Software Project Metrics

Anne Banks Pidduck
Upcoming Topics

- Measurement and metrics, Chapter 21
- Project tracking and control, Chapter 25
  - Scope, schedule, cost, quality, process,
  - Earned Value Assessment
  - Risk management control
- Change management
  - Software Quality Assurance (SQA), Chapter 30
  - Software Configuration Management (SCM), Ch. 31
Optional Topics, if Time

- Testing, validation and verification 23
- Software development tools 24
- Reporting and Communicating (People) 29
- Reliability 20
What is a Metric?

“A quantifiable measurement of software product, process, or project that is directly observed, calculated, or predicted.”

Futrell, p. 729

The *value* of a metric is how useful it is for decision-making.

Boehm
Metrics

- Problem Statement
  “I’m fat and ugly.”

- Analysis
  I need to lose weight and improve appearance.
Metrics ..2

- Requirements/Specifications
  I need to lose 20 pounds by May 15.
  {weigh 120 pounds} {in 8 weeks}

- Focus on measurable results!

[Also fix hair, buy new clothes, and so on.]
Metrics ..3

- Design

  *How* will we meet our requirements?
  Eat fewer calories; Exercise more

Not specific enough!
Eat 1000 calories & walk 1 hour per day
Metrics ..4

- Implement and test our plans
- We can measure specifically whether we’re doing what we should be doing (process)
- We can measure whether we’re making progress (meeting requirements)
- Maintenance, Continuous Improvement
  - Track and keep it up over the long term
SMART

- Specific
- Measurable
- Achievable or Attainable
- Realistic
- Trackable
Meeting Goals

- Goal – to lose weight
- Steps to Achieve Goal - process
- Actions – eat less, exercise more
- Achievements – incremental weigh-ins
- Proof: Measures or Quantities – lost 20 lbs
Measurement Scales

- Observed, predicted, calculated
- Absolute (count, number of whatever)
- Nominal (classes, such as Yes or No)
- Ordinal (augments nominal with ordering, such as low, medium, high)
- Interval (50 LOC C# versus 55 LOC Java)
- Ratio (50 LOC C# versus 55 LOC C#)
# Types of Software Metrics

<table>
<thead>
<tr>
<th>Category</th>
<th>Observed</th>
<th>Predicted</th>
<th>Calculated</th>
</tr>
</thead>
<tbody>
<tr>
<td>Product</td>
<td>LOC, # of test cases</td>
<td>Quality Defects</td>
<td>Efficiency</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Reliability</td>
</tr>
<tr>
<td>Process</td>
<td>SEI level</td>
<td>SEI level</td>
<td>Closure Risk</td>
</tr>
<tr>
<td></td>
<td>Peer review</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Project</td>
<td>Cost to date</td>
<td>Duration</td>
<td>EVA</td>
</tr>
<tr>
<td></td>
<td>Test time</td>
<td>Final cost</td>
<td></td>
</tr>
<tr>
<td>People and</td>
<td># of staff</td>
<td>Loading Effort</td>
<td>Productivity</td>
</tr>
<tr>
<td>Resources</td>
<td>Skills</td>
<td></td>
<td>Availability</td>
</tr>
</tbody>
</table>
Importance of Metrics

• Analyze product errors
• Assess project status
• Basis for estimates
• Determine product complexity
• Establish baselines
Importance of Metrics ..2

- Validate best practices
- Predict quality, schedule, effort and cost
- Track project progress
- Understand when a particular state or quality has been achieved
Useful Metrics

- Should be collected *only* to be useful
- *Precisely* defined, measurable, *quantifiable*
- Terms like “always, complex, efficient, user-friendly” need further definition
- Raw data and audit trails add to the value
- Simple, cheap, robust, consistent, easily accessible by all stakeholders
Basic Metrics

- Actual effort expended against a WBS
- Change request data (corrective, adaptive, preventive, perfective)
  - Figure 21-19, Change Request Form
- Peer review data (errors, defects)
Tracking

- Cost, effort, schedule
- Planned vs. Actual
- How to handle when things go off plan?
Software Metrics

- CMM measures software process
- Pert/Gantt measures schedule
- Budget measures cost
- Quality?
Measurements

- Actual (To date) and Projected
  - Cost
  - Schedule
  - Effort
  - Product features
Measurements …2

- Alternatives
  - Earned value analysis
  - Defect rates
  - Productivity (ex: SLOC)
  - Complexity (ex: function points)
Software Project Control

Work Package Development

1. Work Breakdown Structures (WBS)
2. Work Package Specifications
3. Binary Tracking Technique
4. Earned Value Tracking Technique, comparing to planned values
Software Project Control - 2

Software Metrics

1. Product metrics
2. Process metrics

Necessary for planning and control
Software Project Control - 3

Software Configuration Management

1. Identify software components
2. Track component changes
3. Track documentation
4. Control software evolution
Software Project Control - 4

Software Quality Assurance

1. Use a SQA process
2. Use engineering standards

Peer Reviews

1. Walkthroughs
2. Inspections
Verification and Validation

1. Quality
2. Quantity

Software Engineering Assessment

1. Validate organizational capability
2. Identify software process improvements
Standards

- IEEE 1058, Standards for SPM Planning
- ACM
- CSA
- ISO
- SEI’s CMM
- National Bureau of Standards