Appendix

A.1 A Logbook for Function Point Counts

Note: the following logbook example is an adapted version from the IT department of an international insurance company.

A.1.1 Organizational Information

<table>
<thead>
<tr>
<th>Application System</th>
<th>ZAR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Date:</td>
<td>May 2006</td>
</tr>
<tr>
<td>Reason for Count:</td>
<td>Enhancements per Release April 2006</td>
</tr>
<tr>
<td>Counter:</td>
<td>Mrs. Carolus</td>
</tr>
<tr>
<td></td>
<td>Mr. Alarus</td>
</tr>
<tr>
<td>Application Specialist:</td>
<td>Mrs. Miller</td>
</tr>
<tr>
<td></td>
<td>Mr. Stones</td>
</tr>
</tbody>
</table>

A.1.2 Documentation (Input) for the Function Point count

Documentation for the FP count included the following documents:

- From the Function Point Workbench™ the version »1st count 05.2001« was taken as the basis. Following the quality assurance check and final project delivery, the version »Correction of count 05.2001« was elaborated and stored.
- In the Visio-file ZAR.vsd, the boundary of the system ZAR was illustrated.
- This logbook was used as a basis and was stored for future reporting.
- In the document ZAR-Applications.doc were listed all ZAR-applications with production cycles and jobs (e.g., each PMS-list produced represents an EO from the user perspective).
- The data model in version 3.4 according to the Case Tool xyz was consulted to determine the appropriate data groupings for the ILFs and EIFs.
A.1.3 Architecture, Boundary

![Diagram of ZAR application boundary](image)

Fig. A.1. Example application boundary of ZAR for inclusion in the corporate logbook

A.1.4 Comments Relevant to the Function Point Count

The following text is provided for illustration purposes:

- The log-file from the online portion of the application was counted as an EI. In the batch portion of the application, some listings were crossed out to symbolize that they were counted only once (they were duplicates). See also FP LINK NOTES for the listings.
- The OPC-application CK72B#SPLIT implemented in the fall of 2003 had no new functions from the user perspective. Therefore, this OPC-application was not documented together with the Function Point Application ZAR. It is recommended that the Function Point Applications ZAR and AR be integrated.
- Both systems maintain the same databases that were accordingly counted as ILFs in both applications. Typically, however, a database is only primarily owned and maintained by one application. It was determined that these two applications, while they appeared to be separate from a physical standpoint, actually supported a single set of user functionality and therefore should be counted as one.
- Applications with less than 200 adjusted Function Points deliver outliers in benchmarking, according to our metrics competence center. We know that the rules of thumb can only be used with much care (actually ZAR has 129 adjusted FPs and AR also has 129 adjusted FPs).
A.1.5 Results

![Function Point Summary](image)

**Fig. A.2.** Example count result of application ZAR for inclusion in the corporate logbook

A.2 Checklists

Note: the following checklist examples are adapted and enhanced versions based on existing checklists from the IT department of an international insurance company.

A.2.1 Checklist for Function Point Count Kickoff

Experiences show that to perform a project FP count an average of 1.5–2 days effort should be planned (based on an “average” size project of 500–1,000 FP). A prerequisite to counting is the gathering and assembling of the documentation to support the count (e.g., project documents that describe the functionality from the user perspective). We have found that the most efficient way to do the count is with two persons and a laptop.

