Why We Use Processes

- Many organizations do not know their current performance.
- Without this data, it is impossible to know if they are getting better or worse.
- Having a defined process and collecting data about that process is the first step towards process improvement.
What is a Process

• A process is:
  – A defined set of steps for doing a job
  – Each step or phase has a specified entry criteria that must be met before starting.
  – Each step or phase has a specified exit criteria that must be met before the step is considered complete.

• The quality of the process generally determines the quality of the product.

Some Definitions

• A product is something you produce for a co-worker, an employer, or customer.
• A project typically produces a product.
• A task is a defined element of work.
• A process defines the way to do projects.
• Processes have various phases or steps, like planning, developing, and testing.
• A process phase could be composed of multiple tasks or activities.
• Note a process may have one or many phases and a phase may have one or many tasks or activities.
**Process Script**

<table>
<thead>
<tr>
<th>PROCESS SCRIPT</th>
<th>Purpose</th>
<th>To guide you in developing small programs</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Entry Criteria</strong></td>
<td>The problem description.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Project Plan Summary form.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Actual size, time, and defect data for previous programs.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Time Recording Log.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Defect Recording Log.</td>
<td></td>
</tr>
<tr>
<td>1 Planning</td>
<td>Obtain a description of the program functions.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Estimate the Max., Min., and total LOC required.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Determine the Minutes/LOC.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Calculate the Max., Min., and total development times.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Enter the plan data in the Project Plan Summary form.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Record the planning time in the Time Recording Log.</td>
<td></td>
</tr>
<tr>
<td>2 Design</td>
<td>Design the program.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Record the design in the specified format.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Record design time in the Time Recording Log.</td>
<td></td>
</tr>
<tr>
<td>3 Code</td>
<td>Implement the designs.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Use a standard format for entering the code.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Record coding time in the Time Recording Log.</td>
<td></td>
</tr>
<tr>
<td>4 Compile</td>
<td>Compile the program.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Fix and record all defects found.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Record compile time in the Time Recording Log.</td>
<td></td>
</tr>
<tr>
<td>5 Test</td>
<td>Test the program.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Fix and record all defects found.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Record testing time in the Time Recording Log.</td>
<td></td>
</tr>
<tr>
<td>6 Postmortem</td>
<td>Complete the Project Plan Summary form with actual time, size, and defect data.</td>
<td></td>
</tr>
<tr>
<td><strong>Exit criteria</strong></td>
<td>A thoroughly tested program.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>A properly documented design.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>A complete program listing.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>A completed Project Plan Summary.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Completed time and defect logs.</td>
<td></td>
</tr>
</tbody>
</table>
Planning

- First obtain the requirements for the project and then complete the Project Plan Summary form. Finally, enter the time you took to do this planning in the Time Recording Log.

Entry Criteria

- The problem description.
- Project Plan Summary form.
- Actual size, time, and defect data for previous programs.
- Time Recording Log.
- Defect Recording Log.
Design

- Design the program. While a design need not be elaborate, think though the program logic before starting to write the code. Record the design some format (flowchart, pseudo code, etc.) Record the time spent in the Time Recording Log.
- Design methodology will be discussed in the course and the requirements for the design of later programs will be more specific.

Code

- Implement the design in the selected programming language. Use a consistent coding format. (Coding standards will be introduced later.)
- At the end of the coding phase, record the time in the Time Recording Log.
Compile

• Compile the program; correct and record all defects found. Continue until the program compiles correctly without any error messages.
• While it may seem inefficient, you should fix only one defect at a time, and then re-compile to find the next defect. One defect can produce numerous error messages.
• Record the defects in the Defect Recording Log and time in the Time Recording Log.

Test

• Run enough tests to ensure that the program meets all its requirements. All of the time spent in this phase is counted as testing time, including time to modify the program to correct defects.
• Record the defects in the Defect Recording Log and the time in the Time Recording Log.
Postmortem

• Complete the actual entries in the Project Plan Summary form.
• Since you record the postmortem time before you have actually completed the phase, complete as much work as possible and then allow a few minutes to complete the calculations.

Exit Criteria

• A thoroughly tested program.
• A properly documented design.
• A complete program listing.
• A completed Project Plan Summary.
• Completed time and defect logs.
What is Software Quality

- A quality product is one that meets its user’s needs.
- It must do this reliably and consistently.
- It should be free of unexpected and harmful behavior.

What are Defects

- A defect is something that is wrong with a program.
  - Syntax error
  - Spelling mistake
  - Punctuation mistake
  - Incorrect program statement
- Defects are objective — something that can be identified and counted.
Relative Impact of Defects

- A simple coding mistake can produce very destructive or hard to find defect.
- A sophisticated design defect may be easy to find.
- Most software defects are the result of simple oversights and mistakes.

Errors vs. Defects

- Errors are incorrect things that people do.
- Defects are defective elements of programs.
- People make errors and programs have defects.
- To reduce the number of defects, you must change what you do.
Bugs vs. Defects

• The term bug is often used instead of the term defect.
• This tends to trivialize a potentially critical problem.
• Defects are more like time bombs or land mines.
• If we could identify the important ones, we could only fix them and dismiss the others. Unfortunately, we cannot.

Project Plans

• The project plan defines the work and how it will be done.
• It provides a definition of
  – Each major task
  – Time and resources required
  – Framework for management review and control.
• It is a powerful learning vehicle.
• It is a benchmark to compare with actual performance.
The Need for Project Plans

• The project plan is an essential part of a project.
• The larger the project, the more important it is to have a plan.
• In this course, you will make project plans to gain experience.
• As you gain experience, your planning ability will improve.