MODELING SOFTWARE PROCESSES WITH SYSTEM DYNAMICS: CURRENT DEVELOPMENTS

Raymond J. Madachy
USC Center for Software Engineering
Litton Data Systems

1996 International System Dynamics Conference
July 22-25, 1996

Software Development and Systems Thinking

- Competitive advantage is increasingly dependent on software development in many industrial sectors.
- Software development, a dynamic and complex process, requires systems thinking in order to improve in current environment.
  - Software process: a set of activities, methods, practices and transformations used by people to develop software.
- Models can be used to quantitatively evaluate the software process
  - can experiment with changed processes before committing project resources
  - interactive training for software managers; "process flight simulation"
  - implement process re-engineering and benchmark process improvement

Outline

- Introduction and brief history
- Process improvement initiatives
- Software process applications and current work
- Research issues and future work
- References

Brief History

<table>
<thead>
<tr>
<th>Year</th>
<th>Event</th>
</tr>
</thead>
<tbody>
<tr>
<td>1984</td>
<td>Tarek Abdel-Hamid completes Ph.D. dissertation at MIT.</td>
</tr>
<tr>
<td>late 1980s</td>
<td>NASA JPL and a few others begin research with system dynamics.</td>
</tr>
<tr>
<td>1991</td>
<td>Tarek Abdel-Hamid and Stuart Madnick publish Software Project Dynamics.</td>
</tr>
<tr>
<td>1991-1996</td>
<td>Many industrial and academic implementations, including the effects of process improvement initiatives (see References).</td>
</tr>
</tbody>
</table>

Software Process Initiatives

- Software Engineering Institute (SEI)
  - Capability Maturity Model (CMM) for process improvement
- ISO certification
- SPICE
- ESPRIT, others
- Business process re-engineering
- Several software metrics initiatives

Comparison of Modeling Paradigms

- Software engineers already employ a host of models
  - predictive static cost models
    - these are being extended with dynamic modeling
  - discrete event approaches for low-level process descriptions
    - generally lack feedback
    - comparison study underway

333
**Discipline Comparison**

- Software engineers are particularly well-suited for the system dynamics modeling process
  - systems view and programming experience
  - comfortable with levels of abstraction
  - similar heuristics and incremental development process used
- Model validation involves problems unique to system dynamics simulation

**Software Process Level Instances**

- Work artifacts (requirements, tasks, lines of code, function points, documentation pages, others)
- Defect levels
- Personnel levels
- Effort expenditure
- Schedule date
- Others

**Basic Flow Processes and Infrastructures**

- Software product transformations
- Error co-flows
- Error detection and rework
- Personnel experience pools and effort expenditures
- Cost/quality tradeoffs enabled

**Model Validation**

- Controversial issue in the software community
- Multi-perspective validation with quantitative and qualitative criteria needs to be "sold" and accepted
- Often confusion between point prediction and "understanding"
- Aided by improvements in metrics collection

**Model Implementations**

- **Industry/government**: AT&T, Bellcore, Draper Labs, 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

**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]
- Process model tradeoffs
Process Evaluation (continued)

- 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 et al. 96]

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 85], [Rubin et al. 94]
  - calibrations between [Madachy 94]
  - deriving static parameters with dynamic experiments [Madachy 85]
- Knowledge-based assistance
  - heuristic project risk analysis and input checking [Madachy 94]
  - input evaluation and change recommendation [Lin et al. 92]
  - QA expert simulator
- Examining heuristics
  - Brookes' 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]

Sample Insights (continued)

- Leverage of experienced staff (several)
- Internal workings of Brookes' Law - training and communication losses [Abdel-Hamid 93]
- 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

Directions for Future Work

- Model structures
- Common models and component reusability
- Usability
- Process model selection
- Knowledge-based techniques
- Object orientation
- Related simulation research
- Industrial data analysis
References


- JOURNAL ARTICLES, CONFERENCE PROCEEDINGS AND OTHERS


- Abdel-Halim T, Lessons learned from modeling the dynamics of software development, Communications of the ACM, December 1989

- Abdel-Halim T, Investigating the schedule trade-off in software development, IEEE Software, January 1990

- Abdel-Halim T, Adapting, correcting, and perfecting software estimates: a maintenance metaphor, IEEE Computer, March 1990

- Abdel-Halim T, Thinking in cycles, American Programmer, May 1993

References (continued)


- Lehman M, Process Improvement - The way forward, Proceedings of CASE 95, Las Vegas, June 1995


- Madachy R, A software project dynamics model for process cost, schedule and risk assessment, Ph.D. dissertation, Department of Industrial and Systems Engineering, USC, December 1994

References (continued)


- Reis H, Johnson M, Youdron D, Software process flight simulation: dynamic modeling tools and metrics, Information Systems Management, Summer 1995

References (continued)

- JOURNAL ARTICLES, CONFERENCE PROCEEDINGS AND OTHERS (continued)

- Gunth B. Nguyen H, Vaid R, Death of a software manager: How to avoid career suicide through dynamic process modeling, American Programmer, May 1993


- WORLD WIDE WEB

- My system dynamics site, including updates on forthcoming book: http://www.rcl.ucr.edu/~madachy.html