BELLCORE SLIDE MATERIALS FOR PRESENTATION
SOFTWARE MODELS

• 1965 Cost and schedule estimation models
  - COCOMO, B. Boehm (Software Engineering Economics - 1981), TRW
  - 15 cost drivers, including size = DSI (delivered source instructions)
  - Accuracy +-20% (70% of the time)

• 1980 And quality and reliability
  - Checkpoint, C. Jones (Programming Productivity - 1986), size = SLOC
SOFTWARE MODELS (CON’T.)

- 1980 In-process measurement/forecasting
  - T. DeMarco (Yourdon Associates), including size = function weight, manual approach

- 1984 Software Project Dynamics
  - Abdel-Hamid and Madnick, 1991, MIT
  - 38 cost drivers, including 2 sizes: SLOC and tasks, interacting models across s/w lifecycle, consequences of a decision are dynamic and complex
BELLCORE METHODOLOGY

- Needs Assessment
- Interviews with proj. mgmt/personnel
- Answer questionnaires for each tool
- Create project profile for each tool
- Review results with project personnel
- Calibrate for project uniqueness
- Estimate new projects
- Input actuals & Value realization
## Bellcore Estimation Results

<table>
<thead>
<tr>
<th>Project</th>
<th>Actual</th>
<th>Estimates</th>
<th>COCOMO</th>
<th>Checkpt</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>70</td>
<td>66.8 (-5%)</td>
<td>69.6 (-1%)</td>
<td>62 (-11%)</td>
</tr>
<tr>
<td>B</td>
<td>26</td>
<td>23.9 (-8%)</td>
<td>29.9 *</td>
<td>31.8 (22%)</td>
</tr>
<tr>
<td>C</td>
<td>294.9</td>
<td>254.8 (-13%)</td>
<td>313.3 (6%)</td>
<td>296.9 (1%)</td>
</tr>
<tr>
<td>D</td>
<td>374.2</td>
<td>332.3 (-11%)</td>
<td>378.7 (1%)</td>
<td>390.3 (4%)</td>
</tr>
<tr>
<td>E</td>
<td>25</td>
<td>18.7 (-25%)</td>
<td>23.9 (-4%)</td>
<td>23.8 (-5%)</td>
</tr>
</tbody>
</table>

* Indicates that a constraint was applied to tool to produce estimate.
# BELLCORE ESTIMATION RESULTS

<table>
<thead>
<tr>
<th>Project</th>
<th>Actual</th>
<th>Estimacs</th>
<th>COCOMO</th>
<th>Checkpt</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>8</td>
<td>9 *</td>
<td>8.2 *</td>
<td>9.9 (24%)</td>
</tr>
<tr>
<td>B</td>
<td>10 (0%)</td>
<td>10.5</td>
<td>11.5 *</td>
<td>9.2 (-8%)</td>
</tr>
<tr>
<td>C</td>
<td>29.5</td>
<td>34.4 *</td>
<td>29.5 (0)</td>
<td>29.8 (1%)</td>
</tr>
<tr>
<td>D</td>
<td>20</td>
<td>20 *</td>
<td>21.6 *</td>
<td>23.1 (16%)</td>
</tr>
<tr>
<td>E</td>
<td>7</td>
<td>7 *</td>
<td>8 (14%)</td>
<td>8.2 (17%)</td>
</tr>
</tbody>
</table>
### PROJ. "C" COCOMO ESTIMATION RESULTS

<table>
<thead>
<tr>
<th>Phase</th>
<th>Effort (staff-mo)</th>
<th>Schedule (mo)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Actual</td>
<td>Estimate</td>
</tr>
<tr>
<td>Requirements</td>
<td>101.2</td>
<td>113.6</td>
</tr>
<tr>
<td>Design</td>
<td>89.2</td>
<td>86.7</td>
</tr>
<tr>
<td>Code/UT</td>
<td>64.4</td>
<td>63.6</td>
</tr>
<tr>
<td>Test</td>
<td>40.0</td>
<td>49.4</td>
</tr>
<tr>
<td>Total</td>
<td>294.9</td>
<td>313.3</td>
</tr>
</tbody>
</table>

**KEY TO PROJECT "C" ACCURACY**

- PROJECT "C" WAS MODELED AS TWO INCREMENTS
- NO OVERLAP
- FIRST INCREMENT WAS MODELED TO BE ENTIRELY REQUIREMENTS DEFINITION
FACTORS FOR AN ACCURATE ESTIMATE

