ASSESSING EFFORT ESTIMATION MODELS FOR CORRECTIVE MAINTENANCE THROUGH EMPIRICAL STUDIES

Andrea De Lucia, Eugenio Pompella, Silvio Stefanucci
Information and Software Technology 47 (2005)
Contents

- Introduction
- Overall Approach
- Empirical studies
  - Assessing the maintenance process and building effort estimation models
  - Applicability of the effort prediction models
  - Assessing predictive performances of the effort estimation models on a new project
- Related Works
- Conclusion and Discussion
Planning software maintenance work

- **Key factor** for a successful maintenance project
  - Guarantee the control of the maintenance process
  - Reduce the risks and the inefficiencies related to the maintenance work

To this aim, **Effort estimation**

- **Valuable asset** to maintenance managers
  - In planning maintenance activities and performing cost/benefits analysis
Unfortunately,

- **Effort estimation**
  - One of the most relevant *problems* of the software maintenance process

- **Predicting software maintenance effort**
  - *Complicated* by the many typical aspects of software and software systems that affect maintenance activities

- **Any estimation or control technique**
  - Must reflect *a large number of complex and dynamic factors*
Introduction (3/3)

- Basic premise
  - Can develop accurate quantitative models
    - Predict maintenance effort using historical project data

- Research goal
  - Empirical assessment
  - Improvement of the effort estimation model for corrective maintenance
Overall Approach

- **Multiple linear regression analysis**
  - To construct effort estimation models
    - Validated against real data collected from five corrective maintenance projects

- Replicated assessment of the effort prediction models
  - Built in the previous phase on a new corrective maintenance project
    - within the same enterprise and the same application domain
Assessing the maintenance process and building effort estimation models (1/7)

- **Experimental setting**
  - **Data set (144 observations):** EDS Italia Software
    - Composed of five corrective software maintenance projects
      - Telecommunication, banking, insurance, and government organizations (written in COBOL)
  - **Collected metrics**

<table>
<thead>
<tr>
<th>Metric</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NA</td>
<td>No. of tasks requiring software modification</td>
</tr>
<tr>
<td>NB</td>
<td>No. of tasks requiring fixing of data misalignment</td>
</tr>
<tr>
<td>NC</td>
<td>No. of other tasks</td>
</tr>
<tr>
<td>N</td>
<td>No. of tasks (N=NA+NB+NC)</td>
</tr>
<tr>
<td>SIZE</td>
<td>Size of the system to be maintained [kLOC]</td>
</tr>
<tr>
<td>EFFORT</td>
<td>Actual Effort [man-hours]</td>
</tr>
</tbody>
</table>
## Experimental setting (cont’d)

### Descriptive statistics of the data set

<table>
<thead>
<tr>
<th>Project</th>
<th>Size (kLOC)</th>
<th>Min</th>
<th>Max</th>
<th>Mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1</td>
<td>293</td>
<td>2</td>
<td>120</td>
<td>53.07</td>
</tr>
<tr>
<td>P2</td>
<td>2738</td>
<td>8</td>
<td>93</td>
<td>42.88</td>
</tr>
<tr>
<td>P3</td>
<td>5045</td>
<td>16</td>
<td>36</td>
<td>18.71</td>
</tr>
<tr>
<td>P4</td>
<td>2344</td>
<td>2</td>
<td>34</td>
<td>18.5</td>
</tr>
<tr>
<td>P5</td>
<td>1733</td>
<td>0</td>
<td>43</td>
<td>7.06</td>
</tr>
</tbody>
</table>

**Note:**
- Containing missing values or
- Not distinguishing types of different maintenance tasks
Assessing the maintenance process and building effort estimation models (3/7)

- Building effort estimation models
  - Focus on ordinary multivariate least squares regression
    - Including the number of tasks (total and different types) and the size of the system
  - Metrics correlation matrix