Additional people are not necessarily more efficient or productive to the counting process, and can actually detract from the effort if there is wide disagreement about what constitutes the functionality (sometimes it is a status issue to be involved in all meetings on all topics. Do not allow yourself to be drawn into this unproductive situation).
<table>
<thead>
<tr>
<th>Necessary Documentation</th>
<th>Possible Sources</th>
<th>Annotations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overview, characterization of the application</td>
<td>For example, project manual, Intranet</td>
<td>The focus is on the question which user (groups) uses the application and how they use it.</td>
</tr>
<tr>
<td>Architecture of the application and system environment</td>
<td>For example, context diagram, boundary diagram</td>
<td>Most important are the (logical) user interfaces.</td>
</tr>
<tr>
<td>Data model</td>
<td>For example, UDM, data dictionary, segment catalogue, EAR; DB2-catalogue, etc.</td>
<td>If there is no data model available, the database model can be referenced. It must be consulted if there are other functional files that are part of the application that are outside of the database, for example, VSAM-files, etc. (These may actually end up being EIs, EOs, or EQs if they do not meet the requirements for an ILF or EIF).</td>
</tr>
<tr>
<td>Functionality (online and batch)</td>
<td>Model of functions, list of business use cases</td>
<td>The list of business use cases (highest hierarchical level of functionality) often delivers hints for definition of project structuring.</td>
</tr>
<tr>
<td>Online functionality</td>
<td>Direct viewing of the screens, report layouts, user manual</td>
<td>The granularity of the Function Point count aims at «user related elementary processes» (e.g., create, change, etc.); which are often implemented in practice as menus or initiated via PF-keys.</td>
</tr>
<tr>
<td>Batch parts</td>
<td>Model of functions, user manual</td>
<td>The batch parts of the application must be regarded from user view. Often it is helpful to examine the batch portion of the application by results, e.g.: Output processing, printed output (lists, reports, letters, output files, or datasets) Letters for advertisement campaigns Data stores (e.g. administrations, partner organizations), forwarding letters Annual reports or other actualizations Maintenance of central data, consolidations of data Additionally, a view on the changing counting scope during project progress can help to secure the functional completeness of the batch part for the Function Point count: which batch functions run daily, monthly, annually, etc.</td>
</tr>
</tbody>
</table>
The application specialist should have detailed knowledge about the application as viewed by the user(s). Note that knowledge of the physical and architectural (programming) details are not conducive to the FP count but knowledge of the data model definitely is.

*The main criterion for preparation of the count is the availability and currency of the necessary documentation that describes the user requirements for functionality.*

**A.2.2 Checklist for Function Point Count Assessment**

Table A.3. FP count assessment checklist

| Allocated resource responsible for the FP count (expertise and process): |
| Application system-number, Application name, department: |
| Project-number, -name/department: |
| Phase completed when this FP count is to be done (Study, requirements, actualization, project postmortem): |
| Type of count (Project: new development, enhancement, migration; application system: base count, postenhancement base): |
| Date of the FP count: |
| Date of assessment: |

<table>
<thead>
<tr>
<th>Quality criteria</th>
<th>o.k.</th>
<th>not o.k.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1 Has the FP counter attended a FP course? Or</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.2 Does the person performing the count consulted with and secured the availability of a member of the competence center? Reason (only when not o.k. or if the question is not applicable):</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.1 Is the FP counter a project team member (for the project or application being counted)? Reason (only when not o.k. or the question is not applicable):</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.2 Did the project team participate in the FP count? Reason (only when not o.k. or the question is not applicable):</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.1 Has the FP counter more than one year’s worth of participation or knowledge about the basic application system?</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.2 Was the task of »Function Point Counting« included in the project plan? Reason (only when not o.k. or the question is not applicable):</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4.1 Is a FP count of the basic application system documented in the Function Point Workbench™? Where: Or</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(Continued)
<table>
<thead>
<tr>
<th>Process during FP count</th>
<th>o.k.</th>
<th>not o.k.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>4.2</strong> There was adequate written documentation about the basic system? Where:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>– Screen documentation</td>
<td>☐</td>
<td>☐</td>
</tr>
<tr>
<td>– Transaction documentation</td>
<td>☐</td>
<td>☐</td>
</tr>
<tr>
<td>– Interface documentation</td>
<td>☐</td>
<td>☐</td>
</tr>
<tr>
<td>– Database documentation</td>
<td>☐</td>
<td>☐</td>
</tr>
<tr>
<td>– Output documentation (e.g. Print documentation)</td>
<td>☐</td>
<td>☐</td>
</tr>
<tr>
<td>Or</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>4.3</strong> Was the basic system knowledge obtained through documented interviews?</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Reason (only when not o.k. or the question is not applicable):</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>5.1</strong> Is the logbook of the FP count up to date? Where:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Reason (only when not o.k. or the question is not applicable):</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>5.2</strong> Are the assumptions, suggestions, restrictions, and unanswered questions about the project/application documented?</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Reason (only when not o.k. or the question is not applicable):</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>5.3</strong> Is the migration (if there is one) counted and separately documented?</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Reason (only when not o.k. or the question is not applicable):</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>6.2</strong> Are outsourced parts (if there are) counted and separately documented?</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Reason (only when not o.k. or the question is not applicable):</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Process during FP count</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Quality Criteria</strong></td>
<td>o.k.</td>
<td>not o.k.</td>
</tr>
<tr>
<td><strong>1.</strong> Was the FP count done according to the current IFPUG release (n.n)?</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Reason (only when not o.k. or the question is not applicable):</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>2.</strong> Was there a system diagram with the system boundary and data flows? Where:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Reason (only when not o.k. or the question is not applicable):</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>3.</strong> Are the EIs, EOs, and EQs determined by the system boundary and data flows?</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Reason (only when not o.k. or the question is not applicable):</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>4.</strong> Is there an overview about the data files (entities)?</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Reason (only when not o.k. or the question is not applicable):</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>5.</strong> Are the ILFs and EIFs determined by the data files?</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Reason (only when not o.k. or the question is not applicable):</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>6.</strong> Are the Function Points of the EIs, EOs, EQs, ILFs, and EIFs correctly counted (e.g., in case of an enhancement count)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Reason (only when not o.k. or the question is not applicable):</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>7.</strong> Are the 14 GSCs classified according to the organizational standard?</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Reason (only when not o.k. or the question is not applicable):</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>8.</strong> Did the VAF change?</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Reason (only when not o.k. or the question is not applicable):</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(Continued)
9. Were the 14 GSCs compared to the quality attributes?
   Reason (only when not o.k. or the question is not applicable):
   □ □