- Model the project
  - Use COCOMO Increments & overlap
- Accurately size the project
- Calibrate on project unique factors
- Use tools to bound the estimate
  - ESTIMACS has been calibrated to estimate low, COCOMO high, spread to be between 20%. If Checkpoint lies between, then report CP estimate
SOFTWARE PROJECT DYNAMICS STATUS

- Model is available as shrink-wrap software
- Input parameters fairly well understood
- Questionnaire developed and being used
- Results look fairly reasonable
SOFTWARE PROJECT DYNAMIC RESULTS

- Parameters indicated
- Best in class depends on perceived productivity and estimated schedule
- Some factors which may make a project best in class
  - (19) Nom. potential productivity
  - (25) Schedule compression
  - (29) Test effort needed per error
BELLCORE’S ESTIMATION RISK MODEL

- Three levels of estimation risk have been identified
- Three project sizes
- Estimation risk is a function of product volatility & development environment volatility
- Estimation tools applied to a project are determined by level of project estimation risk
END OF SLIDE PRESENTATION
ALL BELLCORE MATERIALS
One of the most difficult aspects of telecommunications software project development is the accurate sizing and cost estimation for producing a product. Generally, early accurate estimation of software size is crucial to the proper management of a software project. Early sizing errors of up to 200% or more are not uncommon and these can produce symptoms of project stress including late deliveries, budget overruns, and high field failure rates. Therefore, improving early sizing and estimation can dramatically improve customer satisfaction. In an attempt to compare our technologies, Bellcore has teamed up with CSELT of Italy to compare our estimation technologies and to learn from each other. Both teams estimated two Bellcore software projects and two software projects developed by Italian vendors. We will report on the Bellcore efforts.

There is much literature available and many differing opinions exist on the value of estimation tools. In many cases, it appears that software developers are skeptical and test or metrics people are generally supportive of using estimation technology. In an effort to address both of these viewpoints, a Bellcore technology team has developed a new estimation paradigm. This paradigm acknowledges and accepts the skepticism and utilizes methodologies to take advantage of both the project team expertise and the expertise of software estimation experts.

Bellcore combines a set of technologies and processes to help a project team:

- estimate the expected size of the software product
- estimate project schedule, effort, product quality and reliability
- identify process improvements and
- recommend management actions necessary to achieve project goals

The major paradigm shift in our software estimation process is that there is a recognition of both the value and the pitfalls of the "panel of experts" estimation methodology. In this new paradigm, project team members complete their panel of experts estimate and then begin working with the estimation team to input project data into the estimation tools (often Estimacs™, CoStar™, Abdel-Hamid's Systems Dynamics tool, and Checkpoint™). The critical next step is the convergence of the two approaches. The estimation tools are used to produce forecasts of information which are frequently not considered in a project estimate -- impact of process deficiencies on delivered defects, forecasted test cases required to test product, and project weaknesses and strengths. In addition, our software estimation paradigm recommends going back after a project is completed and collecting the actuals.

We will report on the four Bellcore project estimates for Size, Effort, Schedule, and Average Staffing. Some of our clients have indicated that our technology produces more accurate results than they normally obtain. The technology used appears to be very valuable in predicting and developing design reliability process improvement plans. They have begun introducing this technology into their environment.
Bellcore/CSELT Joint Project Estimation Study

- Bellcore Team Members
  Stuart E. Glickman, Scott Duncan, George Mitwasi, Bill Pitterman, Hilla Singh, Amit Dagli, Susan Lichtig, and Eva Owens
Goals and Objectives

- Use estimation tools to accurately estimate software costs, schedules, quality, and productivity
- Calibrate tools for uniqueness, if appropriate
- Gather information to look at standardizing tools, methods, process, etc.
- Begin to develop product/Lab specific historic databases
THE METRIC FUNCTION

THE SPEC

QUANTIFICATION

PRIMITIVE METRICS

COMPOSITION

PREDICTOR

EFFECT VS. PRIMITIVE METRICS

EFFECT VS. PREDICTOR

EFFECT

PROJECT

PROJECT

EFFECT
SOFTWARE MODELS

• 1965 Cost and schedule estimation models
  - COCOMO, B. Boehm (Software Engineering Economics - 1981), TRW
  - 15 cost drivers, including size = DSI (delivered source instructions)
  - Accuracy +/-20% (70% of the time)

• 1980 And quality and reliability
  - Checkpoint, C. Jones (Programming Productivity - 1986), size = SLOC
SOFTWARE MODELS (CON’T.)

