A Model for Estimating Multitasking and Work Interruptions Overhead in Software Projects

Alexey Tregubov

Center for System and Software Engineering
University of Southern California
In environments where software developers work on multiple projects, effort and schedule can be underestimated because:

- Existing cost and schedule estimation models do not explicitly account for cross-project multitasking overhead.

- When work is planned and estimated by experts multitasking overhead is not taken into account.
Cross-project multitasking in software development

- When developers work on several projects at the same time (over a week or even a day), they switch between them spending some time on context switching.

- Not all context switching is bad, but we only focused on excessive switching between different projects, which is interrupting work and causing productivity decline.

- Context switching between projects cost/time consists of
  - **physical switching** – switch between repositories, DBs, servers, etc. takes time
  - **cognitive context switching** – getting into ‘flow mode’ takes time

- Multitasking overhead is not explicit in work log, so it needs to be evaluated. Work log analysis algorithm counts interruptions evaluates overhead in work logs.
Cross-project multitasking and work interruptions

Cross-project multitasking overhead is often observed in

- Matrix organizations - resources are shared between several projects for better resource utilization.

- Multiple releases of a product – if a product is released more than one time, resources are shared between maintenance of previous releases and a new version (typical for small mid-size teams).

- System of System (SoS) environments – in SoSs, if a constituent system is developed for several customers (e.g. different software distributions/releases for each customer), resources are shared between different contexts. Context here is a customer-specific requirements, success-critical stakeholders, and everything that makes each system installation unique.
Gerald M. Weinberg’s heuristic

Weinberg's heuristic

% of effort spent on multitasking vs. Number of projects at the same time
Multitasking in work environment

Work productivity

Work interruptions

Corporate culture

Office Environment

Personal process

Personality

Scheduled / process induced multitasking

Parallel projects (projects that share resources)

Scope of this research
Research questions and hypothesis

- RQ1. What is the quantitative effect of cross-project multitasking overhead on development effort and quality?

  - **H1.a** The number of cross-project interruptions is (not) linearly proportional to the number of projects.
  - **H1.b** The G. Weinberg’s heuristic is (not) applicable for cross-project multitasking overhead estimation in software development teams.
  - **H1.c** The number of cross-project interruptions is (not) linearly proportional to the number of reopened tasks (rework).
Research questions and hypothesis (cont’d)

- RQ2. How can COCOMO II model be improved to account for cross-project multitasking overhead?
  
  - **H2.a** The multitasking effort multiplier (MEM) can be automatically evaluated based on work log observations.

  \[
  MEM = \frac{Effort}{Effort - Overhead} = \frac{E}{E - Reimmersion \times Interruptions} = \frac{1}{1 - R \frac{I}{E}}
  \]

  - **H2.b** Using the multitasking effort multiplier in the locally calibrated COCOMO II model can improve prediction accuracies of the COCOMO II.
Research methodology

- A mixed methods approach (both qualitative and quantitative approach) has been selected for this research. The research is based on work log, schedule, and source code observations collected from two sources:
  - Student class projects (CSCI577 software engineering class)
  - Industry projects

- The simulation is used to develop alternative scenarios of work execution for observed/recorded projects’ work logs
  - Result of each simulation is a work log for an alternative scenario
  - Simulation results are analyzed by work log analysis algorithm to determine cross-project multitasking overhead

- A work log analysis algorithm determines multitasking overhead for each instance of a work log

- Local COCOMO II calibration provides a tool that takes into account cross-project multitasking overhead. The model is calibrated using
  - effort data from work logs
  - SLOC from repository (counted by UCC)
  - number of project involved
Research methodology (cont’d)

RQ1: impact of multitasking on effort and quality

Test hypothesis
H1a,b,c

Compute quality metric
Count cross-project interruptions for each data point
Compute MEM for each data point

RQ2: COCOMO II calibration

Test hypothesis
H2a,b

COCOMO II calibration
Evaluate model prediction accuracies

Compute MEM for each data point
Count cross-project interruptions for each data point

Collected data (work logs and schedules)

Industry projects

Students’ projects

Simulation

Extended data set from simulation

Source data: work logs of different project groups
Reimmersion time determines the cost of work interruption

- Reimmersion time range: 5 minutes – 2 hours

Modeling the reimmersion time
- Constant value (model parameter)
- Self-evaluated values from subjects
- Variable reimmersion time based on interruptions conditions
  - Task complexity
  - Length of interruption
  - Cause of interruption (self-inflicted vs. external)
COCOMO calibration and extension

