A Lightweight Incremental Effort Estimation Model For Use Case Driven Projects

Kan Qi, Dr. Barry Boehm
University of Southern California
{kqi,boehm}@usc.edu
Outline

• Background of use case driven approach
• Motivations
• Software sizing models
• Transaction identification and classification
• Size metrics
• Empirical Study
• Conclusions
Use case driven approach of software engineering

- functional requirements are captured by use cases that define the interactions between actors and a system.
- use cases can be written in textual format as use case narratives or visually represented by use case diagrams.
- system reactions can further be modeled in more detail by robustness diagrams, sequence diagrams, and class diagrams.
Background

- Existing methods for software sizing and effort estimation
  - COCOMO II, uses SLOC as the size metric and 22 factors to model the multiplicative and exponential effects from the project, personnel, product, and process aspects on project effort. The exact effects are calibrated on 161 projects.
  - Function Points, used as a measure of the units of functionality, which specifically are measured by data and transaction functions.
  - Use Case Points, estimate software size by the number of use cases, weighted by the number of transactions.
  - Story Points, models the relative effort for tasks based on a typical task. Popular with agile development.

- Estimate effort at the early stage of a project
  - Schedule and cost estimation
  - Risk management
Motivations

- **Difficulties in applying other estimation methods to use case driven projects**
  - Hard to accurately estimate software size in terms of source lines of code, unless the estimations are done by experts or for precedential systems.
  - Function Points requires detailed information to determine the elementary processes and data elements of different types. It requires expertise to achieve a reproducible calculation.
  - Use Case Points involves manual counting process, and may not be accurate as a size estimator for not considering the internal complexity of the transactions.
  - Story Points is only able to estimate effort of the tasks at local scales, since different groups of users may use a different reference task.
Motivations

- **Our goals**
  - To improve the efficiency of the counting process.
  - To deliver an effort estimation model that is compatible with the lifecycle of use case driven projects.
  - To improve the accuracy based on information available at different phases of the software process.
Challenges and Approaches

- **Lifecycle**: Phase-based effort estimation models to provide multiple estimations at different phases of the process.
  - Two phases of estimation to keep the balance between utility and accuracy.
  - Information availability assumed based on the typical deliverables of the two phases.

- **Agility**: Size metrics are directly countable from the artifacts of the process to avoid investing too much effort in collecting information for effort estimation.
  - Automated counting procedures are developed.

- **Accuracy**: Incrementally integrate information available at the different phases of the process to more accurately estimate software size.
Software Sizing Models

- **Use Cases**
  - Use cases are used to capture the interactions between actors and a system. They are usually represented with use case narratives, activity diagrams, robustness diagrams, or sequence diagrams.

- **Transactions**
  - A sequence of interactions between system components, which realizes a basic unit of system functionality. Use cases are modeled by a set of transactions.

\[
\text{Project Size} = \sum_{c \in C} w_c
\]

- **Weighted Transactions**
  - As an architecture introduced, the connectors and components of a system are decided, so as the internal structure of the transactions. Transactions are weighted by its internal structure to represent its contribution to the overall effort.

\[
\text{Project Size} = \sum_{c \in C} \sum_{t \in c} w_t
\]
Transaction Identification and Classification

- **Transaction identification**
  - For use case narratives, structured scenarios are extracted and converted to activity diagrams. The number of transactions are counted as number of scenarios, or the number of paths in an activity diagram.
  - For robustness diagrams, each independent path of the directed graph is defined as a transaction.
  - For sequence diagrams, the meaningful sequence of messages is defined as a transaction.
Transaction Identification Examples

Transactions are as flows of events of activity diagrams.

Transactions are as independent paths of robustness diagrams.

Transactions are as sequences of messages of sequence diagrams.
Transaction Identification and Classification

**Transaction classification**

- The elements on an independent path represent the internal structure of a transaction.
  - number of system components a transaction is implemented upon.
  - the types of the elements, for example, boundary, control, and entity.
- Transactions are classified into different levels of complexity by the number of user interface elements and domain elements.
- User interface elements (UIE) are defined as identifiable UI components.
- Domain elements (DE) are defined as the elements on the independent paths.

