Interactive Learning : Think Pair share  Academic year: 2019-20

Faculty Name: G. Kalpana

Subject: Software Engineering

Topic: Function Point Calculation

Participants: Students of II/II D section

Think-pair-share (TPS) is a collaborative learning strategy where students work together to solve a problem or answer a question about an assigned reading. This strategy requires students to (1) think individually about a topic or answer to a question; and (2) share ideas with classmates. Discussing with a partner maximizes participation, focuses attention and engages students in comprehending the reading material.

Content:

- **Function-based metrics**: use the function point as a normalizing factor or as a measure of the “size” of the specification
- **Specification metrics**: used as an indication of quality by measuring number of requirements by type

Function-Based Metrics

- The function point metric (FP), first proposed by Albrecht, can be used effectively as a means for measuring the functionality delivered by a system.
- Function points are derived using an empirical relationship based on countable (direct) measures of software's information domain and assessments of software complexity
- Information domain values are defined in the following manner:
  - Number of external inputs (EIs)
  - Number of external outputs (EOs)
  - Number of external inquiries (EQs)
  - Number of internal logical files (ILFs)
  - Number of external interface files (EIFs)
Function Points

To compute function points (FP), the following relationship is used:

$$FP = \text{count total} \times [0.65 + 0.01 \times \sum (F_i)] \quad (1)$$

Where count total is the sum of all FP entries obtained from figure.

The $F_i$ (i = 1 to 14) are value adjustment factors based on responses to the following questions:

Implementation: As part of this activity students are asked to find out the function point for given sample values Using the above formula.

1. Compute the function point value for a project with the following information domain characteristics:
   (1) No. of user inputs = 24
   (2) No. of user outputs = 65
   (3) No. of user inquiries = 12
   (4) No. of files = 12
   (5) No. of external interfaces = 4
   Assume all complexity adjustment values are moderate and 14 algorithms have been counted.

Solution:

<table>
<thead>
<tr>
<th>Measurement Parameter</th>
<th>Count</th>
<th>Weighting factor</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Number of external inputs (EI)</td>
<td>24</td>
<td>*</td>
</tr>
<tr>
<td>2. Number of external outputs (EO)</td>
<td>65</td>
<td>*</td>
</tr>
<tr>
<td>3. Number of external inquiries (EQ)</td>
<td>12</td>
<td>*</td>
</tr>
<tr>
<td>4. Number of internal files (ILF)</td>
<td>12</td>
<td>*</td>
</tr>
<tr>
<td>5. Number of external interfaces (EIF)</td>
<td>4</td>
<td>*</td>
</tr>
</tbody>
</table>

Count-total $\to$ 617

Now $F_i$ for moderate case = 2.

So sum of all $F_i$ (i = 1 to 14) = 14 * 2 = 28
FP = Count-total * [0.65 + 0.01 * S (Fi)] = 617 * [0.65 + 0.01 * 28] = 617 * [0.65 + 0.28] =
617 * 1.23 = 758.91 = 759

**Outcome:** All the students groups are actively participated in this activity and i gave few suggestions for those who are facing difficulty in solving the above problem with that they understood the topic easily and completely.

(Faculty Incharge)  (CSE-HOD)