- 1980 Function Points
  - Albrecht and Gaffney - 1982
  - size from requirements document

- Software sizing model
  - G. Bozoki, Lockheed,
  - $N(N+3)$ data requirement for four input data sets: pairwise data, pert sizing data, sorting data, and ranking data
SOFTWARE MODELS (CON’T.)

- 1980 In-process measurement/forecasting
  - T. DeMarco (Yourdon Associates), including size = function weight, manual approach

- 1984 Software Project Dynamics
  - Abdel-Hamid and Madnick, 1991, MIT
  - 38 cost drivers, including 2 sizes: SLOC and tasks, interacting models across s/w lifecycle, consequences of a decision are dynamic and complex
BELLCORE TOOLS

- COCOMO
- Checkpoint
- ESTIMACS
- Software Project Dynamics
- Software Sizing Model (SSM)
- Function Point Analysis
BELLCORE METHODOLOGY

• Needs Assessment
• Interviews with proj. mgmt/personnel
• Answer questionnaires for each tool
• Create project profile for each tool
• Review results with project personnel
• Calibrate for project uniqueness
• Estimate new projects
• Input actuals & Value realization
Calibrate on completed projects: Know the answer, can model produce an accurate estimate?
- Size Known
- Attributes as known at end of project
New Projects: Don't know the answer, can an accurate estimate be made?
- Use calibration adjustments
- Size and Attributes to be estimated
BELLCORE RESULTS

- Results
  - Calibration & Estimation
  In Italy and Bellcore
# Bellcore Estimation Results

<table>
<thead>
<tr>
<th>Project</th>
<th>Actual</th>
<th>Estimates</th>
<th>COCOMO</th>
<th>Checkpt</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>70</td>
<td>66.8  (-5%)</td>
<td>69.6  (-1%)</td>
<td>62      (-11%)</td>
</tr>
<tr>
<td>B</td>
<td>26</td>
<td>23.9  (-8%)</td>
<td>29.9  *</td>
<td>31.8  (22%)</td>
</tr>
<tr>
<td>C</td>
<td>294.9</td>
<td>254.8 (-13%)</td>
<td>313.3  (6%)</td>
<td>296.9  (1%)</td>
</tr>
<tr>
<td>D</td>
<td>374.2</td>
<td>332.3 (-11%)</td>
<td>378.7  (1%)</td>
<td>390.3  (4%)</td>
</tr>
<tr>
<td>E</td>
<td>25</td>
<td>18.7  (-25%)</td>
<td>23.9  (-4%)</td>
<td>23.8  (-5%)</td>
</tr>
</tbody>
</table>

* Indicates that a constraint was applied to tool to produce estimate.
### BELLCORE ESTIMATION RESULTS

<table>
<thead>
<tr>
<th>Project</th>
<th>Actual</th>
<th>Estimates</th>
<th>COCOMO</th>
<th>Checkpt</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>8</td>
<td>9 *</td>
<td>8.2 *</td>
<td>9.9 (24%)</td>
</tr>
<tr>
<td>B</td>
<td>10</td>
<td>10 (0%)</td>
<td>11.5 *</td>
<td>9.2 (-8%)</td>
</tr>
<tr>
<td>C</td>
<td>29.5</td>
<td>34.4 *</td>
<td>29.5 (0)</td>
<td>29.8 (1%)</td>
</tr>
<tr>
<td>D</td>
<td>20</td>
<td>20 *</td>
<td>21.6 *</td>
<td>23.1 (16%)</td>
</tr>
<tr>
<td>E</td>
<td>7</td>
<td>7 *</td>
<td>8 (14%)</td>
<td>8.2 (17%)</td>
</tr>
</tbody>
</table>
### PROJ. "C" COCOMO ESTIMATION RESULTS

<table>
<thead>
<tr>
<th>Phase</th>
<th>Effort (staff-mo)</th>
<th>Schedule (mo)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Actual</td>
<td>Estimate</td>
</tr>
<tr>
<td>Requirements</td>
<td>101.2</td>
<td>113.6</td>
</tr>
<tr>
<td>Design</td>
<td>89.2</td>
<td>86.7</td>
</tr>
<tr>
<td>Code/UT</td>
<td>64.4</td>
<td>63.6</td>
</tr>
<tr>
<td>Test</td>
<td>40.0</td>
<td>49.4</td>
</tr>
<tr>
<td>Total</td>
<td>294.9</td>
<td>313.3</td>
</tr>
</tbody>
</table>