10. Are the 14 GSCs consistent with each other?
    Reason (only when not o.k. or the question is not applicable):
    □ □

11. Did the requirements change compared to the last FP count?
    Reason (only when not o.k. or the question is not applicable):
    □ □

12. Are the results consistent with other comparable projects in terms of size or with other sizing methods (e.g., SPR-method, number of dialogues × 10, rules of thumb, FP-Prognosis), if available?
    Reason (only when not o.k. or the question is not applicable):
    □ □

13. Did the requirements change (for a delivered project) since the first FP count?
    Reason (only when not o.k. or the question is not applicable):
    □ □

---

### Documentation of the FP count

<table>
<thead>
<tr>
<th>Quality criteria</th>
<th>o.k.</th>
<th>Not o.k.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1 Is the FP Count documented in the Function Point Workbench™ and can it be clearly understood? Where is it stored? Reason (only when not o.k. or the question is not applicable):</td>
<td>□</td>
<td>□</td>
</tr>
<tr>
<td>1.2 Is the description in the Function Point Workbench™ completely answered (per the company standard)? Reason (only when not o.k. or the question is not applicable):</td>
<td>□</td>
<td>□</td>
</tr>
<tr>
<td>2. Is the number of EIs, EOls, EQs, ILFs, and EIFs from the system diagram identical to what the counter has recorded in the Function Point Workbench™? Reason (only when not o.k. or the question is not applicable):</td>
<td>□</td>
<td>□</td>
</tr>
<tr>
<td>3. Are the annotations from the FP count documented? Where: Reason (only when not o.k. or the question is not applicable):</td>
<td>□</td>
<td>□</td>
</tr>
<tr>
<td>4. Are the assumptions and counting decisions for the FP count documented in the logbook? Where? Reason (only when not o.k. or the question is not applicable):</td>
<td>□</td>
<td>□</td>
</tr>
<tr>
<td>5. Are the figures/results comparable to other projects of similar type? Reason (only when not o.k. or the question is not applicable):</td>
<td>□</td>
<td>□</td>
</tr>
<tr>
<td>6. Are the percentages of the EIs, EOls, EQs, ILFs, and EIFs consistent with the rules of thumb? Reason (only when not o.k. or the question is not applicable):</td>
<td>□</td>
<td>□</td>
</tr>
</tbody>
</table>

---

<table>
<thead>
<tr>
<th>Are there any open questions that have to be answered:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nr.</td>
</tr>
<tr>
<td>-----</td>
</tr>
</tbody>
</table>