<table>
<thead>
<tr>
<th>UIE : DE</th>
<th>1-3</th>
<th>4-7</th>
<th>&gt;=8</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-1</td>
<td>Low</td>
<td>Low</td>
<td>Medium</td>
</tr>
<tr>
<td>2-5</td>
<td>Low</td>
<td>Medium</td>
<td>High</td>
</tr>
<tr>
<td>&gt;5</td>
<td>Medium</td>
<td>High</td>
<td>High</td>
</tr>
</tbody>
</table>

Example of transaction classification scheme
Size metric - I

- Early Use Case Points (EUCP)
  - Use the number of scenarios to weight the use cases, and sum all the weighted use cases to calculate the Unadjusted Early Use Case Weight (UEUCW).
    - Transactions are automatically identified from use case narratives through the converted activity diagrams.
  - Calculate Unadjusted Actor Weight (UAW) based on the actors identified from use case narratives.
  - Evaluate the project for the 13 Technical Complexity Factors (TCF) and 8 Environmental Factors (EF) from original UCP definition.
  - Calculate EUCP based on:

\[
EUCP = (UEUCW + UAW) \times TCF \times EF
\]
Extended Use Case Points (EXUCP)

- Transactions are identified as the independent paths of robustness diagrams or sequences of messages from sequence diagrams.
- Use the number of Domain Elements (DE) and UI elements (UIE) that each identified transaction interacts with to weight the transaction as the Unadjusted Transaction Weight (UTW).
- Sum the total UTW to calculate Unadjusted Extended Use Case Weight (UEXUCW) for the use cases.
- Reuse the evaluations for UAW, TCF, and EF from size metric – I.
- Calculate EXUCP by the equation below:

\[
EXUCP = (UEXUCW + UAW) \times TCF \times EF
\]
Empirical Study

Data collection

- 4 projects from master level software engineering courses (csci577 and csci590) from 2014-2017.
- They are mobile applications or web applications, written in 3rd Gen programming languages: PHP, JavaScript, Nodejs, Java, Object-C, etc.
- The projects are lasted for about 4 – 12 months, and done with teams of 5-24 people. Delivered software applications ranging from 3-20 KSLOC.
- 114 use cases were collected in total, including use case narratives, robustness diagrams, and sequence diagrams.
- Effort data were collected through weekly effort reports.
Model Calibration

- Apply the counting processes and algorithms to identify the information needed for Early Use case Points (EUCP) and Extended Use Case Points (EXUCP).
- Calculate EUCP and EXUCP for the 4 projects.
- Normalize Effort.
  - Remove influences from the un-modeled environmental factors.
  - The un-modeled factors are the factors that are modeled by COCOMO II, but not modeled by Use Case Points. Detail is provided in the paper.
- Correlation coefficients are calculated to determine if linear relationships exist between the size metrics and normalized effort within the data set.
- Apply linear regression to calibrate the linear models to understand the exact effects the metrics have on project effort.
- Evaluate the goodness of fit by $R^2$, MMRE, and PRED(.25).
- Hypothesis tests are applied to understand the significance of the estimates.
Empirical Results

- Counting results for the four sample projects

<table>
<thead>
<tr>
<th>Project</th>
<th>UEUCW</th>
<th>UEXUCW</th>
<th>UAW</th>
<th>TCF</th>
<th>EF</th>
<th>EUCP</th>
<th>EXUCP</th>
</tr>
</thead>
<tbody>
<tr>
<td>BDR</td>
<td>125</td>
<td>62</td>
<td>14</td>
<td>1.14</td>
<td>1.21</td>
<td>191.74</td>
<td>104.83</td>
</tr>
<tr>
<td>PCS</td>
<td>200</td>
<td>132</td>
<td>8</td>
<td>1.06</td>
<td>1.03</td>
<td>227.09</td>
<td>152.85</td>
</tr>
<tr>
<td>LBA</td>
<td>370</td>
<td>270</td>
<td>12</td>
<td>1.12</td>
<td>1.32</td>
<td>564.75</td>
<td>416.91</td>
</tr>
<tr>
<td>TIKI</td>
<td>95</td>
<td>34</td>
<td>3</td>
<td>1.18</td>
<td>1.24</td>
<td>143.39</td>
<td>54.14</td>
</tr>
</tbody>
</table>