**KEY TO PROJECT "C" ACCURACY**

- PROJECT "C" WAS MODELED AS TWO INCREMENTS
- NO OVERLAP
- FIRST INCREMENT WAS MODELED TO BE ENTIRELY REQUIREMENTS DEFINITION
FACTORS FOR AN ACCURATE ESTIMATE

- Model the project
  - Use COCOMO Increments & overlap
- Accurately size the project
- Calibrate on project unique factors
- Use tools to bound the estimate
  - ESTIMACS has been calibrated to estimate low, COCOMO high, spread to be between 20%. If Checkpoint lies between, then report CP estimate
SOFTWARE PROJECT DYNAMICS STATUS

- Model is available as shrink-wrap software
- Input parameters fairly well understood
- Questionnaire developed and being used
- Results look fairly reasonable
SOFTWARE PROJECT DYNAMIC RESULTS

- Parameters indicated
- Best in class depends on perceived productivity and estimated schedule
- Some factors which may make a project best in class
  - (19) Nom. potential productivity
  - (25) Schedule compression
  - (29) Test effort needed per error
S/W PROJECT DYNAMICS RESULTS (CON’T)

- Bellcore uses software project dynamics not as an estimation tool but as a project managers aide

- Software project dynamics is aligned to Checkpoint estimate & then used to help project managers solve problems as they arise
## Systems Dynamics Input Values

<table>
<thead>
<tr>
<th>Expanded Name (from Systems Dynamics)</th>
<th>Label</th>
<th>Dimension</th>
<th>Proj H</th>
<th>Proj F</th>
<th>Proj G</th>
<th>Proj B</th>
<th>Proj</th>
<th>Proj A</th>
<th>Proj D</th>
<th>Proj C</th>
</tr>
</thead>
<tbody>
<tr>
<td>#</td>
<td>SSID</td>
<td>Est</td>
<td>Cal</td>
<td>Est</td>
<td>Cal</td>
<td>Est</td>
<td>Cal</td>
<td>Est</td>
<td>Cal</td>
<td>Est</td>
</tr>
<tr>
<td>1 Average Daily Work Effort per Staff</td>
<td>ADWPF</td>
<td>days</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2 Average Daily Delay for Quality Assurance</td>
<td>ADSDLY</td>
<td>days</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3 Assimilation Delay</td>
<td>ASMDY</td>
<td>days</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4 Average Employment Time</td>
<td>AVEMP</td>
<td>days</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5 Switch to Allow/Disallow Willingness to Overwork (allow = 1)</td>
<td>CTRLSW</td>
<td>undum</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6 Desired Work Delay</td>
<td>DESWP</td>
<td>days</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7 Fraction of Effort Assumed for Development vs Total Project</td>
<td>DEVPRT</td>
<td>fraction</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8 Delay Incorporating Discovered (New) Tasks</td>
<td>DINTC</td>
<td>days</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9 Delivered Source Instructions per Task</td>
<td>DSSTK</td>
<td>source lines</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10 Exhaustion Depletion Delay Time</td>
<td>EXDSTK</td>
<td>days</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11 Hiring Delay</td>
<td>HIRED</td>
<td>days</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12 Initial Staffing (1 = 100%)</td>
<td>INISST</td>
<td>fraction</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13 Switch to Calculate or Enter Total Project Effort (1 = calculate)</td>
<td>MSDWCH</td>
<td>undum</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14 New Hires per Experienced Staff</td>
<td>MNHPFS</td>
<td>staff</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15 Maximum Additions Tolerated w/ Planning Added Effort</td>
<td>MSZTWO</td>
<td>fraction</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16 Maximum Tolerable Exhaustion</td>
<td>MSTXNT</td>
<td>days</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>17 Maximum Allowable Schedule Overage Permitted</td>
<td>MSDKC</td>
<td>percent</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>18 Nominal Fraction of Staff-Day on project</td>
<td>NFNSFD</td>
<td>fraction</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>19 Nominal Potential Productivity of Experienced Employees (takas/staff-day)</td>
<td>NPWPDX</td>
<td>takas/staff-day</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20 Nominal Potential Productivity of New Employees</td>
<td>NNPWNP</td>
<td>takas/staff-day</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>21 Percent Bad Fixes</td>
<td>PBBDFX</td>
<td>%</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>22 Quality Objective (negative indicates undemand)</td>
<td>QDO</td>
<td>%</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>23 Real Job Size in Delivered Source Instr.</td>
<td>RDBDS</td>
<td>source lines</td>
<td>148840000</td>
<td>250000000</td>
<td>250000000</td>
<td>144000000</td>
<td>12</td>
<td>50000</td>
<td>19600000</td>
<td>75950000</td>
</tr>
<tr>
<td>24 Reporting Delay to Project Management</td>
<td>RPTSLY</td>
<td>days</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>25 Schedule Compression</td>
<td>SCHEDM</td>
<td>fraction</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>26 Switch to Calculate or Enter Development Time (1 = calculate)</td>
<td>SCSDWS</td>
<td>undum</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>27 Time to Decide to Allocate Test Staff to Error Backlog (add staff)</td>
<td>TARCMP</td>
<td>days</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>28 Time to Develop</td>
<td>TDEV</td>
<td>days</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>29 Test Effort Needed per Error</td>
<td>TMPFE</td>
<td>staff-days/err</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>30 Time Parameter (recommended)</td>
<td>TMPRMR</td>
<td>days</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>31 Total Effort</td>
<td>TOTMNT</td>
<td>staff-days</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>32 Transfer Delay</td>
<td>TRNSD</td>
<td>days</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>33 Number of New Hires per &quot;Trainee&quot; (Experienced Staff Monitoring)</td>
<td>TRNFRY</td>
<td>people</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>34 Time to Smooth Active Error Density</td>
<td>TSAEDS</td>
<td>days</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>35 Test Overhead (recommended value)</td>
<td>TSTOHC</td>
<td>staff-days/KDS</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
</tr>
<tr>
<td>36 Time to Respond to Changes in Testing Productivity (Productivity)</td>
<td>TSTSCD</td>
<td>days</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>37 Fraction of Project Effort (in Staff-Days) Underestimated</td>
<td>UNDESM</td>
<td>fraction</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>38 Fraction of Task Submission Underestimated</td>
<td>UNDEST</td>
<td>fraction</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Schedule, Headcount, and Defect Results