**Done by:**
**Checked by:**
**Released by:**

---

---

---

---

---

---
### Table A.4. Project postmortem checklist

<table>
<thead>
<tr>
<th>Documentation of the FP Count</th>
<th>Quality Criteria</th>
<th>Yes</th>
<th>No</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Is there an actual/final FP count documented in the Function Point Workbench™? Where?</td>
<td>[Reason (only in case of No):]</td>
<td>[ ]</td>
<td>[ ]</td>
</tr>
<tr>
<td>2. Is there a first FP count from specification phase or earlier besides the project postmortem (delivered) FP count?</td>
<td>[Reason (only in case of No):]</td>
<td>[ ]</td>
<td>[ ]</td>
</tr>
<tr>
<td>3. Was the size of the project tracked during the project progress (requirements creep), i.e., was the delivered FP count compared to the first FP count?</td>
<td>[Reason (only in case of No):]</td>
<td>[ ]</td>
<td>[ ]</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Comparison of Estimate to Actual Effort</th>
<th>Quality Criteria</th>
<th>Yes</th>
<th>No</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Was the original estimate compared with the actual project effort at delivery?</td>
<td>[Reason (only in case of No):]</td>
<td>[ ]</td>
<td>[ ]</td>
</tr>
<tr>
<td>2. Were the differences from the original estimate and the actual effort measured and analyzed together with the competence center?</td>
<td>[Reason (only in case of No):]</td>
<td>[ ]</td>
<td>[ ]</td>
</tr>
<tr>
<td>3. Is the Checkpoint/KnowledgePLAN™ file completed for the project delivery (in particular, the classification of the soft factors, and the project classification completed at the end of the project)? Was the file delivered to the competence center?</td>
<td>[Reason (only in case of No):]</td>
<td>[ ]</td>
<td>[ ]</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Analysis of the Actual Project Effort</th>
<th>Quality Criteria</th>
<th>Yes</th>
<th>No</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Are the records in the time accounting system consolidated and complete?</td>
<td>[Reason (only in case of No):]</td>
<td>[ ]</td>
<td>[ ]</td>
</tr>
</tbody>
</table>

(Continued)
2. Was the actual effort analyzed for IT core team, interfaces, support, and end user participation? What are the percentages of effort for each group?  
Reason (only in case of No):

3. Was the actual effort analyzed by phase (study, specification, etc.)? What are the percentages of effort for each phase?  
Reason (only in case of No):

<table>
<thead>
<tr>
<th>Quality Assurance, Reuse</th>
<th>Yes</th>
<th>No</th>
</tr>
</thead>
</table>
| 1. Is there a list of all detected defects and are they all documented? Where?  
Reason (only in case of No): | ☐ | ☐ |
| 2. Was there any analysis done to project how many defects may occur during maintenance? Where is it documented?  
Reason (only in case of No): | ☐ | ☐ |
| 3. Were checklists and procedures developed during the project? If yes, were they presented to the competence center so that they can be reused on future projects?  
Reason (only in case of No): | ☐ | ☐ |
| 4. Were programs and concepts developed in the project? If yes, were they presented to the competence center so that they can be reused on future projects?  
Reason (only in case of No): | ☐ | ☐ |
| 5. Were checklists, processes, procedures, programs, etc. developed for reuse presented to the staff for communication to the rest of the IT department (e.g., via Intranet, project presentation, organizational newspaper, etc.)?  
Reason (only in case of No): | ☐ | ☐ |

<table>
<thead>
<tr>
<th>Metrics of the Project</th>
<th>Yes</th>
<th>No</th>
</tr>
</thead>
</table>
| 1. Was the productivity of the total project calculated in FP per person month? What was it?  
Note for users who prefer the speed of delivery (PDR) instead: Was the speed of Delivery (PDR) calculated with FP per person month? What was it?  
Reason (only in case of No): | ☐ | ☐ |
| 2. Was the delivery rate for the project calculated per calendar day (FP per calendar day)? What was it?  
Reason (only in case of No): | ☐ | ☐ |
| 3. Was the cost ratio of the project measured in US-$ per FP?  
Reason (only in case of No): | ☐ | ☐ |

(Continued)
Table A. 4. (Cont.)

4. Were the costs analyzed by the following criteria?  
   Costs of internal staff  
   Costs of external staff  
   Costs of central resources (computing center, administration, etc.)  
   Costs of purchased tools, software-packages, etc.  
   Reason (only in case of No):  

5. Was there an analysis done related to “on-time” delivery as \( \text{[abs(actual days – planned days)]} \)? What was the deviation?  
   Reason (only in case of No):  

6. Was the actual defect density calculated (number of defects detected to date post-delivery per FP)? What is it?  
   Reason (only in case of No):  

---

Project post-mortem  
This section pertains to the analysis of the most important problems and crises of the project (3–7 topics) are reviewed in order to deliver preventive and improvement recommendations.

