# Software Engineering (ESC501) - Project Planning & Project Estimation Techniques

- Prof. Poulami Dutta

# Intended Learning Outcomes (ILOs)

- Identify the job responsibilities of a software project manager.
- Identify the necessary skills required in order to perform software project management.
- Identify the essential activities of project planning.
- Determine the different project related estimates performed by a project manager and suitably order those estimates.
- Explain what is meant by Sliding Window Planning.
- Explain what is Software Project Management Plan (SPMP).
- Identify and explain two metrics for software project size estimation.
- Identify the shortcomings of function point (FP) metric.

# Organization of this Lecture

- Introduction to Project Planning
- Software Cost Estimation
    - Cost Estimation Models
    - Software Size Metrics
    - Empirical Estimation
    - Heuristic Estimation
    - COCOMO
- Staffing Level Estimation
- Effect of Schedule Compression on Cost
- Summary

# Introduction

- Many software projects fail:
  - due to faulty project management practices:
    - It is important to learn different aspects of software project management.
- Goal of software project management:
  - enable a group of engineers to work efficiently towards successful completion of a software project.

# Responsibility of Project Managers

- take the overall responsibility of steering a project to success.
- difficult to objectively describe the job responsibilities of a project manager.
- ranges from invisible activities like building up team morale to highly visible customer presentations.
- activities can be broadly classified into
  - **project planning** - project planning activity is undertaken before the development starts to plan the activities to be undertaken during development.
  - **project monitoring** and
  - **control activities** – undertaken once the development activities start with the aim of ensuring that the development proceeds as per plan and changing the plan whenever required to cope up with the situation.

# Responsibility of Project Managers (Contd.)

- Project proposal writing,
- Project cost estimation,
- Scheduling,
- Project staffing,
- Project monitoring and control,
- Software configuration management,
- Risk management,
- Managerial report writing and presentations, etc.

# Skills for Software Project Management

- theoretical knowledge,
- good qualitative judgment and decision making capabilities,
- good grasp of the latest software project management techniques such as cost estimation, risk management, configuration management,
- good communication skills and the ability to get work done,
- skills such as tracking and controlling the progress of the project, customer interaction, managerial presentations, and team building are largely acquired through experience.