- **Checkpoint Schedule**: Calendar Months
  - Scheduled: 13,000
  - Days: 5,300
  - Days: 25,200
  - Days: 9,300
  - Days: 15,200
  - Days: 20,200

- **Systems Dynamics Schedule**: SCHDTS
  - Calendar Months: 12,100
  - Days: 9,000
  - Days: 13,300
  - Days: 17,700
  - Days: 6,000
  - Days: 11,200
  - Days: 23,000
  - Days: 20,200

- **Schedule**: % Deviation (SD/CP)
  - People: 46.200
  - Days: -3.226
  - Days: 8,629
  - Days: 13,131
  - Days: 3,134

- **Checkpoint Headcount**: People
  - Days: 4,400
  - Days: 4,800
  - Days: 6,000
  - Days: 6,500

- **Systems Dynamics Headcount**: TOTWF
  - People: 4,400
  - Days: 8,629

- **Headcount**: % Deviation (SD/CP)
  - People: 38.125
  - Days: 37,111
  - Days: 8,629

- **Checkpoint Adjusted Defects**: CMERG
  - Numeric: 278,000
  - Numeric: 472,000
  - Numeric: 478,000
  - Numeric: 286,000

- **Defects**: % Deviation (SD/CP)
  - Days: 0.381
  - Days: 2.767
  - Days: 20,874
  - Days: 30,874

- **Systems Dynamics Perceived Productivity**: PRDPRD
  - Tasks/Staff-Day: 0.65
  - Days: 0.470
  - Days: 1.135
  - Days: 0.300
  - Days: 0.640
  - Days: 0.150
  - Days: 0.180
  - Days: 0.190

---

*9/14/94*
BELLCORE'S S/W PROJ. DYNAMICS QUESTIONNAIRE

- Indicated is Bellcore's questionnaire used in interviewing project personnel
- Other factors required are obtained from Checkpoint and COCOMO
System Dynamics Questionnaire

Note 1: Values expressed in days are based on normal work days, not including holidays, weekends, etc.

