Evaluation of multitasking and work interruptions in software projects

Alexey Tregubov
University of Southern California
Motivation

- 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.

- Cross-project multitasking overhead is often not accounted when projects are estimated. This may cause underestimation in project planning.
Problem statement

In environments where software developers work on multiple projects, effort and schedule can be underestimated because of the following:

- 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.
Gerald M. Weinberg’s heuristic

Multitasking overhead

<table>
<thead>
<tr>
<th>proj. 1</th>
<th>proj. 2</th>
<th>proj. 3</th>
<th>proj. 4</th>
<th>proj. 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>% total effort</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Q1. Is the cross-project multitasking overhead (effort, number of interruptions) linearly correlated with the number of projects?

Q2. What is the quantitative effect of cross-project multitasking on development effort?
Multitasking in work environment

Work productivity

Work interruptions

Corporate culture

Office Environment

Personality

Personal process

Scheduled / process induced multitasking

Parallel projects (projects that share resources)

Scope of this research
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.
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 (from “MSS-Holding” Co.Ltd.)
- A work log analysis algorithm determines multitasking overhead for each instance of a work log
Work log analysis algorithm overview

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

- Reimmersion time determines the cost of work interruption

- 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)

- Reimmersion time range: 5 minutes – 2 hours
Data collection

- Industry projects:
  - 6 projects, 81 software developers
  - 1 year of work logs
  - Work logs in JIRA – hours reported daily for each task in each project, schedules were updated daily.
Results: multitasking overhead
Results

Average number of interruptions per week for each project

<table>
<thead>
<tr>
<th>Project id</th>
<th>Number of interruptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>120.00</td>
</tr>
<tr>
<td>2</td>
<td>100.00</td>
</tr>
<tr>
<td>3</td>
<td>90.00</td>
</tr>
<tr>
<td>4</td>
<td>100.00</td>
</tr>
<tr>
<td>5</td>
<td>80.00</td>
</tr>
<tr>
<td>6</td>
<td>50.00</td>
</tr>
</tbody>
</table>
Average number of interruptions

\[ y = 2.9149x - 1.5141 \]
\[ R^2 = 0.5676 \]
\[ p = 0.0058 \]
Results: observations vs. heuristic

% of effort total spent on interruptions

Observations
G. Weinberg's heuristic

Number of projects per week

0 1 2 3 4 5 6 7
Conclusion

- Analysed work logs of 81 software developers working on 6 projects for one year.
- Among all 6 projects at least 14% effort was spent on context switching between tasks from different projects.
- Developers who were involved in more projects tend to have more cross-project work interruptions.
- The linear correlation between the number of projects each resource is working on in one week and the number of interruptions exists but it is relatively weak.
Future work

- The impact of multitasking on effort can be integrated into parametric cost and schedule estimation models such as the Constructive Cost Estimation Model (COCOMO ®) for better effort estimation.

- 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.
Questions

Q&A
References


References


17. http://www.businessdictionary.com/definition/matrix-organization.html