# Project Planning Activities

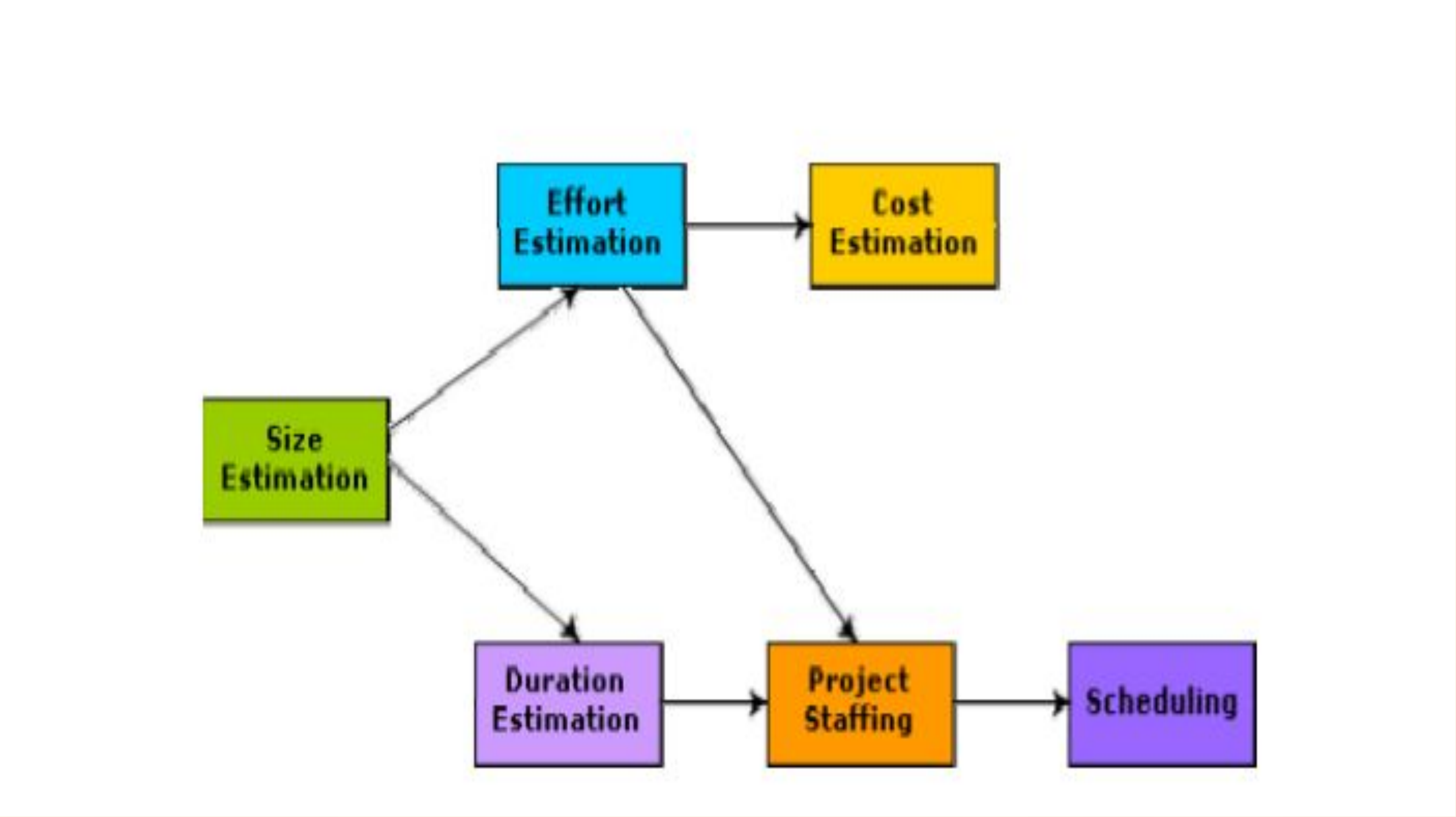
- **Estimation**:
  - **Project size**: What will be problem complexity in terms of the effort and time required to develop the product?

    **Cost**: How much is it going to cost to develop the project?

    **Duration**: How long is it going to take to complete development?

    **Effort**: How much effort would be required?
- **Project scheduling** of man power and other resources.
- **Staff organization**:
  - staffing plans
- **Risk handling**:
  - identification, analysis, and abatement procedures.
- **Miscellaneous plans**: quality assurance plan, configuration management plan, etc.

# Precedence of Project Planning Activities



# Sliding Window Planning

- Requires utmost care and attention since commitment to unrealistic time and resource estimates result in schedule slippage.
- Delays can cause customer dissatisfaction and adversely affect team morale. It can even cause project failure.
- For large projects, it is very much difficult to make accurate plans. A part of this difficulty is due to the fact that the proper parameters, scope of the project, project staff, etc. may change during the span of the project.
- To overcome this problem, sometimes project managers undertake project planning in stages.
  - Protects from making commitments too early.
  - Technique of staggered planning is known as **Sliding Window Planning**
    - Starts with an initial plan,
    - Project is planned accurately in the successive stages,
    - After the completion of every phase, the project managers can plan each subsequent phase more accurately and with increasing levels of confidence.

# SPMP (Software Project Management Plan)

- After planning is complete:
  - Document the plans:
    - in a Software Project Management Plan(SPMP) document.
    - List of different items.
  - SPMP

# Organization of SPMP Document

- **Introduction** (Objectives, Major Functions, Performance Issues, Management and Technical Constraints)
- **Project Estimates** (Historical Data, Estimation Techniques, Effort, Cost, and Project Duration Estimates)
- **Project Resources Plan** (People, Hardware and Software, Special Resources)
- **Schedules** (Work Breakdown Structure, Task Network, Gantt Chart Representation, PERT Chart Representation)
- **Staff Organization** (Team Structure, Management Reporting)
- **Risk Management Plan** (Risk Analysis, Risk Identification, Risk Estimation, Abatement Procedures)
- **Project Tracking and Control Plan**
- **Miscellaneous Plans** (Process Tailoring, Quality Assurance)

