelor's degree

10. Dentist

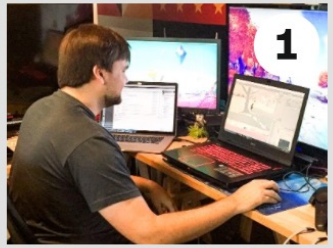
Median salary: $160,370
Projected job growth by 2031: 7,700 positions
Education requirements: Doctorate degree

Overview

- CNN-Money: Best Job 2017 (Reference: http://money.cnn.com/pf/best-jobs/)

Top Ten Jobs

See the full list



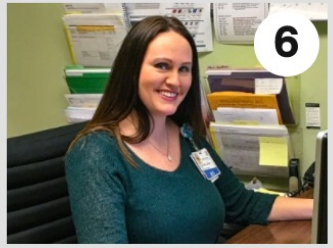
Mobile App Developer

Risk Management Director

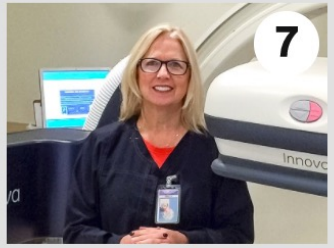
Landman

Product Analyst

Info Assurance Analyst

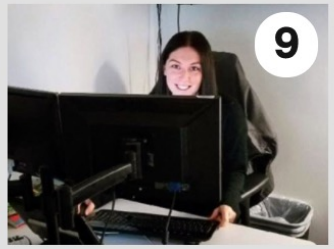


QA Coordinator



Clinical Applications Specialist

Hospital Administrator



Database Analyst

Director, Finance & Administration

Overview

- CNN-Money: Best Job 2015 (Reference: http://money.cnn.com/pf/best-jobs/)

Top Ten Jobs

See the full list

Software Architect

Video Game Designer

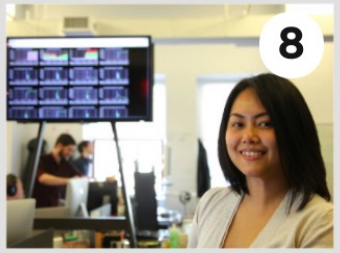
Landman

Patent Agent

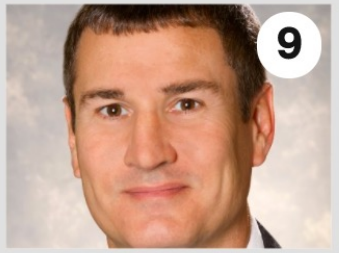
Hospital Administrator

Continuous Improvement Mgr.

Clinical Nurse Specialist



Database Developer



Info Assurance Analyst

Pilates/Yoga Instructor

Overview

- CNN-Money: Best Job 2011 (Reference: http://money.cnn.com/pf/best-jobs/)

Software Engineering

- Many systems are software controlled

- Software engineering is concerned with:
  - Theories
  - Methods
  - Tools
- Target: Professional software development

Software Costs

- Software costs often dominate computer system costs
  - Costs of software on a PC are often ***greater than*** the hardware cost
- Software costs more to maintain than it does to develop
- Software engineering is concerned with cost-effective software development

Software Types

Computer programs and associated documentation

- Ex. Requirements, Design models and User manuals

Software products may be developed for:

- A particular customer or a general market

Software products Types

- Generic
  - Developed to be sold to a range of different customers e.g. PC software such as Excel or Word
- Bespoke: a custom software
  - Developed for a single customer according to their specification

Software Engineering

Software engineering is:

"An engineering discipline that is concerned with all aspects of software production"

Software engineers use appropriate tools and techniques depending on:

- Problem to be solved
- Development constraints
- Resources available

What is the difference between software engineering and computer science?

- Computer science is concerned with theory and fundamentals
- Software engineering is concerned with the practicalities of developing and delivering useful software

- Computer science theories are still insufficient to act as a complete underpinning for software engineering

What is the difference between software engineering and system engineering?

System engineering concerns with all aspects of computer-based systems development including:

- Hardware
- Software
- Process

System engineers are involved in:

- System specification
- Architectural design
- Integration and deployment

Software engineering concerns with:

- Software infrastructure
- Control
- Applications
- Databases

What is a software process?

A set of activities whose goal is the development or evolution of software

Generic activities in all software processes are:

1. *Specification* - what the system should do and its development constraints
2. *Development* - production of the software system
3. *Validation* - checking that the software is what the customer wants
4. *Evolution* - changing the software in response to changing demands

Software Process Model?