<table>
<thead>
<tr>
<th>Problem, description of crisis or situation:</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Actions taken to mitigate crisis:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Effectiveness of these actions:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Post-project evaluation of the situation and recommendations of future actions:</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

Are there any open questions that have to be answered:

<table>
<thead>
<tr>
<th>Nr.</th>
<th>Question</th>
<th>Who (with whom)</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Name  
Signature

Done by:
Checked by:
Released by:

---

A.2.4 Checklist for Assessment of Estimation

Table A.5. Estimation assessment checklist  

| Name of person responsible for the estimation (for functionality and for the process): |  |  |
| Project number, -name/department: |  |  |
| Phase (Study, requirements, actualization, project postmortem): |  |  |

(Continued)
| Type of project (new development, enhancement, migration): |
| Development platform (Host, PC, C/S, Data Warehouse, Web): |
| Date of project postmortem: |
| Date of quality assurance: |

<table>
<thead>
<tr>
<th>Prerequisites for the estimation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quality Criteria</td>
</tr>
<tr>
<td>1.1 Did the estimator have a Checkpoint/KnowledgePLAN™-training? Or 1.2 Did the estimator have the counsel and availability of a member of the competence center? Reason (only when not o.k. or the question is not applicable):</td>
</tr>
<tr>
<td>2. Was the estimator a member of the project team? Reason (only when not o.k. or the question is not applicable):</td>
</tr>
<tr>
<td>3. Is there a quality-assured first FP count from the end of the requirements phase? Reason (only when not o.k. or the question is not applicable):</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Process of Estimation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quality criteria</td>
</tr>
<tr>
<td>1. Is the logbook of the estimation completed? Where? Reason (only when not o.k. or the question is not applicable):</td>
</tr>
<tr>
<td>2. Did the estimate include a description of the phases it included? Reason (only when not o.k. or the question is not applicable):</td>
</tr>
<tr>
<td>3. Are the assumptions, decisions, and open questions documented? Reason (only when not o.k. or the question is not applicable):</td>
</tr>
<tr>
<td>4. Was the appropriate template for the project estimate been chosen? Which one? Reason (only when not o.k. or the question is not applicable):</td>
</tr>
<tr>
<td>5. Was the appropriate estimation mode chosen (Quick, Detailed Estimate)? Which one? Reason (only when not o.k. or the question is not applicable):</td>
</tr>
<tr>
<td>6. Is the Setup correct (according to the organizational Time Accounting)? Reason (only when not o.k. or the question is not applicable):</td>
</tr>
<tr>
<td>7. Are the hard factors classified and are they plausible? Reason (only when not o.k. or the question is not applicable):</td>
</tr>
<tr>
<td>8. Are the soft factors classified and are they plausible? Reason (only when not o.k. or the question is not applicable):</td>
</tr>
<tr>
<td>9. Have changes to the restrictions of the hard factors as compared to the last estimate been considered? Reason (only when not o.k. or the question is not applicable):</td>
</tr>
</tbody>
</table>

(Continued)
Table A. 5. (Cont.)

<table>
<thead>
<tr>
<th></th>
<th>Have changes to the restrictions for personnel compared to the last estimate been considered?</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>10.</td>
<td>Reason (only when not o.k. or the question is not applicable):</td>
<td></td>
</tr>
<tr>
<td>11.</td>
<td>Have changes to the restrictions for technology compared to the last estimate been considered?</td>
<td></td>
</tr>
<tr>
<td>12.</td>
<td>Have changes to the restrictions for the process compared to the last estimate been considered?</td>
<td></td>
</tr>
<tr>
<td>13.</td>
<td>Have changes to the restrictions for environment compared to the last estimate been considered?</td>
<td></td>
</tr>
<tr>
<td>14.</td>
<td>Have changes of the restrictions for risks compared to the last estimate been considered?</td>
<td></td>
</tr>
<tr>
<td>15.</td>
<td>Was a sensitivity analysis performed?</td>
<td></td>
</tr>
<tr>
<td>16.</td>
<td>Was a comparison made between versions of the estimate?</td>
<td></td>
</tr>
<tr>
<td>17.</td>
<td>Is the actual project effort documented? Where?</td>
<td></td>
</tr>
</tbody>
</table>