# Metrics For Software Project Size Estimation

- size of a problem is obviously not the number of bytes that the source code occupies
- neither is it the byte size of the executable code
- project size is a measure of the problem complexity in terms of the effort and time required to develop the product
- Currently two metrics are popularly being used widely to estimate size:
  - lines of code (LOC) and
  - function point (FP)

# Software Size Metrics

- **LOC (Lines of Code):**
  - Simplest and most widely used metric.
  - Comments and blank lines should not be counted.
- project size is estimated by counting the number of source instructions in the developed program.
- accurate estimation of the LOC count at the beginning of a project is very difficult.
- project managers usually divide the problem into modules, and each module into submodules and so on, until the sizes of the different leaf-level modules can be approximately predicted.
- using the estimation of the lowest level modules, project managers arrive at the total size estimation.

# Disadvantages of Using LOC

- Size can vary with coding style.
- Focuses on coding activity alone.
- Correlates poorly with quality and efficiency of code.
- Penalizes higher level programming languages, code reuse, etc.
- Measures lexical/textual complexity only.
  - does not address the issues of structural or logical complexity.
- Difficult to estimate LOC from problem description.
  - So not useful for project planning.

# Function Point Metrics

- **Function point metric (FP)** was first proposed by Albrecht in early 80's.
- FP can be used to:
  - Estimate the cost of effort required to design, code and test the software,
  - Predict the number of errors that will be encountered during testing,
  - Forecast the number of components and/or the number of projected source lines.
- FPs are derived using an empirical relationship based on countable measures of software's information domain and assessment of its quality.

# Function Point Metric – Information Domain Values (IDVs)

- **Number of External Inputs (EIs):**
  - A set of related inputs is counted as one input.
  - It originates from a user or is transmitted from another application.
- **Number of External Outputs (EOs):**
  - A set of related outputs is counted as one output.
  - It is derived within the application and provides information to the user.
  - It refers to reports, screens, error messages etc.
- **Number of External Inquiries (EQs):**
  - Each user query type is counted.
- **Number of Internal Logical Files (ILFs):**
  - Files are logically related data and thus can be data structures or physical files.
  - It resides within the application’s boundary.
- **Number of External Interface Files (EIFs):**
  - Data transfer to other systems.
  - Resides external to the application.

# Function Point Metric (Contd.)

- **Computing Function Points**
  - Once the data have been collected, the following table is computed and a complexity value is associated with each count.
  - Organizations develop a criteria for determining whether a particular entry is simple, average or complex.

| IDV | COUNT | X | SIMPLE | AVERAGE | COMPLEX | = | |
|---|---|---|---|---|---|---|---|
| EIs | | | 3 | 4 | 6 | | |
| EOs | | | 4 | 5 | 7 | | |
| EQs | | | 3 | 4 | 6 | | |
| ILFs | | | 7 | 10 | 15 | | |
| EIFs | | | 5 | 7 | 10 | | |
| Count Total | | | | | | | |