<table>
<thead>
<tr>
<th></th>
<th>N</th>
<th>NA</th>
<th>NB</th>
<th>NC</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>N</td>
<td>1.0000</td>
<td>0.2078</td>
<td>0.9466</td>
<td>0.8340</td>
<td>0.1010</td>
</tr>
<tr>
<td>NA</td>
<td>0.2078</td>
<td>1.0000</td>
<td>0.4002</td>
<td>0.6458</td>
<td>0.0759</td>
</tr>
<tr>
<td>NB</td>
<td>0.9466</td>
<td>0.4002</td>
<td>1.0000</td>
<td>0.0864</td>
<td>0.3079</td>
</tr>
<tr>
<td>NC</td>
<td>0.8340</td>
<td>0.6458</td>
<td>0.0864</td>
<td>1.0000</td>
<td>0.0012</td>
</tr>
<tr>
<td>Size</td>
<td>0.1010</td>
<td>0.1217</td>
<td>0.2857</td>
<td>0.2652</td>
<td>1.0000</td>
</tr>
</tbody>
</table>

Tasks of type A are less recurrent than other two task types.
Assessing the maintenance process and building effort estimation models (4/7)

- Building effort estimation models (cont’d)
  - Model A: Effort = $b_1N + b_2\text{Size}$
    - Including the total number of maintenance tasks $N$ and the size of the system being maintained
  - Model B: Effort = $b_1\text{NA} + b_2\text{NBC} + b_3\text{Size}$
    - NBC is the sum of NB and NC
  - Model C: Effort = $b_1\text{NA} + b_2\text{NB} + b_3\text{NC} + b_4\text{Size}$
Assessing the maintenance process and building effort estimation models (5/7)

- Evaluating model performances
  - Model parameters
  - Model predictive performances

<table>
<thead>
<tr>
<th>Model</th>
<th>Var.</th>
<th>$b_i$ (coeff.)</th>
<th>$p$-value</th>
<th>$R^2$</th>
<th>Adj $R^2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>N</td>
<td>1.342904</td>
<td>$&lt; 10 \times 10^{-7}$</td>
<td>0.8257</td>
<td>0.8245</td>
</tr>
<tr>
<td></td>
<td>Size</td>
<td>0.169086</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>NA</td>
<td>9.053286</td>
<td>$&lt; 10 \times 10^{-7}$</td>
<td>0.8891</td>
<td>0.8876</td>
</tr>
<tr>
<td></td>
<td>NBC</td>
<td>0.138275</td>
<td>$&lt; 10 \times 10^{-7}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Size</td>
<td>0.164826</td>
<td>$&lt; 10 \times 10^{-7}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>NA</td>
<td>7.86988</td>
<td>$&lt; 10 \times 10^{-7}$</td>
<td>0.8963</td>
<td>0.8941</td>
</tr>
<tr>
<td></td>
<td>NB</td>
<td>0.514121</td>
<td>$&lt; 10 \times 10^{-7}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>NC</td>
<td>2.81486</td>
<td>0.000001</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Size</td>
<td>0.130507</td>
<td>$&lt; 10 \times 10^{-7}$</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Model A</th>
<th>Model B</th>
<th>Model C</th>
</tr>
</thead>
<tbody>
<tr>
<td>Press</td>
<td>21,630,440</td>
<td>14,163,410</td>
<td>13,591,780</td>
</tr>
<tr>
<td>SPR</td>
<td>46,602.18</td>
<td>37,397</td>
<td>35,596.61</td>
</tr>
<tr>
<td>MMRE</td>
<td>42.53%</td>
<td>36.40%</td>
<td>32.25%</td>
</tr>
<tr>
<td>MdMRE</td>
<td>37.57%</td>
<td>29.16%</td>
<td>25.35%</td>
</tr>
<tr>
<td>PRED$_{25}$</td>
<td>31.25%</td>
<td>40.36%</td>
<td>49.31%</td>
</tr>
<tr>
<td>PRED$_{50}$</td>
<td>66.75%</td>
<td>74.56%</td>
<td>82.64%</td>
</tr>
</tbody>
</table>

2007-04-11
Software Engineering Lab, KAIST
Assessing the maintenance process and building effort estimation models (6/7)

- Assessing predictive performances through leave-more-out cross-validation
  - Leave more out cross validation with time-based partition