- Actual and normalized effort

<table>
<thead>
<tr>
<th>Project</th>
<th>PH\text{actual}</th>
<th>PH\text{norm}</th>
<th>PH_{model\text{I}}</th>
<th>PH_{model\text{II}}</th>
</tr>
</thead>
<tbody>
<tr>
<td>BDR</td>
<td>1392</td>
<td>1270.21</td>
<td>1213.92</td>
<td>1197.26</td>
</tr>
<tr>
<td>PCS</td>
<td>2016</td>
<td>1711.80</td>
<td>1392.65</td>
<td>1490.05</td>
</tr>
<tr>
<td>LBA</td>
<td>3680</td>
<td>3029.65</td>
<td>3099.51</td>
<td>3100.19</td>
</tr>
<tr>
<td>TIKI</td>
<td>737</td>
<td>663.96</td>
<td>969.54</td>
<td>888.13</td>
</tr>
</tbody>
</table>

- Calibrated coefficients for the linear effort estimation models

<table>
<thead>
<tr>
<th>Model</th>
<th>\beta_0</th>
<th>\beta_1</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>244.68</td>
<td>5.06</td>
</tr>
<tr>
<td>II</td>
<td>558.01</td>
<td>6.10</td>
</tr>
</tbody>
</table>
Model Evaluation

- $R^2$ for the two calibrated models

<table>
<thead>
<tr>
<th>Model</th>
<th>$R^2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>0.933</td>
</tr>
<tr>
<td>II</td>
<td>0.964</td>
</tr>
</tbody>
</table>

- MMRE and PRED (.25) for the linear models

<table>
<thead>
<tr>
<th>Model</th>
<th>MMRE</th>
<th>PRED (.25)</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>17.9%</td>
<td>75%</td>
</tr>
<tr>
<td>II</td>
<td>13.7%</td>
<td>75%</td>
</tr>
</tbody>
</table>

- Hypothesis tests to evaluate the significance of the estimates

<table>
<thead>
<tr>
<th>Model</th>
<th>$\beta_0$</th>
<th></th>
<th>$\beta_1$</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SE</td>
<td>t-v</td>
<td>p-v</td>
<td>SE</td>
</tr>
<tr>
<td>I</td>
<td>313.92</td>
<td>0.779</td>
<td>0.517</td>
<td>0.960</td>
</tr>
<tr>
<td>II</td>
<td>192.25</td>
<td>2.903</td>
<td>0.101</td>
<td>0.837</td>
</tr>
</tbody>
</table>
Conclusions

• The preliminary calibration results have shown linear relationships exist between the proposed size metrics and project effort.
• Linear regression shows the linear models fit the data set well in terms of MMRE, PRED(.25), and $R^2$. MMRE and PRED(.25) are not accuracy indices since they are calculated based on the training dataset.
• Extended Use Case Points model is superior to Early Use Case Points model for its higher value for $R^2$ and the lower value for MMRE.
• Further evaluation of the estimation accuracy is needed, which requires more data points to be collected.
Future Directions

- More data points need to be collected to draw conclusions about the estimation accuracy of the models, and also to evaluate if the superiority of EXUCP model over EUCP model is significant.
- Supporting software tools need to be developed to streamline the process of training and testing the models.
- Explore applicability of the evaluation metrics on more types of UML diagrams.
- Extend the proposed metrics to other software management decisions, for example, resource allocation, schedule estimation, design quality assessment, etc.
Survey Link

- http://52.15.204.194:8081/surveyproject
Thanks! & Questions?
References-I