# Function Point Metric (Contd.)

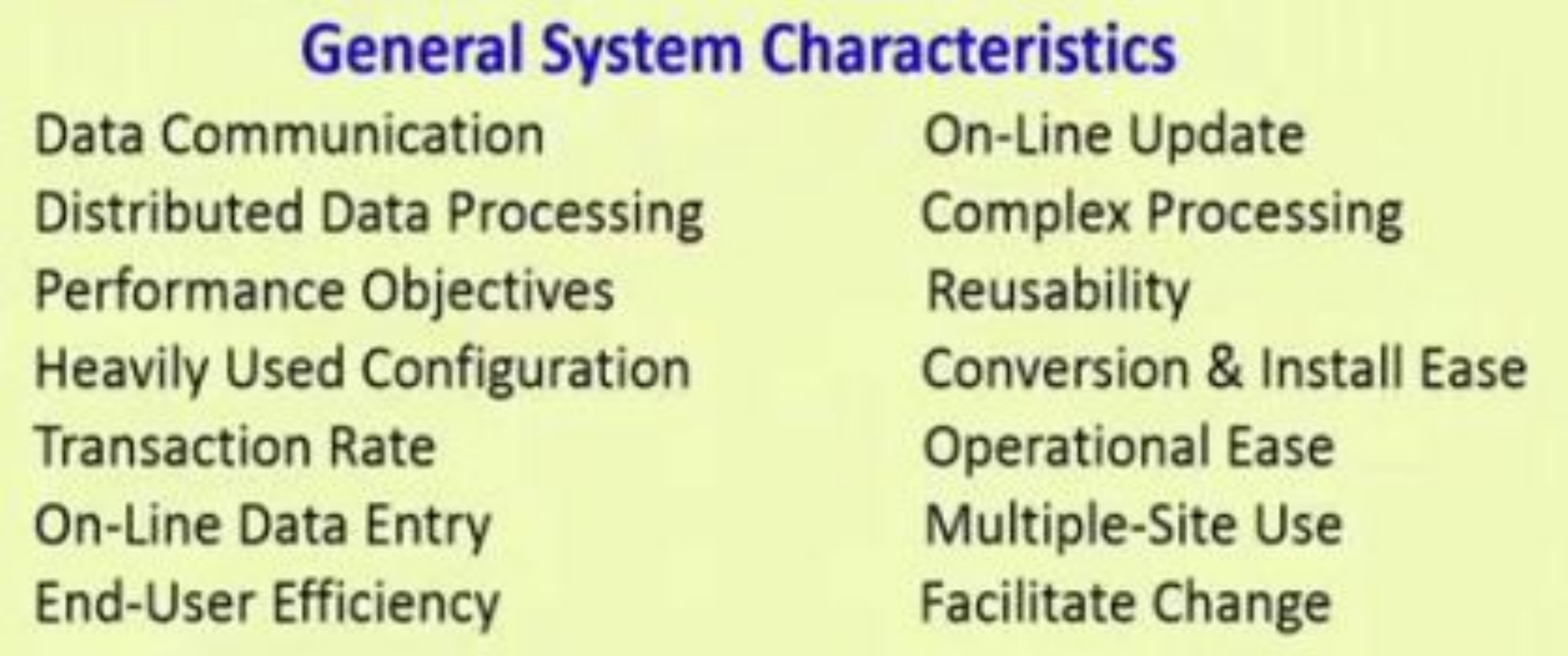
- **Computing Function Points**

  - FP = Count Total x [0.65 + 0.01 x Σ $(F_i)$], constants/weighing factors are empirically determined.

# Function Point Metric (Contd.)

- $F_i$ (i = 1 to 14) are Value Adjustment Factors (VAFs)/Technical Complexity Factors (TCFs) based on responses to the following questions.
  - Does the system require reliable backup and recovery?
  - Are specialized data communications required to transfer information to or from the application?
  - Are there distributed processing functions?
  - Is performance critical?
  - Will the system run in an existing, heavily utilized operational environment?
  - Does the system require online data entry?
  - Does the online data entry require the input transaction to be built over multiple screens or operations?
  - Are the ILFs updated online?
  - Are the inputs, outputs, files or enquires complex?
  - Is the internal processing complex?
  - Is the code designed to be reusable?
  - Are conversion and installation included in the design?
  - Is the system designed for multiple installations in different organizations?
  - Is the application designed to facilitate change and for ease of use by the user?