<table>
<thead>
<tr>
<th>Learning %–test %</th>
<th>L_{40}–T_{60} (%)</th>
<th>L_{50}–T_{50} (%)</th>
<th>L_{60}–T_{40} (%)</th>
<th>L_{70}–T_{30} (%)</th>
<th>L_{80}–T_{20} (%)</th>
<th>L_{90}–T_{10} (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MMRE</td>
<td>56.93</td>
<td>50.40</td>
<td>45.91</td>
<td>46.03</td>
<td>43.78</td>
<td>39.29</td>
</tr>
<tr>
<td>MdMRE</td>
<td>49.73</td>
<td>34.73</td>
<td>31.14</td>
<td>35.12</td>
<td>33.97</td>
<td>31.83</td>
</tr>
<tr>
<td>PRED_{25}</td>
<td>28.82</td>
<td>35.14</td>
<td>41.43</td>
<td>40.48</td>
<td>40.74</td>
<td>39.46</td>
</tr>
<tr>
<td>PRED_{50}</td>
<td>47.06</td>
<td>60.81</td>
<td>67.14</td>
<td>66.67</td>
<td>66.67</td>
<td>71.23</td>
</tr>
</tbody>
</table>

- Leave more out cross validation with project partitions

<table>
<thead>
<tr>
<th>Maintenance project</th>
<th>P1 (%)</th>
<th>P2 (%)</th>
<th>P3 (%)</th>
<th>P4 (%)</th>
<th>P5 (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MMRE</td>
<td>64.65</td>
<td>25.60</td>
<td>79.20</td>
<td>31.99</td>
<td>35.17</td>
</tr>
<tr>
<td>MdMRE</td>
<td>50.63</td>
<td>21.48</td>
<td>66.96</td>
<td>33.72</td>
<td>26.33</td>
</tr>
<tr>
<td>PRED_{25}</td>
<td>33.33</td>
<td>60.61</td>
<td>10.00</td>
<td>33.33</td>
<td>47.22</td>
</tr>
<tr>
<td>PRED_{50}</td>
<td>46.67</td>
<td>84.85</td>
<td>30.00</td>
<td>86.67</td>
<td>83.33</td>
</tr>
</tbody>
</table>

2007-04-11
Assessing the maintenance process and building effort estimation models (7/7)

- Oldest vs newest observations prediction performances

<table>
<thead>
<tr>
<th></th>
<th>$L_{\text{Old}} - T_{\text{Old}}$ (%)</th>
<th>$L_{\text{New}} - T_{\text{New}}$ (%)</th>
<th>$L_{\text{Old}} - T_{\text{New}}$ (%)</th>
<th>$L_{\text{New}} - T_{\text{Old}}$ (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MMRE</td>
<td>35.00</td>
<td>27.32</td>
<td>50.40</td>
<td>36.24</td>
</tr>
<tr>
<td>MdMRE</td>
<td>28.34</td>
<td>21.31</td>
<td>34.73</td>
<td>28.65</td>
</tr>
<tr>
<td>PRED$_{25}$</td>
<td>46.48</td>
<td>56.34</td>
<td>35.14</td>
<td>38.89</td>
</tr>
<tr>
<td>PRED$_{50}$</td>
<td>80.28</td>
<td>84.51</td>
<td>60.81</td>
<td>77.78</td>
</tr>
</tbody>
</table>

To confirm stability and maturity of the corrective maintenance process of the organization
Applicability of the effort prediction models (1/3)

- Critique
  - The number of maintenance tasks are not known at the beginning of a project.
  - Distribution of the number of maintenance tasks over monthly periods for project P4.

