WHAT IS SOFTWARE MANAGEMENT?

The only unforgivable failure is the failure to learn from past failure

*(Tom DeMarco)*

WHAT IS MANAGED?

- **Schedule** – milestones, tasks sequencing. *Is progress being made?*
- **Cost** – cost allocation to work tasks (WBS).
  *Is project (or individual task) within budget?*
- **Resources** – computational, facilities, personnel recruitment and assignment, management and team organization.
- **Quality** – procedures for SQA, CM, requirements management. Quantitative (statistical) methods. *Will the product be satisfactory?*
- **Risk** – recognizing what can go wrong, and defining steps to take when risks materialize.
- **Process** – choice of life cycle model (development paradigm), TQM or continuous process improvement. SEI capability maturity model.
- **Technology** – use of new ideas, methods, tools for development and/or management.

WHAT ARE THE MAJOR MANAGEMENT FUNCTIONS?

- **Planning** – estimation, proposal writing (draft technical/mgmt plan)
- **Organizing** – deployment of resources, schedules, milestones
- **Monitoring** – regular status reporting, troubleshooting, risk mgmt
- **Control** – support structures; intervention when *indicators* signal pending trouble.

“You can’t control what you can’t measure.”  *(Tom DeMarco)*

In a manager, individual experience is a valuable asset. Corporate experience is even more valuable.
A step in the classical waterfall software lifecycle: A close-up view

Verification is hard!: "Have Outputs_i been derived correctly from Inputs_i?"

- Different representations \(\Rightarrow\) can't automate "equivalence" checking
- Transformation \(\text{Phase } i: \text{in} \rightarrow \text{out}\) is not formal (i.e., automatable).
- Verification process is itself a fallible, human process

THINGS ADDRESSED AT EACH PHASE OF LIFECYCLE:

- Who works on this phase? What skill level is required?
- What support functions are needed (e.g., CM, SQA)?
- What inputs are required to produce the outputs? Are any missing? extra?
- What outputs are produced in this phase? Can any be eliminated?
- Where can automated tools be applied during this phase?
- What engineering techniques can be used to produce the outputs?
- When is this phase completed? How can one tell?
- What value system is used to judge/evaluate the goodness of the outputs?
- What indicators will be used by management to judge project health?
- What management techniques can be applied to enhance the outputs of this phase, or reduce schedule or cost?
- How can the process for this phase be improved?
CURRENT THINKING – Process Management

Software Engineering Institute’s (SEI) Capability Maturity Model (CMM)

“The quality of a software system is governed by the quality of the process used to develop and maintain it.” (SEI, 1990)

Empirical evidence and projections(*)

<table>
<thead>
<tr>
<th>CMM Level</th>
<th>Quality (defect/K)</th>
<th>Productivity (LOC/hr)</th>
<th>Project Cost</th>
<th>Unit Cost ($/LOC)</th>
<th>Schedule (months)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>9+</td>
<td>1</td>
<td>66M</td>
<td>$66</td>
<td>40</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>3</td>
<td>15M</td>
<td>$30</td>
<td>32</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>5</td>
<td>7M</td>
<td>$14</td>
<td>25</td>
</tr>
<tr>
<td>4*</td>
<td>0.3</td>
<td>8</td>
<td>3M</td>
<td>$6</td>
<td>19</td>
</tr>
<tr>
<td>5*</td>
<td>0.1-</td>
<td>12</td>
<td>1M</td>
<td>$2</td>
<td>16</td>
</tr>
</tbody>
</table>

Total Quality Management

- Organizational culture directed toward improving the process based on lessons learned
- Focus on maximizing quality for internal/external customers (consumers) of all products flowing in an organization
- Empowerment of users of the process

Zero–Defect Software Processes

- Cleanroom techniques – “formal methods” + statistical process control (Harlan Mills)

Risk Management

- Spiral and evolutionary project management paradigms
- Techniques for risk assessment, monitoring, control (Boehm, Gilb)

Reuse, Open Systems, and Other Emerging Technologies

What is Software Engineering/Management?
SOFTWARE ENGINEERS / MANAGERS ARE CHALLENGED TO:

- Improve ability to estimate software project costs and schedules.
- Identify, use and improve good software processes.
- Develop sound engineering methods for capturing, describing and validating user needs.
- Understand how to manage software risks.
- Develop ways of quantitatively describing desirable attributes of software processes and products (e.g., goodness criteria for work products).
- Understand how to control the software engineering process similar to industrial process control (e.g., chemical plant or production line).
- Identify indicators of progress for a software engineering activity.
- Understand the impact of making changes to the process and to particular work products.
- Develop ways of capturing and institutionalizing application expertise, making an organization as "smart" as its smartest people.
- Evaluate and incorporate beneficial software engineering technology.
- Learn how to select methods and tools for an optimal software engineering environment.

What is Software Engineering/Management?