Note 2: Every answer entered must be a whole number. No ranges are allowed. However, the Questionnaire identifies sample ranges of values in braces { }.

1. QA Delay
How many days elapse between the time a module is completed and QA activity on it begins? (By QA, we mean development staff planning or performing unit testing, design reviews, and code reviews.)

   Days

   {5-15}

2. Assimilation Delay
How many days does it take a newly hired employee to become productive? (If this project has no new hires, extrapolate the case where you had to hire an inexperienced person.)

   Days

   {30-50}

Note: A newly hired employee is considered productive when he/she is considered experienced enough to work independently without regular supervision.

3. Average Employment Time
A) Will people rotate off this project release? 

   Yes/No

B) How long do people work on this project on average? (Take actual duration of last release, add planned duration for this release, round up to nearest year.)

   Years

4. Overtime Switch
Are people willing to work overtime?

   1=yes  0=no  {1}

Note: If any overtime is worked, answer this as Yes.

5. Rework Delay
On average, how long does it take to repair a defect discovered in design/code reviews or unit test?

   Days

   {5-15}

6. Delay Incorporating New Tasks
How many days does it take from the time that new tasks are identified to the time that they are incorporated into the project schedule?

   Days

   {5-20}

7. Productivity Delay After Exhaustion
How many days does it take to resume normal productivity levels after a period of degraded productivity due to overwork?

   Days

   {10-20}

Note: Overwork is defined as overtime in excess of the agreed-to overtime hours...
System Dynamics Questionnaire

for this project, or overtime in excess of normally accepted overtime in your work environment. (For example, working many consecutive days late into the evening or early morning or many consecutive weekends.)

8. **Hiring Delay**
   How many days elapse between an offer of employment, and the start of work? (Consider required paperwork and administrative tasks) _________ Days {20-80}

9. **Fraction Understaffed**
   Based on your experience, what percent have you generally understaffed your projects? _________ Percent

10. **Number of New Hires Per Experienced Staff**
    A) For this project, how many new hires or inexperienced transferees are on the project? _________ People

    **Note:** An employee is considered inexperienced until he/she becomes fully productive. See Question 2.

    B) For this project, how many experienced employees are on the project? _________ People

    **Note:** These values must be in whole numbers, whether people work full time or part time on the project.

11. **% New Tasks Absorbed**
    What percent increase in project size (Lines of Code) can you absorb without initiating planning for added staff effort (e.g. hiring or allocation of resources)? _________ Percentage {10-25}

12. **Maximum Overwork Tolerated**
    What is the maximum number of consecutive days of overwork your staff can tolerate? _________ Days {5-20}

    **Note:** Overwork is defined as overtime in excess of the agreed-to overtime hours for this project, or overtime in excess of normally accepted overtime in your work environment.

13. **Maximum Schedule Overrun Permitted**
    How many days over your scheduled delivery date can you slip without serious customer complaint? _________ Days {0-50}

14. **Quality Objective**
    What is your planned effort in staff months for QA? _________ Staff-months {1-15}
15. **Reporting Delay**  
How many days elapse between task completion and reporting to project management? _____ Days  
\[5-10\]  
**Note:** Consider the time taken to reflect task completion on the project schedule.

16. **Error Backlog Delay - Development**  
Once you discover, while in development, that there are more errors than planned for, how many days elapse before you assign additional QA effort (i.e., additional personnel or overtime)? _____ Days  
\[5-15\]  

17. **Transfer Delay**  
How many days does it take to move a person to this project from another project? _____ Days  
\[20-90\]  
(If you cannot add new people to the project, then provide a number that is the expected duration of the project + 50%.)

18. **Number of Mentors**  
How many experienced employees on your staff are expected to mentor inexperienced people? _____ People  
**Note:** This value must be in whole numbers, whether people work: full time or part time on the project.

19. **Error Elimination Rate**  
How long does it take to fix a problem and all associated problems? _____ Days  
\[10-40\]  
**Note:** For example, a requirements error discovered may have resulted in several design errors and even more code errors. It may then take several weeks to rectify all errors and produce fault-free testing. (Give careful consideration to your answer. Indicating too short a time to this effort can result in schedule estimation problems.)