# VAFs/TCFs



## General System Characteristics

Data Communication

Distributed Data Processing

Performance Objectives

Heavily Used Configuration

Transaction Rate

On-Line Data Entry

End-User Efficiency

On-Line Update

Complex Processing

Reusability

Conversion & Install Ease

Operational Ease

Multiple-Site Use

Facilitate Change

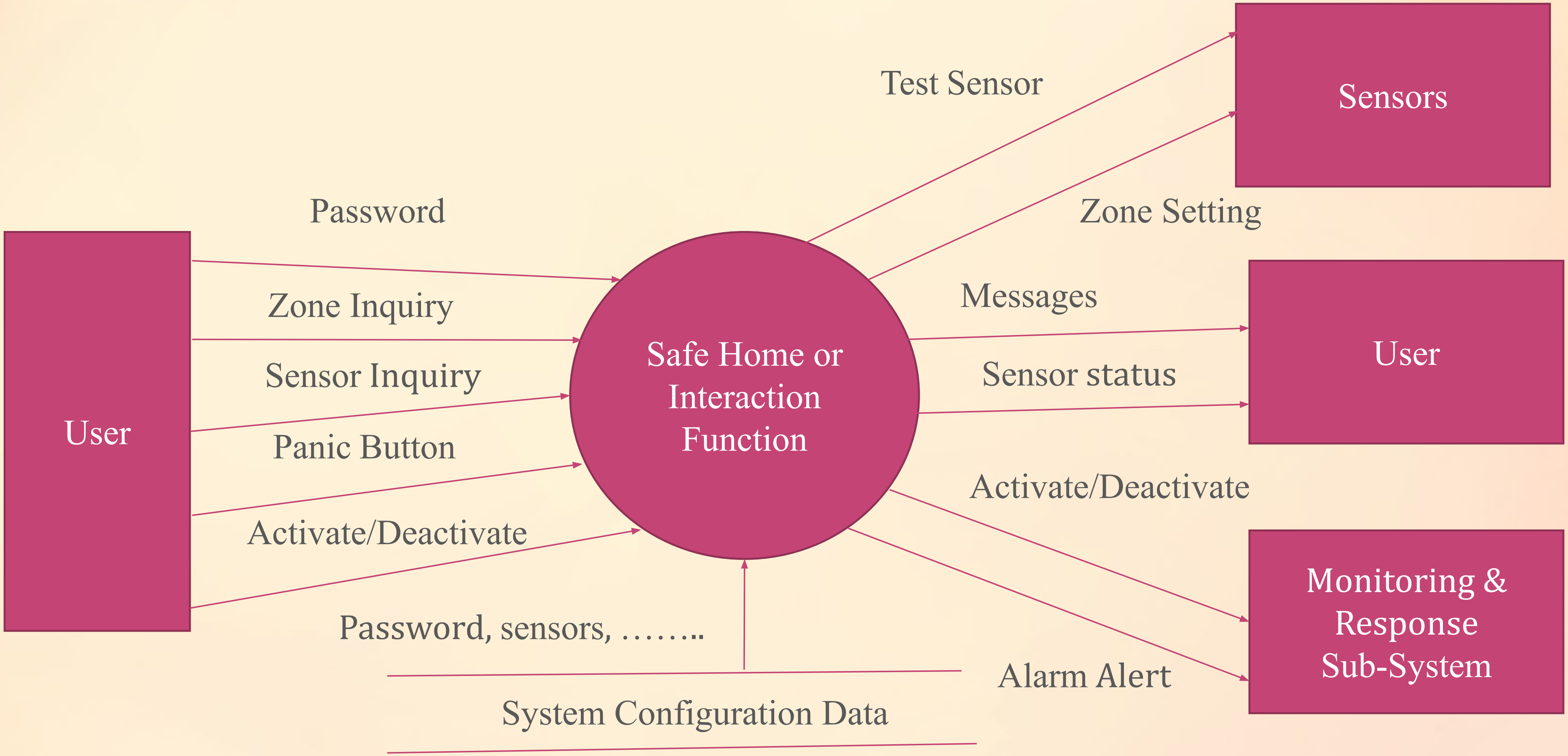
# Function Point Metric (Contd.)

- Degrees of Influence for VAFs/TCFs
  - Each of these questions is answered using a scale that ranges from 0 – 5
    - 0 – No Influence
    - 1 – Incidental
    - 2 – Moderate
    - 3 – Average
    - 4 – Significant
    - 5 - Essential

# Case Study I: Safe Home Software

- The function manages user interaction, accepting a user password to activate or deactivate the system and allows inquires on the status of security zones and various security sensors.
- The function displays a series of prompting messages and sends appropriate control signals to various components of the security system.