- Lehman’s laws of software evolution:
  - Self regulation
  - Conservation of organizational stability laws
Applicability of the effort prediction models (2/3)

- Model predictive performance with the average number of maintenance tasks

<table>
<thead>
<tr>
<th></th>
<th>Model A (%)</th>
<th>Model B (%)</th>
<th>Model C (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MMRE</td>
<td>45.12</td>
<td>37.44</td>
<td>38.14</td>
</tr>
<tr>
<td>MdMRE</td>
<td>39.48</td>
<td>33.54</td>
<td>32.13</td>
</tr>
<tr>
<td>PRED$_{25}$</td>
<td>29.17</td>
<td>29.86</td>
<td>34.72</td>
</tr>
<tr>
<td>PRED$_{50}$</td>
<td>63.89</td>
<td>78.47</td>
<td>81.25</td>
</tr>
</tbody>
</table>

- Trend of average number of maintenance tasks for project P4

1. Quickly become stable
2. Increase before getting stable
Model predictive performance with the estimate of the average number of maintenance tasks

<table>
<thead>
<tr>
<th></th>
<th>Model A (%)</th>
<th>Model B (%)</th>
<th>Model C (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MMRE</td>
<td>45.80</td>
<td>44.59</td>
<td>51.36</td>
</tr>
<tr>
<td>MdMRE</td>
<td>39.32</td>
<td>36.04</td>
<td>37.28</td>
</tr>
<tr>
<td>PRED_{25}</td>
<td>27.78</td>
<td>27.08</td>
<td>26.39</td>
</tr>
<tr>
<td>PRED_{50}</td>
<td>67.36</td>
<td>68.06</td>
<td>64.58</td>
</tr>
</tbody>
</table>

The effort predicted by the models using as independent variables the average numbers of maintenance tasks of the first 6 months of each project.

The average number of maintenance tasks of the first few months of a corrective maintenance project can be used to get a roughly but reasonable estimate of the effort.
Assessing predictive performances of the effort estimation models on a new project (1/4)

- Need to improve the model on a new project
  - Hypothesis
    - Type A might be sensibly different than the effort required to perform a task of type B or C
  - Limitation of data set
    - Only total effort of each maintenance period
- Using a specific process management tool (PMT)
  - To overcome limitations
  - To automatically collect and maintain the data about process execution suggested by previous study
Assessing predictive performances of the effort estimation models on a new project (2/4)

- New project
  - PMT collect about 30,000 observations, concerning 7,310 maintenance requests received in first 6 months

- Phases of the corrective maintenance process

<table>
<thead>
<tr>
<th>Phase</th>
<th>Short description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Define</td>
<td>Requirements identification and definition</td>
</tr>
<tr>
<td>Analyze</td>
<td>Requirements analysis</td>
</tr>
<tr>
<td>Design</td>
<td>Design of software modules and test cases</td>
</tr>
<tr>
<td>Produce</td>
<td>Implementation of software modules and execution of test cases</td>
</tr>
<tr>
<td>Implement</td>
<td>Delivery and introduction of the new modules in the software system</td>
</tr>
</tbody>
</table>

Software analysts accept the request
- accepted request is assigned a typology: Change, Defect, other

For each maintenance request
- Effort (on each phase: measured in man-hours)
- Priority (high, medium, low)
Assessing predictive performances of the effort estimation models on a new project (3/4)

- Descriptive statistics of the new project metrics

<table>
<thead>
<tr>
<th>Metric</th>
<th>Min</th>
<th>Max</th>
<th>Mean</th>
<th>Median</th>
<th>Std. dev.</th>
</tr>
</thead>
<tbody>
<tr>
<td>NA</td>
<td>66</td>
<td>96</td>
<td>83.33</td>
<td>83</td>
<td>10.23</td>
</tr>
<tr>
<td>NB</td>
<td>276</td>
<td>472</td>
<td>353.83</td>
<td>348</td>
<td>69.39</td>
</tr>
<tr>
<td></td>
<td>625</td>
<td>927</td>
<td>780.5</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1423</td>
<td>1217.6</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>4857</td>
<td>38.7</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

For the first monthly observation
- Add previous observation
- Recalibrate the models