20. **Response to Change in Test Productivity**  
Once you notice a change in testing effectiveness, how many days elapse before you incorporate productivity changes into the planned test phase (e.g., integration and system test)? _____ Days  
\[10-20\]  
**Note:** Changes in planned test phase can include such items as addition of test staff, changes in test plan, strategies to reduce test set, acquisition of additional test tools, etc.
21. **Project Effort Estimated**
Based on your experience, what percent have you generally underestimated your project effort? (Consider effort for development and test planning, design, code, unit test, design/code reviews, and all levels of test activities. Do not include effort for requirements phase, project management, documentation, or administrative assistance overhead.)

Consider the previous release. At this point in the lifecycle, for the previous release, what was the project effort estimate compared to the actual final project effort? Please indicate the percent underestimate (- %) or overestimate (+%).

22. **Coding Effort Estimate**
Based on your experience, what percent have you generally underestimated your project size? (Consider requirements added/deleted after estimating project size. Consider change of scope that caused the code to grow/shrink in size.)
EXPERIMENTAL OBSERVATIONS

- Combined estimation tools produce estimates which are accurate to within 20%
- Expertise required for accurate estimate
- In general, ongoing experience with tools are required
- Parts of a project and the total project can be estimated
OTHER OBSERVATIONS

- Checkpoint and S/W Proj. Dyn. provide information useful in project planning:
  - Number of potential defects
  - Expected testing efficiency
- Function point analysis is crucial for accurate estimation
- Future projects can use the data base settings as starting points
CONCLUSIONS

• Estimation tools can be useful in accurately predicting to within 20%

• Standardization of tools, methods, process, etc. needs to continue
  - Current tools seem appropriate as a starting point
  - Questionnaires have been standardized for all tools
  - Methodology is sound and fits within existing policies
BELLCORE'S ESTIMATION RISK MODEL

- Three levels of estimation risk have been identified
- Three project sizes
- Estimation risk is a function of product volatility & development environment volatility
- Estimation tools applied to a project are determined by level of project estimation risk
3. ASSESSING PROJECT ESTIMATION RISK

Prior to estimating a software development project, it is recommended to evaluate the degree of risk involved in committing to an estimate. The risk is evaluated based on the size of the project and the degree of stability of the information available to estimate the project. The larger the project is and the more volatile the project information is, the more effort and expertise should be put into developing the estimate.

The Project Estimation Risk is defined to have one of four levels: very high, high, medium, and low. The guidelines for assessing the degree of risk for a project estimate are based on the fact that the project estimation risk factor is dependent mainly on three elements: project size, product volatility, and development environment volatility. The level of risk involved determines the recommended steps to develop the estimate.

4. RECOMMENDED ACTIONS

The following are the minimum recommended actions in developing a project estimate and are based on the project estimation risk level. The personnel developing the estimates using one or more estimation tools are assumed to have sufficient training on the use of the tools (e.g., COCOMO, ESTIMACS, and Checkpoint). Until sufficient level of training with the tools is achieved, the estimates should be developed with the help of an experienced user of the tools.

<table>
<thead>
<tr>
<th>Project Estimation Risk</th>
<th>Action(s)</th>
</tr>
</thead>
</table>
| Low                     | • Develop a bottom-up estimate  
                         | • Develop a top-down estimate using COCOMO and/or ESTIMACS  
                         | • Compare and iterate the bottom-up and top-down estimates to generate the final estimate |
| Medium                  | • Develop a bottom-up estimate  
                         | • Develop a top-down estimate using COCOMO and/or ESTIMACS  
                         | • If a higher level of confidence in the estimate is desired, consult with an estimation expert to develop a top-down estimate using Checkpoint  
                         | • Compare and iterate the bottom-up and top-down estimates to generate the final estimate |
| High                    | • If possible, initiate a risk reduction plan to reduce the project estimation risk  
                         | • If project estimation risk cannot be reduced any further, develop an initial bottom-up estimate  
                         | • Consult with an estimation expert to assist you in developing and validating the top-down estimate  
                         | • With the assistance of an estimation expert, compare and iterate the bottom-up and top-down estimates and generate the final estimate |
| Very High | - Initiate a risk reduction plan to reduce the project estimation risk  
|          | - If the project risk cannot be reduced, inform upper management of the situation and get their approval to proceed with the project  
|          | - Develop an initial bottom-up estimate  
|          | - Consult with an estimation expert to assist you in developing and validating the top-down estimate  
|          | - With the assistance of an estimation expert, compare and iterate the bottom-up and top-down estimates and generate the final estimate |