# Case Study I: Safe Home Software



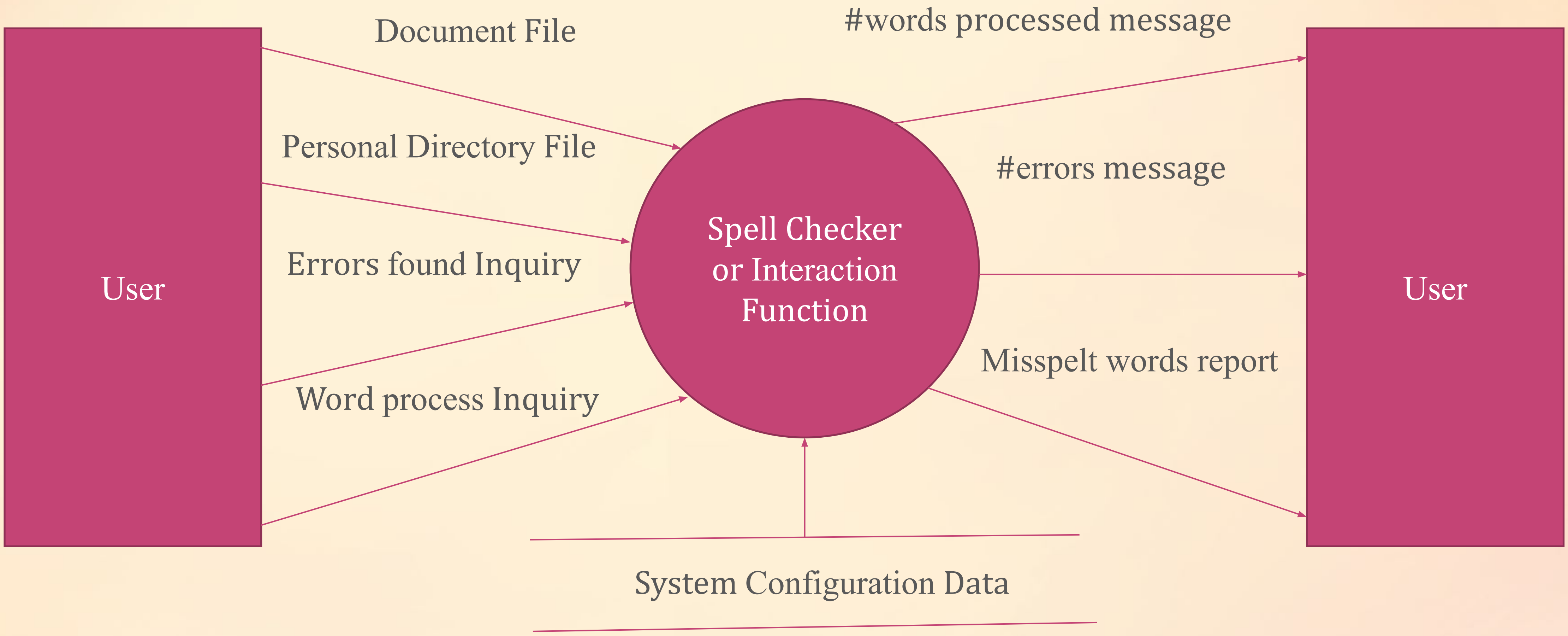
# Case Study I: Safe Home Software (Computing FP)

- Function__Point__Metric__Safe__Home__Software.xls

# Case Study II: Spell Checker Software

- The spell-checker accepts as input, a document file and an optional personal directory file.
- Then the spell checker lists all words which are not contained in either of these files.
- The user can query the number of words processed and the number of spelling errors which are found at any stage during the processing.

# Case Study II: Spell Checker Software



# Case Study II: Spell Checker Software (Computing FP)

- Function__Point__Metric__Spell__Checker__Software.xls

# Function Point Metric (CONT.)

- Suffers from a major drawback:
  - the size of a function is considered to be independent of its complexity.
- Extend function point metric:
  - Feature Point metric:
  - considers an extra parameter:
    - Algorithm Complexity.

# Function Point Metric (CONT.)

- Proponents claim:
  - FP is language independent.
  - Size can be easily derived from problem description.
- Opponents claim:
  - it is subjective --- Different people can come up with different estimates for the same problem.

# Case Study III: Mobile Application (Class Assignment)

- The activity has been designed and developed specifically to run on tablet computers for teaching autistic children.

- They are very important visual support systems for the autistic children to develop skills like getting organized, acting independently and making selections.

# Case Study IV: Payroll Application

- Transaction to input, amend and delete employee details.
- Transaction to calculate pay details from timesheet data.
- Transaction to print pay-to-date details for each employee.
- A file of payroll details for each employee.
- A personnel file maintained by another system that is accessed for name and address details.

THANK
YOU