CT-21015 Software Engg Mini Project Stage-II

Assignments/Quizzes followed by Oral	20 Marks
Mini Project(Submission in Stages)	40 Marks
Theory Exam	40 Marks

Prof. Harish D.G.
Dept. of Computer and IT
College of Engineering,Pune
www.harishgadade.com

Software Management and Metrics

- Outline:
  - Project Management
  - Software Metrics
  - Metrics for software quality
  - Integrating Metrics within software process

Project Management

- Effective project management focuses on three P's
  - People
  - Problem
  - Process

People

- Team Leader
  - Motivation
  - Organization
  - Ideas or Innovation
  - Problem Solving
  - Managerial Identity
  - Achievement

People

- Team Leader
  - Motivation
  - Organization
  - Ideas or Innovation
  - Problem Solving
  - Managerial Identity
  - Achievement
- Software Team
  - Democratic Decentralized(DD)
  - Controlled Decentralized(CD)
  - Controlled Centralized(CC)

Problems

- Before a project can be planned, its objectives and scope should be established, alternative solutions should be considered, technical and management constraints should be identified.
- Without this information, it is impossible to define reasonable(or accurate)estimates of cost; an effective assessment of risk; a realistic breakdown of the project task; or manageable schedule.
- The s/w developer and customer must meet to define project objectives and scope.

Problems

- Therefor, we must examine the problem at very beginning of the project. At a minimum, the scope of the problem must be established.
    - Software scope:
    - Problem Decomposition

The Process

- A s/w process provides the framework from which a comprehensive plan for s/w development can be established.
- The generic phases that characterize the software process are-
  - Definition
  - Development
  - Maintenance, are applicable to all softwares.
- The Project manager must decide which process model is most appropriate for the project,then define a preliminary plan based on the set of common process framework activities.

The Process

- **Melding the Problem and the Process:**

  Project planning begins with the melding of the problem and process.Each activity must pass through the set of framework activities like

  - Customer communication
  - Planning
  - Risk Analysis
  - Engineering
  - Construction and Release
  - Customer evaluation

- **Process Decomposition**

Software Metrics

- Software metrics refer to a broad range of measurements for computer software(It is a standard of measure)
- Measurement in physical world can be categorized in two ways
  - Direct Measures
  - Indirect Measures
- Direct measures of the software engineering process includes;
  - Cost
  - Line of Code(LOC) Produced
  - Execution Speed
  - Memory Size
- Indirect measures includes;
  - Functionality
  - Quality
  - Complexity
  - Efficiency
  - Reliability
  - Maintainability

Software Metrics

- Types of Software Metrics
  - Product Metrics
  - Process Metrics
  - Project Metrics

Software Metrics

- **Product Metrics**
- Describes the characteristics of the product
  - size
  - Complexity
  - Design features
  - Performance
  - Quality level
  - Reliability
  - Functionalities

- **Process Metrics**
- Used to improve the development process and maintenance activities of the software
  - Effort required
  - Time to produce the product
  - No of defects found
  - Tools and tech
  - Quality
  - Efficiency

- **Project Metrics**
- Describes the project characteristics and execution
  - No of s/w developers
  - Staffing patterns
  - Cost
  - Schedule
  - Productivity
  - Quality
  - Assess status of ongoing project

Efforts and Software Estimation Techniques

- Size-Oriented Metrics
- Function Oriented Metrics

Size-Oriented Metrics

- Measure size of the s/w produced
- Any lines in code apart from comments and blanks, header, declaration as well as executable and non executable statements.
- Normalized by counting in Thousand Lines Of Code(KLOC)
- Size oriented metrics of project computed by:
  - Error per KLOC
  - Defect per KLOC
  - Rs per KLOC
  - Pages of documentation per KLOC
- In addition, other metrics also be computed;
  - Errors per person-month
  - LOC per person-month
  - Rs/pages of documentation
- **Advantage:**
  - Easy to compute from developed code
- **Disadvantages:**
  - Programming Language Dependent
  - Penalize well-designed but short programs
  - Not Universally accepted.

