CS599 Software Process Modeling
Week 10

Ray Madachy

November 2, 1999
Outline

• COCOMO Forum and Workshop debriefing
• Homework feedback and revised assignment
• Common structures supplement
• Overview of Past Applications
• Exhaustion model in Abdel-Hamid
• Global process feedback research
• Rayleigh staffing curves revisited
• Interactive simulation tools
  – Incremental Rayleigh and Dynamic COCOMO demos
• Earned value model revisited - management training
• Homework
CS599 Software Process Modeling
Student Term Projects

• Dynamics of architecture development process in MBASE inception and elaboration phases
• Application of RAD techniques to pre-IPO internet companies
• COTS glue-code development and integration dynamics
• Reuse and language-level effects in software development
• CMM-based process improvement strategies
Software Production Structure

- Combines task development and personnel chains.
- Production constrained by productivity and applied personnel resources.
Feedback Loops

• A feedback loop is a closed path connecting an action decision that affects a level, then information on the level being returned to the decision making point to act on.

• This example is the simplest form of feedback without additional delays, information distortions or interconnected feedback loops.

• Only available, *apparent* information is used for decision basis.

  feedback data may be late or erroneous
# Brief History

<table>
<thead>
<tr>
<th>Year</th>
<th>Event</th>
</tr>
</thead>
<tbody>
<tr>
<td>1961</td>
<td>Jay Forrester publishes <em>Industrial Dynamics</em></td>
</tr>
<tr>
<td>1984</td>
<td>Tarek Abdel-Hamid completes Ph.D. dissertation at MIT</td>
</tr>
<tr>
<td>late 1980's</td>
<td>NASA JPL and a few others begin research with system dynamics</td>
</tr>
<tr>
<td>1991</td>
<td>Abdel-Hamid and Stuart Madnick publish <em>Software Project Dynamics</em></td>
</tr>
<tr>
<td>1991-1999</td>
<td>Many industrial and academic implementations, including the effects of process improvement initiatives</td>
</tr>
</tbody>
</table>
Model Implementations

- **Industry/government**: AT&T, Bellcore, Draper Labs, Fedex, Hughes, Litton, Mitre, NASA, Siemens, others
- **Academic**: ASU, Imperial College, Stanford, MIT, Naval Postgraduate School, USC, others
- **Tool vendors/workshops**: Bartz Associates, Dynamica, Rubin Systems
- Many other companies are evaluating system dynamics for process improvement
- Several academic research projects in proposal stage or dissertations being written
Process Evaluation

- Investigating the dynamic effects of inspections [Madachy 94], [Tvedt 95]
- Incremental development [Tvedt 95]
- Unit testing phase [Collafello et al. 96]
- Requirements phase (several)
- Investigating software reuse from a macro-inventory perspective [Abdel-Hamid 93a]
- Software outsourcing [Collafello et al. 99]
- Process model tradeoffs
Process Evaluation (continued)

- Organizational CMM-based process improvement (Burke 96)
- Other process improvement investments
  - staffing policies
  - work environment investments
  - computer aided tool investments
  - staff training investments
  - metrics, reuse, risk management and others
- Global software process feedback, stability and product evolution [Lehman 98]
Flight Simulators

• Personnel training
  • graduate software project management (ASU)
  • vendor tools (Rubin et al.)
• Navigating new skies
  • process maturity initiatives
• Stimulate dialogues for shared mental models
• Virtual reality for court cases
Other Applications

• Integration with cost estimation models
  • improving on static assumptions [Madachy 95], [Rubin et al. 95]
  • calibrations between [Madachy 95]
  • deriving static parameters with dynamic experiments [Madachy 95]

• Knowledge-based assistance/expert systems
  • heuristic project risk analysis and input checking [Madachy 94]
  • input evaluation and change recommendation [Lin et al. 92]
  • QA expert simulator

• Examining heuristics
  • Brookes’s Law (several)
  • cost estimation correction processes [Abdel-Hamid 93]
  • others
Sample Insights