Documentation of the Estimation

<table>
<thead>
<tr>
<th>Quality Criteria</th>
<th>o.k.</th>
<th>Not o.k.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Are there any open questions that have to be answered:

<table>
<thead>
<tr>
<th>Nr.</th>
<th>Question</th>
<th>Who</th>
<th>With whom</th>
</tr>
</thead>
</table>

Done by:________________________
Checked by:______________________
Released by:_____________________
A.3 FiSMA Situation Analysis Model MT22

The purpose of the Experience situation analysis method MT22 is to help to estimate annual maintenance and modification projects. The model consists of 22 standard productivity factors. They are classified into four categories: Organization (6 factors), Process (5 factors), Product (6 factors) and People (5 factors). Each factor in each category has five alternative values. The basic idea in rating is that “the better the circumstances for the maintenance are, the more positive rating the factor gets.”

“++” = Excellent situation, circumstances much better than in average case
“+” = Good situation, circumstances better than in average case
“+/−” = Normal situation in the productivity point of view
“−” = Bad situation, circumstances worse than in average case
“−−” = Very bad situation, circumstances much worse than in average case.

Rating of each factor is weighted based on experience data. The ideal or target weights should be 1.10, 1.05, 1.00, 0.95, and 0.90 (from −− to ++) and they should be distributed normally, 5–20–50–20–5%.

<table>
<thead>
<tr>
<th>Category</th>
<th>Name of MT22 productivity factor</th>
</tr>
</thead>
<tbody>
<tr>
<td>Organization</td>
<td>Release and version policy</td>
</tr>
<tr>
<td>Organization</td>
<td>Resource availability for future needs</td>
</tr>
<tr>
<td>Organization</td>
<td>Contracting procedure</td>
</tr>
<tr>
<td>Organization</td>
<td>Number of stakeholders</td>
</tr>
<tr>
<td>Organization</td>
<td>Priority setting and control of changes</td>
</tr>
<tr>
<td>Organization</td>
<td>Organizational culture</td>
</tr>
<tr>
<td>Process</td>
<td>Source code edition methods and tools</td>
</tr>
<tr>
<td>Process</td>
<td>Testing methods and tools</td>
</tr>
<tr>
<td>Process</td>
<td>Documentation methods and tools</td>
</tr>
<tr>
<td>Process</td>
<td>Communication methods and tools</td>
</tr>
<tr>
<td>Process</td>
<td>Roll-out methods and tools</td>
</tr>
<tr>
<td>Product</td>
<td>Functionality requirements</td>
</tr>
<tr>
<td>Product</td>
<td>Reliability requirements</td>
</tr>
<tr>
<td>Product</td>
<td>Usability requirements</td>
</tr>
<tr>
<td>Product</td>
<td>Efficiency requirements</td>
</tr>
<tr>
<td>Product</td>
<td>Maintainability requirements</td>
</tr>
<tr>
<td>Product</td>
<td>Portability requirements</td>
</tr>
<tr>
<td>People</td>
<td>Development environment skills of staff</td>
</tr>
<tr>
<td>People</td>
<td>Application knowledge of staff</td>
</tr>
<tr>
<td>People</td>
<td>Networking skills of staff</td>
</tr>
<tr>
<td>People</td>
<td>Motivation and responsibility of staff</td>
</tr>
<tr>
<td>People</td>
<td>Team atmosphere</td>
</tr>
</tbody>
</table>

Note that FiSMA also supports the ND21 (New Development) situation analysis for gauging the productivity factors on new development projects. See www.fisma.fi/in-english/methods to download this and the MT22 situation analysis presented below.
### A.3.1 Organization Factors

#### Release and Version Policy

The clarity, formality, internal integrity, and long-term planning of future releases and versions.

<table>
<thead>
<tr>
<th>Table A.7. Release and version policy factors</th>
</tr>
</thead>
<tbody>
<tr>
<td>— Many different customer specific versions and delivery packages, which are built case by case after required modifications.</td>
</tr>
<tr>
<td>— Many customer specific versions and release packages, and their installation and deployment is allocated to end-users.</td>
</tr>
<tr>
<td>+/- Some customer-specific versions and releases of the same delivery.</td>
</tr>
<tr>
<td>+ All customers get the same standard delivery. New versions are released according to future needs.</td>
</tr>
<tr>
<td>++ All customers get the same standard delivery. Future releases are planned and agreed for the foreseen future.</td>
</tr>
</tbody>
</table>

#### Resource Availability for Future Needs

Adequacy of resources and systematic allocation of staff, hardware, software, work space, and required skills for the planned maintenance period.