A simplified representation of a software process, presented from a process perspective.

Examples of process perspectives:

- Workflow perspective - sequence of activities
- Data-flow perspective - information flow
- Role/action perspective - who does what

Generic process models

- Waterfall model
- Spiral model
- Component-based model

Costs of Software Engineering

60% of costs are development costs (roughly)

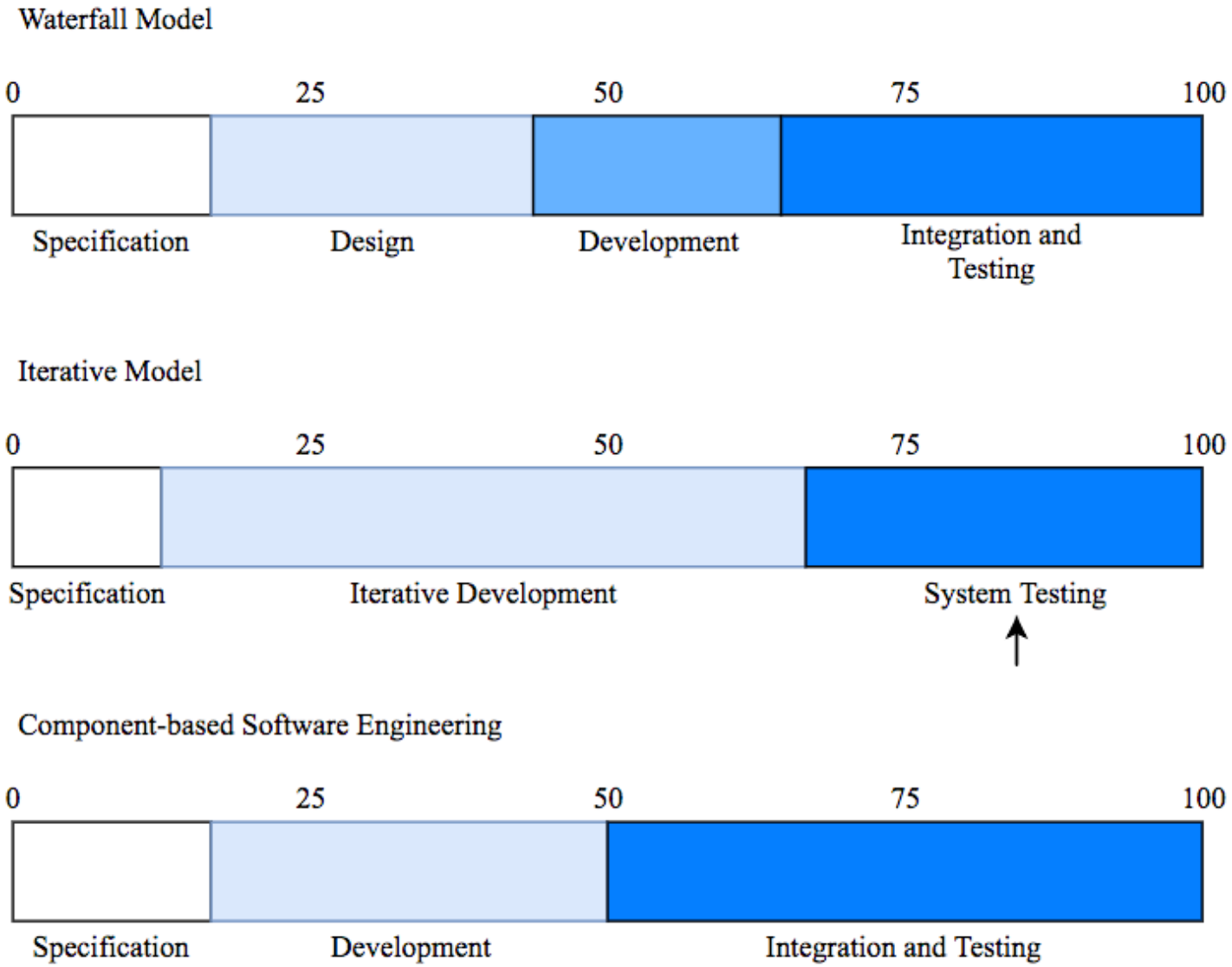
40% are testing costs

Costs vary depending on:

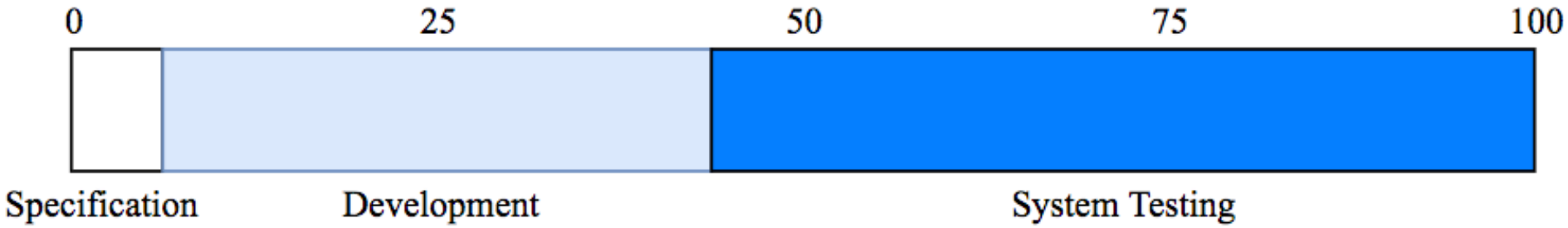
- Type of developed system
- Requirements of system attributes
  - Such as performance and system reliability

Distribution of costs depends on the development model

Activity Cost Distribution [1]



Product Development Costs



Software Engineering Methods

1. **Model descriptions**
   - Descriptions of graphical models which should be produced
2. **Rules**
   - Constraints applied to system models
3. **Recommendations**
   - Advice on good design practice
4. **Process guidance**
   - What activities to follow

Computer-Aided Software Engineering (CASE)

Software systems that are intended to provide automated support for software process activities.

Types of CASE Tools

- Upper-CASE Tool
  - Tools to support the early process activities of requirements and design
- Lower-CASE Tool
  - Tools to support later activities such as programming, debugging and testing

Attributes of Good Software

The software should deliver the required functionality and performance to the user and should be maintainable, dependable and acceptable

1. Maintainability
   - Software must evolve to meet changing needs
2. Dependability
   - Software must be trustworthy
3. Efficiency
   - Software should not make wasteful use of system resources
4. Acceptability
   - Software must accepted by the users for which it was designed.

Professional and ethical responsibility

Software engineering involves wider responsibilities than simply the application of technical skills

Software engineers must behave in an honest and ethically responsible way if they are to be respected as professionals

Ethical behavior is more than simply upholding the law

Issues of professional responsibility

Confidentiality

- Engineers should normally respect the confidentiality of their employers or clients irrespective of whether or not a formal confidentiality agreement has been signed

Competence

- Engineers should not misrepresent their level of competence. They should not knowingly accept work which is out with their competence

Issues of professional responsibility (cont.)

Intellectual property rights

- Engineers should be aware of local laws governing the use of intellectual property such as patents, copyright, etc.
- They should be careful to ensure that the intellectual property of employers and clients is protected

Computer misuse

- Software engineers should not use their technical skills to misuse other people's computers
- Computer misuse ranges from relatively trivial (game playing on an employer's machine, say) to extremely serious (dissemination of viruses)

System Components

- Software
- Mechanical, electrical and electronic hardware
- Be operated by people

System Categories

1. Technical computer-based systems
   - Include *hardware* and *software*
   - Operators and operational processes are *not considered to be part of the system* *(No interaction with human or user)*

2. Socio-technical systems
   - Include *technical systems*
   - Include operational processes and people who use and interact with the technical system
   - Socio-technical systems are governed by organizational *policies and rules*

Types of Emergent (Critical) Property

1. Functional properties
   - Appear when all the parts of a system work together to achieve some objectives
   - Ex. Bicycle has the functional property of being a transportation device once it has been assembled from its components

2. Non-functional emergent properties
   - Ex. *reliability, performance, safety, and security*
   - Relate to the behavior of the system in its operational environment
     - They are often critical for computer-based systems as failure to achieve some minimal defined level

Examples of emergent properties (Non-functional Properties)

