CIS W338

Introduction to **PSP^SM**

Lecture 2

Need for Product Plans

- **Watt’s IBM Experience**
  - No Product plans:
    - Projects late
  - Product plans
    - Projects on time
- Planning provides sound basis for committing to completion dates
- Planning allows for coordination among team members and interfacing organizations
What is a Product Plan

- The size and important features of the product
- An estimate of the time required to do the work
- A projection of the schedule

Planning Small Jobs

- Planning is a skill that will improve with practice
- Simple plans for simple jobs, more complicated plans for larger ones.
  - Initially we will make very simple plans
Program Plan Form

Project Plan Summary

Student
Program
Instructor

<table>
<thead>
<tr>
<th>Summary</th>
<th>Plan</th>
<th>Actual</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minutes/LOC</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LOC/Hour</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Program Size (LOC):

<table>
<thead>
<tr>
<th>Total New &amp; Changed</th>
<th>Plan</th>
<th>Actual</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum Size</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Minimum Size</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Time in Phase (min.):

<table>
<thead>
<tr>
<th>Total</th>
<th>Plan</th>
<th>Actual</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum Time</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Minimum Time</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Product Plan Summary Instructions

Project Plan Summary Instructions

<table>
<thead>
<tr>
<th>Project</th>
<th>Program Name</th>
<th>Instructor</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Project</th>
<th>Program Name</th>
<th>Instructor</th>
</tr>
</thead>
</table>

8/1/2001 © 1999 P. Wolfgang
The Product Planning Process

• Not an exact process
• Will improve with practice and available data
• Compare what you plan to do with what you have done before

Program Size

• Basic unit of program size is the “Line Of Code”
• Comments are generally excluded
• Two types:
  – Physical
    • Number of “new lines” in the file
  – Logical
    • Independent of layout
Program Size -- Example 1

-- comment
if (X_Average >= 100) then
    Size := X_Average;
else
    Size := X_Average / 2;
end if;

Program Size -- Example 2

if (X_Average >= 100) then
    Size := X_Average;
else
    Size := X_Average; end if;
Program Size -- Example 3

// comment
if (X_Average >= 100)
    Size = X_Average;
else
    Size = X_Average;

Program Size -- Example 4

if (X_Average >= 100) Size = X_Average;
else Size = X_Average/2;
Program Size -- Example 5

```cpp
while (n > 0)
{
    s.push(n);
    cout << "Enter a positive integer. \n";
    cout << "Enter 0 to stop. \n";
    cin >> n;
}
while (!s.empty())
{
    cout.width(8);
    cout << s.top();
    s.pop();
}
```

Coding Guideline

- Layout code so that a physical line corresponds to a logical line
- if and while statements count as two statements
- for statements count as 4 statements
- compound statements count as the number of statement they contain
- place ‘{’ and ‘}’ on their own lines, and do not count them
Cautions on Using Size Measures

• Even following these guidelines, two equivalent programs may have different LOC counts.
• Complexity is not considered.
• Only count the lines your create.
• No consensus on how to count effort involved in using a code generating environment such as Visual C++ or Visual Basic.

Estimating Program Size

• No program to count until you have written it.
• Compare program to ones you have previously written
• Rank the new program relative these
• Based upon your opinion where the new program is likely to fall, estimate the new size.
Making a Larger Size Estimate

- List functions and procedures separately
- Establish categories of functions. E.G.
  - text, control, logic, display
- For the new program, estimate the functions (i.e.,
  do a quick high level design)

### Size Estimating Form

<table>
<thead>
<tr>
<th>Program, LOC, Prior Functions, Estimated Functions, Min, Avg, Max</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

**Comments:**

---

8/1/2001 © 1999 P. Wolfgang
## Size Estimating Form -- Example

### Program Size Estimating Form

<table>
<thead>
<tr>
<th>Program</th>
<th>LOC</th>
<th>Prior Functions</th>
<th>Estimated Functions</th>
<th>Min</th>
<th>Avg</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>Loops</td>
<td>4</td>
<td>Simple while</td>
<td>10</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>Medium do-while</td>
<td>14</td>
<td>7</td>
<td>11</td>
<td>14</td>
</tr>
<tr>
<td>Case</td>
<td>2</td>
<td>small case</td>
<td>11</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>larger case</td>
<td>14</td>
<td>5</td>
<td>8</td>
<td>11</td>
</tr>
<tr>
<td>Data</td>
<td>18</td>
<td>small linked list</td>
<td>14</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Estimate</td>
<td></td>
<td></td>
<td></td>
<td>12</td>
<td>19</td>
<td>25</td>
</tr>
</tbody>
</table>

Comments:

---

## Time Estimating

- Initially you will have no basis -- use your best judgement
- As you collect more data, use your average productivity rate (minutes / LOC)
- With larger data sample, you can use more sophisticated statistical methods.
Process for the First 4 Lab Assignments

1. Make an estimate of the size of the program (See Chapter 6 of Humphrey)
2. Fill out the plan portion of the Project Plan Summary. 
   (Note: for the first program you will have no existing code that you can re-use, so all of the code will be new.)
3. Write and test the program.
4. Fill out the actual portion of the Project Plan Summary. (A program to count actual lines of code will be provided.)