- Step 0. Identify COCOMO II parameters for a specific organization.
- Step 1. Locally calibrate COCOMO II using effort and SLOCs.
  - In this research we only calibrate the multiplicative constant $A$.
- Step 2. Introduce a new effort multiplier – MEM for multitasking in the model:

$$Effort = A' \times \left[ \text{Size} \right]^{B + \sum_{j=1}^{5} SF_j} \times \prod_{j=1}^{17} EM_j \omega \times MEM(I_{avg}, R)$$

  - Calibrate $A'$ using available data and evaluate prediction accuracy
  - $MEM$ is calculated from observations of multitasking overhead using metrics such as number of interruptions per week and number of parallel projects per week

**Result:** locally calibrated COCOMO II extension for a specific organization or team, which accounts for multitasking overhead.
Data collection

Industry projects:
- Two groups of projects, one year each
- Work logs in JIRA – hours reported daily for each task in each project
- Schedules were updated every 2-3 days
- SLOC added and changed for each project

577 students projects
- 10 teams/projects, 5-11 weeks development phase
- Work logs in JIRA – hours reported weekly for each task in each project
- Schedules were updated every 2 weeks
- SLOC added and changed for each project
COCOMO II calibration (industry)
COCOMO II calibration (cont’d)

<table>
<thead>
<tr>
<th></th>
<th>Local calibration of COCOMO II</th>
<th>Local calibration of COCOMO II with MEM</th>
</tr>
</thead>
<tbody>
<tr>
<td>$R^2$</td>
<td>0.33</td>
<td>0.44</td>
</tr>
<tr>
<td>$p$</td>
<td>0.02</td>
<td>0.03</td>
</tr>
<tr>
<td>$n$</td>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td>$A$ (multiplicative constant)</td>
<td>1.57</td>
<td>1.34</td>
</tr>
</tbody>
</table>
Calibrated COCOMO II prediction accuracies

<table>
<thead>
<tr>
<th></th>
<th>COCOMO II</th>
<th>Local calibration of COCOMO II</th>
<th>Local calibration of COCOMO II with MEM</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>PRED(.20)</strong></td>
<td>32%</td>
<td>56%</td>
<td>63%</td>
</tr>
<tr>
<td><strong>PRED(.25)</strong></td>
<td>37%</td>
<td>58%</td>
<td>75%</td>
</tr>
<tr>
<td><strong>PRED(.30)</strong></td>
<td>38%</td>
<td>61%</td>
<td>81%</td>
</tr>
</tbody>
</table>
Cross-project work interruptions (CSCI577 projects)

Average number of interruptions per week per developer

- **$R^2$**: 0.6110
- **$S$**: 2.418
- **$p$**: 0.0003
- **n**: 29
Cross-project work interruptions (industry)

Average number of interruptions per week per developer

- **R²**: 0.4563
- **S**: 2.198
- **p**: <0.005
- **n**: 154
Cross-project multitasking effort multiplier (CSCI577 projects)

Average MEM per week per developer

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>R²</td>
<td>0.11</td>
</tr>
<tr>
<td>S</td>
<td>0.2</td>
</tr>
<tr>
<td>p</td>
<td>0.008</td>
</tr>
<tr>
<td>n</td>
<td>29</td>
</tr>
</tbody>
</table>

Average number of projects per week

MEM

0.00  0.50  1.00  1.50  2.00  2.50

0.00  1.00  2.00  3.00  4.00  5.00  6.00
Cross-project multitasking effort multiplier (industry)

Average MEM per week per developer

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>R²</td>
<td>0.45</td>
</tr>
<tr>
<td>S</td>
<td>0.01</td>
</tr>
<tr>
<td>p</td>
<td>&lt;0.005</td>
</tr>
<tr>
<td>n</td>
<td>154</td>
</tr>
</tbody>
</table>
Comparison with Weinberg’s heuristic (CSCI577 projects)

Comparison of effort evaluation and estimations

- Average effort on spent on interruptions
- Weinberg heuristic

% of full time equivalent (FTE)

Number of projects per week

1
2
3
Comparison with Weinberg’s heuristic (industry)
Impact on quality (CSCI577 projects)