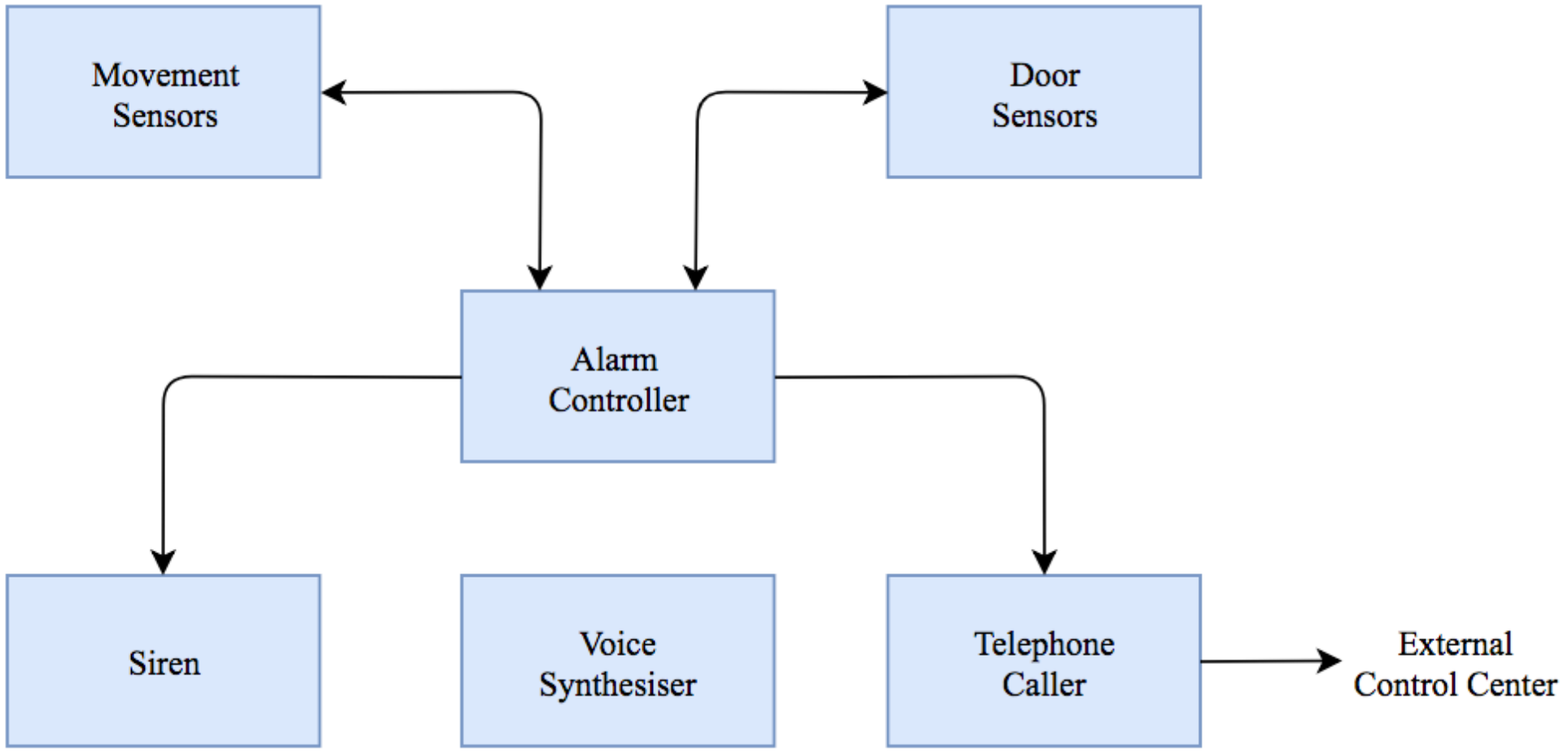
Property	Description
Volume	The volume of a system (the total space occupied) varies depending on how the component assemblies are arranged and connected.
Reliability	System reliability depends on component reliability but unexpected interactions can cause new types of failure and therefore affect the reliability of the system.
Security	The security of the system (its ability to resist attack) is a complex property that cannot be easily measured. Attacks may be devised that were not anticipated by the system designers and so may defeat built-in safeguards.
Repairability	This property reflects how easy it is to fix a problem with the system once it has been discovered. It depends on being able to diagnose the problem, access the components that are faulty and modify or replace these components.
Usability	This property reflects how easy it is to use the system. It depends on the technical system components, its operators and its operating environment.