- Assessed model predictive performance

<table>
<thead>
<tr>
<th>Model</th>
<th>Var.</th>
<th>( b_t \text{ (coef.)} )</th>
<th>( p )-value</th>
<th>( R^2 )</th>
<th>( \text{Adj R}^2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>N</td>
<td>1.342904</td>
<td>&lt; 10 \times 10^{-7}</td>
<td>0.8257</td>
<td>0.8245</td>
</tr>
<tr>
<td></td>
<td>Size</td>
<td>0.169086</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>NA</td>
<td>9.053286</td>
<td>&lt; 10 \times 10^{-7}</td>
<td>0.8891</td>
<td>0.8876</td>
</tr>
<tr>
<td></td>
<td>NBC</td>
<td>0.138275</td>
<td>&lt; 10 \times 10^{-7}</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Size</td>
<td>0.164826</td>
<td>&lt; 10 \times 10^{-7}</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>NA</td>
<td>7.85988</td>
<td>&lt; 10 \times 10^{-7}</td>
<td>0.8963</td>
<td>0.8941</td>
</tr>
<tr>
<td></td>
<td>NB</td>
<td>0.541421</td>
<td>&lt; 10 \times 10^{-7}</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>NC</td>
<td>2.81486</td>
<td>0.000001</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Size</td>
<td>0.130507</td>
<td>&lt; 10 \times 10^{-7}</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Model A (%)</th>
<th>Model B (%)</th>
<th>Model C (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>36.91</td>
<td>31.40</td>
<td>16.60</td>
</tr>
<tr>
<td>32.31</td>
<td>27.29</td>
<td>14.31</td>
</tr>
<tr>
<td>0.00</td>
<td>33.33</td>
<td>83.33</td>
</tr>
<tr>
<td>66.66</td>
<td>66.66</td>
<td>100.00</td>
</tr>
</tbody>
</table>
Assessing predictive performances of the effort estimation models on a new project (4/4)

- Effort distribution box plot with respect to maintenance request types

- Effort distribution among task types
  
  Indicating the presence of statistical outlier
  
<table>
<thead>
<tr>
<th></th>
<th>NA</th>
<th>NB</th>
<th>NC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean</td>
<td>12.78</td>
<td>2.43</td>
<td>1.94</td>
</tr>
<tr>
<td>Std.dev.</td>
<td>20.30</td>
<td>6.86</td>
<td>7.25</td>
</tr>
<tr>
<td>10th percentile</td>
<td>1.75</td>
<td>0.40</td>
<td>0.40</td>
</tr>
<tr>
<td>Median</td>
<td>6.75</td>
<td>1.20</td>
<td>1.00</td>
</tr>
<tr>
<td>90th percentile</td>
<td>30.00</td>
<td>4.00</td>
<td>3.00</td>
</tr>
</tbody>
</table>

Confirm hypothesis:
Tasks of type A generally require greater effort than the other two types
Related Works (1/2)

- Development and evaluation of formal models
  - To assess and predict the software maintenance task effort
    - Decomposition techniques (top down, bottom up)
    - Analogy based estimations
    - Algorithmic approaches (statistical, theoretical) to hybrid or mixed approaches
    - Combining algorithmic methods and expert judgments
Our work is different

- Specialized on the corrective maintenance of legacy systems of an organization
  - Following a well defined process almost complaint to the IEEE Standard for Software Maintenance
- Build effort estimation models using productivity metrics
  - Not present or used in the other works
- Assess the effort estimation models on a new project
Conclusion

- Verify the prediction performances of the models on the new maintenance project
- Verify in a quantitative way the hypothesis that different effort is needed for maintenance tasks of different types
- Improve the understanding of the corrective maintenance process its trend
Discussion

- Metric collection
  - Supported by automatic tools, such as workflow management systems
  - But, needs other technical metrics
    - Such as software complexity metrics

- Data set
  - Real data collected from five corrective maintenance projects
  - But, needs to analyze domain and technological environment
Thank you!!

- End of presentation
Early estimates and accurate evaluations

- Permit to significantly reduce project risks
- Useful to support and guide:
  - Software related decision making;
  - Maintenance process efficiency and parameters assessment;
  - Maintain versus buy decisions, comparing productivity and costs among different project;
  - Resource and staff allocation, and so on.
Management

- Use cost estimates
  - To approve or reject a project proposal
  - To manage the maintenance process more effectively

- Accurate cost estimates
  - Allow organizations to make realistic bids on external contracts
Complex factors

- Predictor variables typically constitute
  - Measure of size or complexity
    - Measured in terms of LOC or a functional size measure
  - Number of productivity factors
    - Collected through a questionnaire