Grade deduction

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>R²</td>
<td>0.44</td>
</tr>
<tr>
<td>S</td>
<td>2.25</td>
</tr>
<tr>
<td>p</td>
<td>0.051</td>
</tr>
<tr>
<td>n</td>
<td>9</td>
</tr>
</tbody>
</table>

Average number of interruptions per week per developer
Impact on quality (industry)

Average number of reopened JIRA tickets

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>R²</td>
<td>0.71</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S</td>
<td>1.01</td>
<td></td>
<td></td>
</tr>
<tr>
<td>p</td>
<td>0.004</td>
<td></td>
<td></td>
</tr>
<tr>
<td>n</td>
<td>9</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Research contributions

- Quantified effect of cross-project multitasking on overall project effort and quality.
- A model for cross-project multitasking overhead evaluation based on work logs.
  - The model can also be used for evaluation of teams’ productivity in multitasked environment (MEM evaluation).
- A framework for COCOMO family models calibration that accounts for multitasking overhead.
- Examined relationships between cross-project multitasking overhead and number of parallel projects.
Future work

- The approach for counting work interruptions and evaluating their impact on effort can be applied to other types of multitasking. To do so, we need to develop methods for measuring the number of work interruptions caused by certain type of multitasking.

- The further accuracy evaluation of models with MEM (any COCOMO family model with MEM) should be done on a bigger sample of projects.

- The work log analysis tools, we used in the research, can be integrated with project tracking systems such as Atlassian JIRA to provide real-time information about work interruptions and their impact on productivity (e.g., calculate MEM).

- Further research should be done on how to schedule and organize work to reduce negative effects of work interruption on productivity.
Completed work

- Tools:
  - A hybrid agent based & discrete-event simulation model was developed and independently implemented in two software tools:
    - DES framework (old name – KSS simulator)
    - DATASEM – tool available on line: http://datasem.eng.auburn.edu/

- Published conference papers:
  - ICSSP’17: Impact of context switching and work interruptions on software development efficiency
  - CSER’17: Evaluation of cross-project multitasking in software projects.
  - CSER’15: Simulation of Kanban-based scheduling for systems of systems: initial results.
  - INCOSE Symposium 2016: What does it mean to be Lean in SoSE environment?
Backup slides
Conclusions: industry vs. 577 projects

**Industry**

Interruptions between projects per week

- average
- Multiple
- $R^2 = 0.7779$
- Adjusted
- $R^2 = 0.7576$
- $p < 0.01$

**577 class projects**

Number of interruptions per week

% of effort spent on interruptions
Work log analysis algorithm overview

- Computes how often each task was interrupted over the week
- Computes reimmersion time of each interruption

Work log analysis algorithm:

1. Identify cross-project interruptions
2. Determine reimmersion time for each cross-project work interruption
3. Sum up cost/effort of multitasking overhead of all interruptions
4. Multitasking overhead for a team working on a group of projects

Diagram:

- Work log observations of multitasked teams
- ‘Perfect world’ - no excessive context switching.
- ‘Imperfect world’ interpretation
- Real world - excessive context switching, multiple interruptions
Reimmersion time evaluation

<table>
<thead>
<tr>
<th>Average reimmersion time</th>
<th>Study</th>
</tr>
</thead>
<tbody>
<tr>
<td>~1 hour</td>
<td>Estimate by DeMarco and Lister [13]</td>
</tr>
<tr>
<td>15 minutes – 2 hours</td>
<td>DeMarco and Lister [13]</td>
</tr>
<tr>
<td>23 minutes</td>
<td>Gonzalez and Harris [40]</td>
</tr>
<tr>
<td>15-25 minutes</td>
<td>Van Solingen et al [47]</td>
</tr>
<tr>
<td>20-30 minutes</td>
<td>Parnin and Rugaber [27]</td>
</tr>
<tr>
<td>35-45 minutes</td>
<td>Self-evaluate in CSCI577 class projects (via surveys)</td>
</tr>
</tbody>
</table>
Reimmersion time: comparison with MS research

Reimmersion time distribution

- Self-evaluated by CSCI577 students
- Parnin and Rugaber's study

K-S test

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$D_{\text{stat}}$</td>
<td>0.223</td>
</tr>
<tr>
<td>$D_{\text{crit}}$</td>
<td>0.229751</td>
</tr>
</tbody>
</table>

- CSCI577 sample size: 65
- Parnin and Rugaber's sample size: 444
References