<table>
<thead>
<tr>
<th>Table A.8. Resource and availability for future needs factors</th>
</tr>
</thead>
<tbody>
<tr>
<td>— Organization has no defined rules and practices in resource management. Applications have no responsible person. Continuous lack of resources.</td>
</tr>
<tr>
<td>— Organization has mechanism for workload management, and it is at least partially in use. Each application has responsible person, but he/she has many other responsibilities. Availability of resources is uncertain.</td>
</tr>
<tr>
<td>+/- Organization has defined mechanism to manage critical resources. Each application has responsible person(s). Some delays to get other resources.</td>
</tr>
<tr>
<td>+ Organization has well-defined mechanism to manage all resources, and it is followed largely. Responsibilities are fully allocated to suitable person(s) and also required back up resources are nominated. Other resources are available on request with short notice.</td>
</tr>
<tr>
<td>++ Organization has well-defined mechanism to manage all resources, and it is followed fully. All required responsibilities and back up resources are nominated and their availability is well ensured. Also other resources are available on request.</td>
</tr>
</tbody>
</table>

#### Contracting Procedure

Consistency, completeness, and granularity of maintenance contract to define each service type and/or service transaction, mutual responsibilities, level of services, acceptance criteria of deliveries, and other required contract conditions.
Table A.9. Contracting procedure

--- Maintenance service is not based on any contract or other documented practice.
- Maintenance service is performed according to continuous framework agreement, but separate services and deliveries are not identified.
+- Maintenance service is based on continuous framework agreement, and each service transaction is recorded by supplier and accepted at least orally by customer/end user.
+ Maintenance service is based on separate service agreements and each service delivery is based on mutually accepted documents.
++ Each service type is based on documented mutual agreement, and is an element of continuous framework agreement. Each delivery is based on mutually approved specification document.

**Number of Stakeholders**

Number of people and/or organizations involved in management and decision making of maintenance service and deliveries

Table A.10. Number of stakeholders

--- Number of people and organizations involved in implementation and decision making of change requests is high (both more than 5).
- Either the number of people or organizations involved in implementation and decision making of change requests is high (either number of people or organizations more than 5).
+- Number of people and organizations involved in implementation and decision making of change requests is typical/average (both 2–4).
+ Either the number of people or organizations involved in implementation and decision making of change requests is low (1–2) and the other is not high (not more than 5).
++ Number of people and organizations involved in implementation and decision making of change requests is low (both 2 or less).

**Priority Setting and Control of Changes**

Classification and analysis of change requests by defined criteria (for example, criticality, urgency, and cost) to prioritize change requests and decide on required actions of both parties.

Table A.11. Priority setting and control of changes

--- No agreed classification for errors, failures, and change requests.
- Only application specific error classification is in use.
+- Organization wide error and failure recording and classification is in use and it is used to prioritize fixing actions.
+ Organization has classified each application by business criticality and classifies also each error and failure, respectively. Each change request is analyzed by benefit/cost method.

(Continued)
All applications and error and failure types have widely known criticality classification and consistent benefit/cost analysis method. All responsible persons (operators etc.) are fully aware of all problem situations and required actions.

Organizational Culture

Common attitudes among staff and appreciation of maintenance at company level, appropriate awarding mechanism, and other cultural factors.

Table A.12. Organizational culture

<table>
<thead>
<tr>
<th>Level</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>——</td>
<td>Organization and people are enthusiastic of new technologies and projects only. New development projects are highly appreciated, maintenance “just must.” No visibility for maintenance work, no awarding mechanism for maintenance projects and services.</td>
</tr>
<tr>
<td>—</td>
<td>Importance of maintenance is known, but not shown. No communication and awarding mechanism for maintenance.</td>
</tr>
<tr>
<td>+/-</td>
<td>Organization values maintenance but does not motivate people in maintenance work in any means. Maintenance is mentioned in top management presentations and is part of company-wide measurement program.</td>
</tr>
<tr>
<td>+</td>
<td>Maintenance has good image in company as a key long-term success factor and profit maker. Maintenance is a profession, and is part of recruiting campaigns</td>
</tr>
<tr>
<td>++</td>
<td>Maintenance has good image in company and has strong motivation and commitment among top management and staff. People want maintenance responsibilities and activities. Maintenance is measured at organizational delivery and individual levels and is part of awarding mechanism.</td>
</tr>
</tbody>
</table>

A.3.2 Process Factors

Source Code Edition Methods and Tools

The level and impact of code editors, translation tools, code libraries, and code integrity tools and procedures.

