Management

An overview of problem, process & product

⇒ Chapters 3 - 9
⇒ Concepts
⇒ Metrics - measurements to direct management
⇒ Planning
⇒ Risk Analysis
⇒ Quality Assurance
⇒ Configuration management

Concepts

People, product, process, project

⇒ People
  ⇒ Senior management
  ⇒ Technical management
  ⇒ Engineers & scientist - technical personnel
  ⇒ Customers
  ⇒ End users

Management

As many approaches as managers

⇒ Decentralized leadership
  - democratic (group, consensus) - decisions by agreement, no designated decision makers.
  - communication - mainly within group and by group members with upper leadership
⇒ Centralized leadership
  - hierarchical - may be multi-level
  - leader with designated authority and responsibility.
  - decisions made by leader - consults with team members
  - communication - formal to and through team leader.
  - leader may establish interface with team members
⇒ Combination - mix between decentralized and centralized
⇒ Adversarial - find other teams’ errors
⇒ Management by walking around - additional insight
Management (cont’d)

Reasons for selection

⇒ The normal management structure of the organization
⇒ Difficulty of the problem
⇒ Personnel familiarity - past activity & project estimated time
⇒ Modularity of the problem & product
⇒ Many other aspects of the problem
⇒ Delivery schedule
⇒ Socializability of the project
May need to revise management approach during project

Management (cont’d)

Coordination and communication

⇒ Formal - impersonal
  ⇒ designated documents, tech memos, milestones, schedules, error tracking reports, repository, etc.
⇒ Formal - interpersonal
  ⇒ focus on quality assurance activities, status reviews, code reading and inspection meetings
⇒ Informal
  ⇒ collocation - with encouragement to “water-cooler” activities
  ⇒ communications with persons with experience and insight
In most cases all types of communication exist to some degree

Management

Project - process decomposition

⇒ Customer communication - need definition, prel. approach, justification
⇒ Planning - requirements, preliminary design, solution approach(s)
⇒ Risk analysis - solution criticality, need and failure problems
⇒ Design - decomposition, modularization, interface definitions
⇒ Construction - coding, using components, packages, frameworks, libraries, etc.
⇒ Test and evaluation - unit, I/O module testing, white and black box
⇒ Customer release and evaluation
Project definition

WSHH Principle

- Why? - Justification
- What? - Solution approach
- When? - Define the criticality, when and where does the need exist
- Who? & Where? - The overall team including management, outside contracting and customer interface
- How - technically and managerially - establish the organization, the technical strategy
- How much - the resources including new capital expenditures required for success
- All parts need to be in place. Omit even one may well lead to failure

Chapter 4 - Software Project Metrics

"If you can't measure it, you really don't understand it"

- Viable data
  - Lines of Code
  - Effort - people - time may be subdivided between project tasks
  - Cost - direct and possibly overhead costs
  - Documentation pages
  - Errors - problems found before release. Due to many causes. It may be important to relate error to cause.
  - Defects - after release

- Metrics
  - Errors per KLOC, Defects per KLOC, $ per KLOC, Documentation per KLOC

Function Oriented Metrics

Object - Function Point

- A means of providing project independent metrics
  
<table>
<thead>
<tr>
<th>Measure, Par</th>
<th>Par. Value</th>
<th>Weight factor Complexity</th>
<th>Weighted Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>User Inputs</td>
<td>10</td>
<td>3</td>
<td>30</td>
</tr>
<tr>
<td>Num. of Files</td>
<td>6</td>
<td>7</td>
<td>42</td>
</tr>
<tr>
<td>Ext. Interfaces</td>
<td>4</td>
<td>5</td>
<td>20</td>
</tr>
</tbody>
</table>

  Function Point Total 92

- Use - Errors per FP; $ per FP, etc.
Metric Evaluation

### Project Assembled Metrics

<table>
<thead>
<tr>
<th>No</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
</tr>
</thead>
<tbody>
<tr>
<td>Val</td>
<td>3.2</td>
<td>4.4</td>
<td>1.3</td>
<td>5.1</td>
<td>3.3</td>
<td>4.9</td>
<td>2.0</td>
<td>4.5</td>
<td>4.6</td>
<td>2.3</td>
<td>3.6</td>
<td>5.7</td>
<td>4.7</td>
</tr>
<tr>
<td>[Dif]</td>
<td>1.2</td>
<td>3.1</td>
<td>3.8</td>
<td>1.8</td>
<td>1.6</td>
<td>1.4</td>
<td>2.9</td>
<td>2.5</td>
<td>0.1</td>
<td>2.3</td>
<td>1.3</td>
<td>2.1</td>
<td>1.0</td>
</tr>
</tbody>
</table>

Steps:
1. Calculate Moving Range: |Dif|
2. Plot Mean of |Dif| - mR_bar
3. Multiply - mR_bar*3.26 (Upper Control Limit)
4. Question - are moving averages inside UCL - yes - stable

Metric Evaluation (cont’d)

Indicates Management Software Operations are “Stable”

Corrections Using Failure Analysis

Beyond statistical analysis where failure is critical
- Analyze the reason for each failure
- For start determine reason(s) for failure (Fig. 4.3)
- Bad specification
- Improper coding due to miss interpretation
- Change not coordinated properly
- Wrong customer input - inadequate inquiry
- Determine how best the failure can be corrected in future.
- Implement the necessary changes in procedure to alleviate any future failures
- Do so without looking for scapegoats, blame, etc.

You can learn more from failure than from success.
Management Metrics - Summary

- Goals - provide quantitative information
- about Software Development Group's viability
- to direct improvements to Group's capabilities
- to enable better individual project management
- to better serve the company's customers while improving profits