Size-Oriented Metrics

Project	LOC	Efforts	Rs(000)	pp.doc.	Errors	Defects	People
Alpha	12,100	24	168	365	134	29	3
Beta	27,200	62	440	1224	321	86	5
Gamma	20,200	43	314	1050	256	64	6
•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•

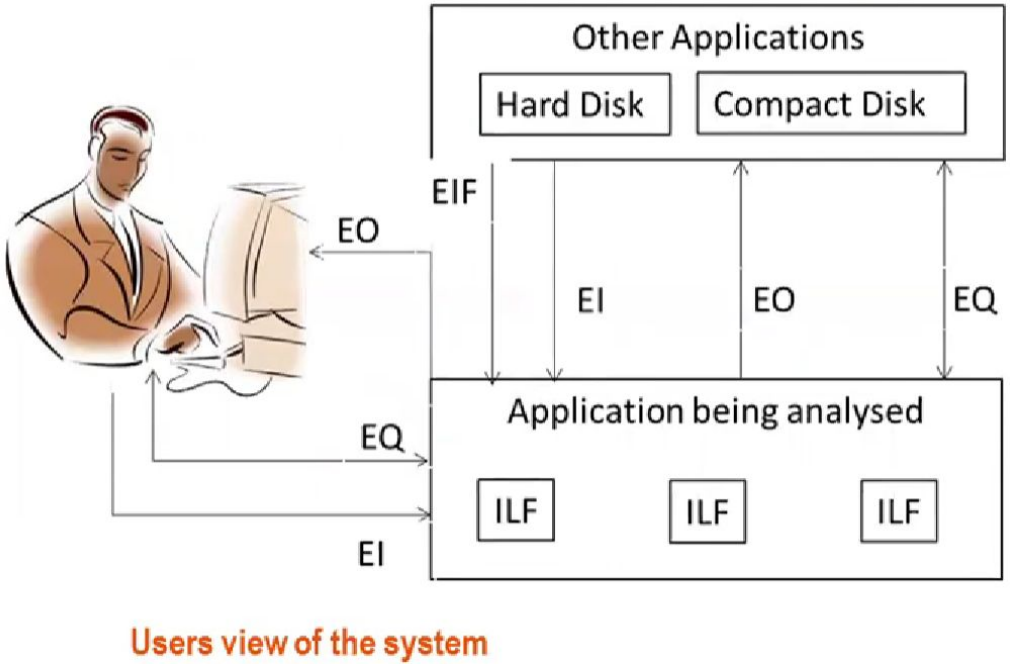
Function-Based Metrics

- Measure the functionality delivered by the application
- Most widely used metrics of this type is function point
- Using Historical data it can:
  - Estimates the cost or efforts required to design, code and test the software.
  - Predicts the number of errors that will be encountered during the testing.
  - Forecast the number of components or number of projected source lines in the implemented system.
- **Advantages:**
  - Programming language independent
  - Based on data that are more likely to be known in the early stage of a project.
- **Disadvantages:**
  - Computation is based on subjective data
  - Counts of the information domain can be difficult to collect
  - Has no direct physical meaning, its just a number

Function-Based Metrics

Principle of FPA

- System is decomposed into five functional units
  - External Inputs (EI)
  - External Outputs (EO)
  - External Enquiries (EQ)
  - Internal Logical Files (ILF)
  - External Interface Files (EIF)



Function-Based Metrics

Calculations

- FPA = UFP*CAF
- $UFP = \sum_{i=1\ j=1}^{i=5,j=3} W_{ij} * C_{ij}$
- $CAF=0.65+[0.01*\sum F_i]$

Function Point Analysis (FPA)
Unadjusted Function Point (UFP)
Complexity Adjustment Factor (CAF)
Weighting Factor (W)
Count (C)
Total Complexity Adjustment Value (F)

FPA Matrix

Information Domain Value	Weighting factor		
	Simple	Average	Complex
External Inputs (EIs)	3	4	6
External Outputs (EOs)	4	5	7
External Inquiries (EQs)	3	4	6
Internal Logical Files (ILFs)	7	10	15
External Interface Files (EIFs)	5	7	10