• Inspection policy tradeoff analysis - diminishing returns from inspections as a function of error generation rates [Madachy 94]

• QA policy tradeoff analysis - finding the optimal QA effort [Abdel-Hamid/Madnick 91]

• Rework staffing allocation [Tvedt 95]

• Organizational process improvement transition requires temporary productivity setbacks [Rubin, Johnson, Yourdon 95]

• Maximize your pro-SPI people (Burke 96)
Sample Insights (continued)

- Leverage of experienced staff (several)
- Internal workings of Brookes’s Law - training and communication losses
- Schedule compression not a static decision [Abdel-Hamid 90]
- Anchor-dragging in project control [Abdel-Hamid 93]
- Competing feedback loops in software reuse factory [Abdel-Hamid 93b]
- Many others
Abdel-Hamid Model Subsystems

- Human Resource Management
  - work force available
  - progress status
  - tasks completed
- Software Production
  - work force needed
  - schedule
- Controlling
  - efforts remaining
- Planning
  - planning
Abdel-Hamid Model Behavior

- Underestimation factor = .67
QA Policy Tradeoff Analysis

- Note non-standard QA definition
Exhaustion Model
Exhaustion Model Assumptions

- Workers increase their effective hours by decreasing slack time or working overtime.
- The maximum shortage that can be handled varies.
- Workers are less willing to work hard if deadline pressures persist for a long time.
- The overwork duration threshold increases or decreases as people become more or less exhausted.
- The exhaustion level also increases with overwork.
- The multiplier for exhaustion level is 1 when people work full 8 hour days, and goes over 1 with overtime. The exhaustion increases at a greater rate in overtime mode up to the maximum tolerable exhaustion.
- The exhaustion level slowly decreases when the threshold is reached or deadline pressures stop with an exhaustion depletion delay.
- During this time, workers don’t go into overwork mode again until the exhaustion level is fully depleted.
Exhaustion Model Relationships

exhaustion flow

multiplier to overwork duration threshold
Exhaustion Behavior
Abdel-Hamid Model Critique

Advantages

- The model includes a good deal of important dynamic effects. It does well to illustrate some methods of poor software management (this can also be a downfall if certain policies are emulated, or the model becomes prescriptive instead of descriptive).
- It uses the very realistic notion that management perceptions, rather than true conditions, dictate actions.
- Delays in action are also important realistic considerations that the model covers well.
- The inclusion of important personnel attributes like motivation, exhaustion and schedule pressure effects.
- The model is strong in terms of planning and control structures.

Downfalls

- The model contains too many elements for managers to understand and use, and requires initialization of many parameters and functions.
- The software production model is too simplistic for many purposes. Design and code are aggregated together.
- The model may be wrongly used or overly relied on to perpetuate poor management practices, such as not being able to status progress early on.
- The definition of QA is non-standard. The activities modeled as QA would be considered part of standard development activities in most installations, and performed by the same people who develop the software.
- The control function uses a misleading indicator of progress.
Risk Analysis

• A deterministic point estimate from a simulation run is only one of many actual possibilities

• Simulation models are ideal for exploring risk
  • test the impact of input parameters
  • test the impact of different policies

• Monte-Carlo analysis takes random samples from an input probability distribution
Monte-Carlo Example

- Results of varying inspection efficiency:

Effort Bin (Person-days)

<table>
<thead>
<tr>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
</tr>
<tr>
<td>8</td>
</tr>
<tr>
<td>7</td>
</tr>
<tr>
<td>6</td>
</tr>
<tr>
<td>5</td>
</tr>
<tr>
<td>4</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>1</td>
</tr>
</tbody>
</table>