- Size of a maintenance task
  - Used to estimate the effort required to implement the single change
    - Useful for larger adaptive or perfective maintenance tasks during software evolution
A major international software enterprise

Solution Center setup in Italy (in the town of Caserta) by EDS Italia Software adopt corrective maintenance

- Maintaining third party software is the core business of this company
- Therefore the skill level of the maintenance teams involved in the corrective maintenance projects is comparable and in general very high.
**P-value**

- **In statistical hypothesis testing,**
  - The probability of obtaining a result at least as extreme as a given data point
  - Assuming the data point was the result of chance alone

- **A criterion for null hypothesis**
  - Support the null hypothesis
    - If $p$-value is large
  - Reject the null hypothesis
    - If $p$-value is small
**R-square**

- **R**: Correlation coefficient
  - Customarily, the degree to which two or more predictors (independent or X variables) are related to the dependent (Y) variable

- *In multiple regression*,
  - R can assume values between 0 and 1
  - To interpret the direction of the relationship between variables
Assess the quality of future prediction

PREdiction Sum of Squares (PRESS) statistics

- The smaller the PRESS, the better the predictability of the model
- Residual
  - Difference between an observed value of the dependent variable $y$ and the value predicted by the model
- PRESS value: $\text{PRESS} = \sum (\hat{y}_r - y_i)^2$
  - Sum of the squared prediction error
- PRESS residual: $\hat{y}_r - y_i$
  - Value of the $i$th value of the dependent variable as observed in the data set: $y_i$
  - Prediction value from a regression equation trained with all the observation except the $i$th: $\hat{y}_r$
Assess the quality of future prediction (cont’d)

**SPR statistic**

- Sum of the absolute value of the PRESS residuals or prediction errors:
  \[ \text{SPR} = \sum |\hat{y}_r - y_r| \]

- If a large value for the PRESS is due to one or a few large PRESS residual, the SPR statistic may be a more accurate way to evaluate predictability

**PRESS and SPR measures**

- Allow to compare models on the same data set
- But, **cannot be used as indicators alone**
Assess the quality of future prediction (cont’d)

- Statistics can be used as indicators alone
  - Calculated from the model developed using the training data set and evaluated on the test data set
  
  **Magnitude of Relative Error (MRE\(_i\))** on an observation \(i\) is defined as:

\[
MRE_i = \frac{|\hat{y}_i - y_i|}{y_i}
\]

- Value of the \(i\)th value of the dependent variable as observed in the data set: \([y_i]\)
- Corresponding predictive value from the regression equation: \([\hat{y}_i]\)
Class of evaluation criteria (4/4)

- Assess the quality of future prediction (cont’d)
  - Statistics can be used as indicators alone (cont’d)
    - Mean Magnitude Relative Error (MMRE)
      - Average of the MREs
        - MMRE accuracy measure is a clear and intuitive evaluation criterion
        - Regularly used to compare the accuracy of cost estimation models and modeling techniques
    - Shortcoming
      - Value may be strongly influenced by a few very high relative error (RE)
    - Median Magnitude Relative Error (MdMRE)
      - To overcome shortcoming of MMRE
Process management tool (PMT)

- **Main capability**
  - Recording time and effort needed to carry out each phase of a maintenance request
  - Notifying events to the maintenance team members
  - Interfacing existing tools for configuration management

- **Empirical study for new project**
  - Collect about 30,000 observations
    - Concerning 7,310 maintenance requests received in these 6 months
Assessing predictive performances of the effort estimation models on a new project

- Discarding outlier
  - According to heuristic (five outliers)
    - One of type A, B and three of type C

- Assessed model predictive performances (without outliers)

<table>
<thead>
<tr>
<th></th>
<th>Model A (%)</th>
<th>Model B (%)</th>
<th>Model C (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MMRE</td>
<td>37.72</td>
<td>28.06</td>
<td>15.69</td>
</tr>
<tr>
<td>MdMRE</td>
<td>38.68</td>
<td>30.40</td>
<td>13.56</td>
</tr>
<tr>
<td>PRED_{25}</td>
<td>16.66</td>
<td>33.33</td>
<td>83.33</td>
</tr>
<tr>
<td>PRED_{50}</td>
<td>66.66</td>
<td>83.33</td>
<td>100.00</td>
</tr>
</tbody>
</table>
Stensrud and Myrtveit [IEEE TSE, 1999]