Function-Based Metrics

Calculations

- FPA = UFP*CAF
- $UFP = \sum_{i=1\ j=1}^{i=5,j=3} W_{ij} * C_{ij}$
- $CAF=0.65+[0.01*\sum F_i]$

Function Point Analysis (FPA)
Unadjusted Function Point (UFP)
Complexity Adjustment Factor (CAF)
Weighting Factor (W)
Count (C)
Total Complexity Adjustment Value (F)

FPA Matrix

Information Domain Value	Weighting factor		
	Simple	Average	Complex
External Inputs (EIs)	3	4	6
External Outputs (EOs)	4	5	7
External Inquiries (EQs)	3	4	6
Internal Logical Files (ILFs)	7	10	15
External Interface Files (EIFs)	5	7	10

- ***Calculate FPA if all CAF and WAF are average for the following count values EI=10, EO=30, EQ=50, EIF=10, ILF=20***

Function-Based Metrics

Questionnaires for software development

1. Does the system require reliable backup and recovery? (3)
2. Are specialized data communications required to transfer information to or from the application?
3. Are there distributed processing functions?
4. Is performance critical?
5. Will the system run in an existing, heavily utilized operational environment?
6. Does the system require online data entry?
7. Does the online data entry require the input transaction to be built over multiple screens or operations?
8. Are the ILFs updated online?
9. Are the inputs, outputs, files, or inquiries complex?
10. Is the internal processing complex?
11. Is the code designed to be reusable?
12. Are conversion and installation included in the design?
13. Is the system designed for multiple installations in different organizations?
14. Is the application designed to facilitate change and ease of use by the user?

Significance Values

- **0- Not present/ No influence**
- **1- Incidental**
- **2- Moderate**
- **3- Average**
- **4- Significant**
- **5- Essential**

Function-Based Metrics

Calculations

- FPA = UFP*CAF
- UFP = $\sum_{i=1\ j=1}^{i=5,j=3} W_{ij} * C_{ij}$
- CAF=0.65+[0.01*$\sum F_i$]

Function Point Analysis (FPA)
Unadjusted Function Point (UFP)
Complexity Adjustment Factor (CAF)
Weighting Factor (W)
Count (C)
Total Complexity Adjustment Value (F)

FPA Matrix

Information Domain Value	Weighting factor		
	Simple	Average	Complex
External Inputs (EIs)	3	4	6
External Outputs (EOs)	4	5	7
External Inquiries (EQs)	3	4	6
Internal Logical Files (ILFs)	7	10	15
External Interface Files (EIFs)	5	7	10

- ***Calculate FPA if all CAF and WAF are average for the following count values EI=10, EO=30, EQ=50, EIF=10, ILF=20***

Step1: Calculate UFP = $\sum_{i=1\ j=1}^{i=5,j=3} W_{ij} * Ci_j$

UFP=(10*4)+(30*5)+(50*4)+(10*7)+(20*10)
UFP=40+150+200+70+200
UFP=660

- **Step 2: Calculate** CAF=0.65+[0.01*$\sum F_i$]

CAF=0.65+[0.01*14*3]
CAF=1.07

Step 3: Calculate FPA=UFP*CAF
FPA=660*1.07
FPA=706.2

Function-Based Metrics

Questionnaires for software development

1. Does the system require reliable backup and recovery? (3)
2. Are specialized data communications required to transfer information to or from the application?
3. Are there distributed processing functions?
4. Is performance critical?
5. Will the system run in an existing, heavily utilized operational environment?
6. Does the system require online data entry?
7. Does the online data entry require the input transaction to be built over multiple screens or operations?
8. Are the ILFs updated online?
9. Are the inputs, outputs, files, or inquiries complex?
10. Is the internal processing complex?
11. Is the code designed to be reusable?
12. Are conversion and installation included in the design?
13. Is the system designed for multiple installations in different organizations?
14. Is the application designed to facilitate change and ease of use by the user?

Significance Values

- **0- Not present/ No influence**
- **1- Incidental**
- **2- Moderate**
- **3- Average**
- **4- Significant**
- **5- Essential**