Effort Bins: 3991, 4058, 4126, 4193, 4260, 4328, 4395
Global Feedback Model

• Review handouts

- new specification unit arrival rate
- new specification unit generation rate arising from field trials
- specification unit to be processed
- specification units believed implemented
- specification unit completion rate
- specification unit deferral and abandonment rate
- specification unit growth and rescheduling rate
- released specification unit falsification rate
- post release specification rework rate
- new specification unit generation rate arising from field trials
Rayleigh Manpower Distribution

• Rayleigh curve is a popular model of personnel loading

• Assumptions:
  – Only a small number of people are needed at the beginning of a project to carry out planning and specification. As the project progresses and more detailed work is required, the number of staff builds up to a peak. After implementation and unit testing is complete, the number of staff required starts to fall until the product is delivered.
  – The number of people working on a project is approximately proportional to the number of problems ready for solution at that time
Rayleigh Formula

- A Rayleigh curve describes the rate of change of manpower effort per the following first order differential equation:

\[
\frac{dC(t)}{dt} = p(t)[K - C(t)]
\]

where \(C(t)\) is the cumulative effort at time \(t\), \(K\) is the total effort, and \(p(t)\) is a learning function. The learning function is linear and can be represented by

\[
p(t) = 2at
\]

where \(a\) is a positive number.

- The manpower rate of change represents the number of people involved in development at any time (staffing profile).
- The \(a\) parameter is an important determinant of the peak personnel loading called the manpower buildup parameter.
Rayleigh Model

cumulative effort

effort rate

learning function

estimated total effort

manpower buildup parameter

Graph 1: p2 (Untitled)

a=2

a=.5

a=.25

a=.125

\[ \text{cumulative}_\text{effort}(t) = \text{cumulative}_\text{effort}(t - dt) + (\text{effort}_\text{rate}) \times dt \]

INIT cumulative_effort = 0

INFLOWS:

\[ \text{effort}_\text{rate} = \text{learning}_\text{function} \times (\text{estimated}_\text{total}_\text{effort} - \text{cumulative}_\text{effort}) \]

\[ \text{estimated}_\text{total}_\text{effort} = 15 \]

DOCUMENT: Estimated total effort for the project.

\[ \text{learning}_\text{function} = \text{manpower}_\text{buildup}_\text{parameter} \times \text{time} \]

DOCUMENT: The Norden learning function.

\[ \text{manpower}_\text{buildup}_\text{parameter} = 5 \]

DOCUMENT: The manpower buildup parameter determines the steepness of buildup and the peak personnel loading.
Interactive Rayleigh Model Demo

- Vary manpower buildup parameter
- Demonstrate S-curve
- Show effect of midstream added requirements
Hypothetical Simulation Interface
Earned Value

• Earned value is a method for measuring project performance.
• It compares the amount of work that was planned with what was actually accomplished to determine if cost and schedule performance is as planned.
• Cost performance index (CPI) is the ratio of budgeted costs to actual costs (BCWP/ACWP)
• Schedule performance index (SPI) is the ratio of work performed to work scheduled (BCWP/BCWS)
Guidelines for Using Earned Value

• If effort is more than 15% complete and your baseline estimates are being overrun, you cannot recover.

• The overrun at completion will be greater than current overrun (based on over 700 DoD contracts).
  – far term estimates are probably worse than short-term estimates

• Adjust far-term estimates and figure out how to not get worse.
Earned Value Model
Homework

• Revised process concurrence homework due next week (11/9)
  – clearly identify the test cases including the process concurrence relationships used and staffing input profiles
  – summarize results in table/graphs showing cost and schedule for systems engineering, software development, and total cost/schedule for each trial
    • cost must include downtime periods
  – also report the total “dead time” for each run when the development process is constrained yet people are on board (i.e. constraint flag is on)
  – experiment with at least one other process concurrence relationship besides the given
  – more thorough analysis discussion, and address the hypothesis
Homework (cont.)

• Prepare for high-level model design reviews
• Prepare research summaries:
  – Ho - Burke SPI model
  – Mehta, Fakharzadeh - Ford-Sterman iterative model
  – others
• Short quiz again next week?