Demonstrate that human performances are increased when a tool based either on analogy or on regression models is available

However, their work is not generalized to maintenance activities
Analogy models [ICSE, 1996]

- Based on historical data and previous project experience:
  - Current software project is compared with one or more previous similar projects carried in the same organization to relate their costs
  - Can be viewed as a procedure a historical maintenance database from which appropriate information is extracted
Algorithmic approaches

Involve the construction of mathematical models from empirical data following a well-defined procedure

Boehm[Software Engineering Economics, 1981]

- Presents one of the first approaches to estimate maintenance effort:
  - He extends his COCOMO model for development costs to the maintenance phase through a scaling factor
  - This factor, named annual change of traffic
    - An estimate of the size of changes expressed as the fraction of software total LOC, which undergoes to changes during the year
Lehman and Belady [Statistical Computer Performance Evaluation, 1972]

Propose a model for estimating the effort required to evolve a system from a release to the next

Their model has two terms accounting

- For progressive activities
  - Enhancing system functionality

- For anti-regressive activities
  - Compensating negative effects of system evolution
Ramil [ICSE, 2000]

- Use historical data about different versions of the same application to calibrate effort models and assess their predictive power
  - Best linear regression models extracted
    - Based on coarse-grained metrics
      - Number of added, updated, and deleted subsystems
    - Predict well
      - MMRE = 19.3%, PRED$_{25}$ = 73.3%

- Drawbacks
  - Effort data were not directly available in the data set
  - The models were built and tested only on the first segment of the data set, while yield poorer predictive results over the second segment
Lindvall [Software Process Improvement and Practice, 1998]

- Show coarse-grained metrics
  - Such as the number of classes
    - Have stronger accuracy than other finer-grained metrics on predicting changes
Niessink and van Vliet [ICSM, 1997, 1998]

- Use linear regression analysis and a size measure based on Function Points to predict maintenance effort
  - Prediction accuracy of the produced models
    - Less satisfactory than this paper’s models
    - Best results
      - MMRE = 47%, PRED_{25} = 28%
Eick et al [IEEE TSE, 2001]

Consider software system evolution to define code decay and a related number of measurements

These indexes are used

- To produce a non-linear cost estimation model quantifying the effort required to implement changes
Related Works (9/13)

- Jorgensen [IEEE TSE, 1995]
  - Compare the prediction accuracy of different models
    - Using regression, neural networks, and pattern recognition approaches
    - Last approach involves
      - Decomposition of a precious experience data set to select a subset that is more similar to the current case and is consequently a good starting point for the estimate
  - All the models assume
    - Existence of a size estimate, measured as sum of added, updated, and deleted LOC during a maintenance task which can be accurately predicted and sufficiently meaningful
Related Works (10/13)

- Jorgensen [IEEE TSE, 1995]
  - Compare the prediction accuracy of different models
    - Using regression, neural networks, and pattern recognition approaches
  - Last approach involves
    - Decomposition of a precious experience data set to select a subset that is more similar to the current case and is consequently a good starting point for the estimate
  - All the models assume
    - Existence of a size estimate, measured as sum of added, updated, and deleted LOC during a maintenance task which can be accurately predicted and sufficiently meaningful
Fioravanti and Nesi [IEEE TSE, 2001]

Presented and evaluated a model and metrics for estimation and prediction of adaptive maintenance effort of object-oriented systems
De Lucia et al [JSS, 2003]

- Present effort estimation models resulting from an empirical study of a large incremental and distributed Y2K project
- Analyzed adaptive massive maintenance project was carried out by the same organization of this paper’s study
Caivano et al [ICSM, 2001]

- Present a method and a tool for the estimation of the effort required in software process execution
  - Validate it on a legacy system renewal project
  - Show that fine granularity and model recalibration are effective for improving model performance
  - Verify that the estimation model is process dependent