System Architectural Model

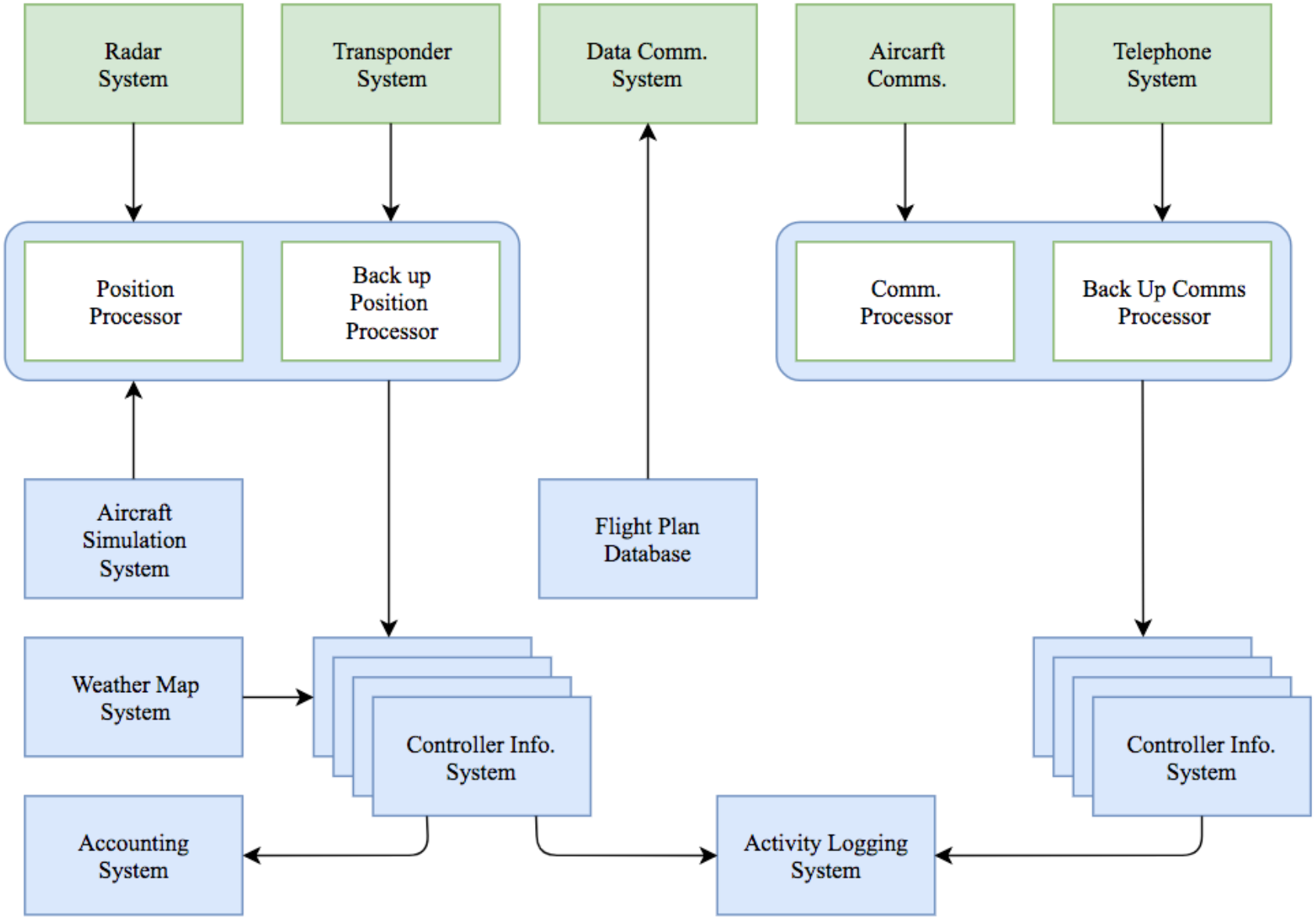
An abstract view of the subsystems making up a system

- Include major information and flow between sub-systems
- Presented as a block diagram
- Identify different types of functional components in the model

System Architectural Model: **Burglar Alarm System [1]**



System Architectural Model: **Airplane Travel Control [1]**



System Evolution

Large systems have a long lifetime.

- They must evolve to meet changing requirements

System Evolution

- Changes must be analyzed from:
  - Technical perspective
  - Business perspective

Legacy Systems : Existing systems which must be maintained

Legacy Systems

Socio-technical systems that have been developed using old or obsolete (not modern) technology

Crucial to the operation of a business

- Too risky to discard these systems
  - Such as Bank customer accounting system and Aircraft maintenance system

Legacy System Components

Hardware - may be obsolete (modern) mainframe hardware

Support software - may rely on support software from suppliers who are no longer in business

Application software - may be written in obsolete programming languages

Application data - often incomplete and inconsistent

Business processes - may be constrained by software structure and functionality

Business policies and rules - may be implicit and embedded in the system software

References

1. Ian Sommerville, Software Engineering 10th Edition, Pearson, April 2015

Any Questions?

:O)

Thank you