Table A.13. Source code edition methods and tools

<table>
<thead>
<tr>
<th>Level</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>——</td>
<td>Development environment and tools are not in proper use and widely known. Several hardware platforms</td>
</tr>
<tr>
<td>—</td>
<td>Development environment and tools are in moderate use, but are immature and new versions are needed frequently. Some guidelines and standards are in partial use.</td>
</tr>
<tr>
<td>+/-</td>
<td>Development environment and tools are in common use. Guidelines, procedures, and standards are created, but only in partial use.</td>
</tr>
<tr>
<td>+</td>
<td>Development environment and tools are well known and in common use. Guidelines, procedures, and standards are in use and easily accessible.</td>
</tr>
</tbody>
</table>

(Continued)
++ Environments and tools are an integrated set, and automate major parts of manual tasks. Simple, well-known development environment and only one hardware platform.

**Testing Methods and Tools**

Level and impact of tools and procedures to manage test cases and materials, test activities, regression tests, and test results.

<table>
<thead>
<tr>
<th>Table A.14. Testing methods and tools</th>
</tr>
</thead>
<tbody>
<tr>
<td>— No testing practices and standards. Test materials are derived separately each time when required.</td>
</tr>
<tr>
<td>– Testing activities and standards exist, but test case derivation and reuse is difficult. All data is file-based, only manual handling of files and data.</td>
</tr>
<tr>
<td>+/- Testing is well performed and largely supported by standards. Test data is managed with appropriate tools and/or scripts.</td>
</tr>
<tr>
<td>+ One test material package, which can be modified for different test situations. Testing process and appropriate tools are well documented and in proper use.</td>
</tr>
<tr>
<td>++ Each application/software component has well-defined test suite (scripts and materials) for all defined testing phases. Regression testing is tool-supported, where appropriate.</td>
</tr>
</tbody>
</table>

**Documentation Methods and Tools**

Level and impact of tools and procedures to create, manage, and distribute required application documents for maintenance staff and end users.

<table>
<thead>
<tr>
<th>Table A.15. Documentation methods and tools</th>
</tr>
</thead>
<tbody>
<tr>
<td>— No common procedure and widely used professional practices for documentation.</td>
</tr>
<tr>
<td>– No common guidelines and procedures for any documentation, only some version and change control in use.</td>
</tr>
<tr>
<td>+/- Good documentation of each application, change request, and error/failure. Follow-up of documented changes, errors, and failures is in use.</td>
</tr>
<tr>
<td>+ Application documents are well managed, controlled, and maintained. Mostly manual documentation.</td>
</tr>
<tr>
<td>++ Well-defined process for documentation of each application. Documentation is tool-based and in wide, firms use.</td>
</tr>
</tbody>
</table>

**Communication Mechanisms**

Level and impact of methods, tools, and procedures to record, communicate, and handle change requests, errors, and failures.
Table A.16. Communication mechanisms

<table>
<thead>
<tr>
<th>Level</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>—</td>
<td>No defined approach for communicating. Required information is distributed for all potential parties to avoid “communication gap.” Many kinds of media are in use.</td>
</tr>
<tr>
<td>–</td>
<td>Communication mechanism is defined, but only in partial use. Some guidelines are available.</td>
</tr>
<tr>
<td>+/-</td>
<td>Communication mechanism is documented, and in proper use. It is not integral part of maintenance process.</td>
</tr>
<tr>
<td>+</td>
<td>Communication is well integrated with maintenance activities and process. No tools, but some templates and distribution lists are in use.</td>
</tr>
<tr>
<td>++</td>
<td>Multiple tools for communication are in proper use and well aligned with actual work processes. Templates support major part of communication.</td>
</tr>
</tbody>
</table>

Roll-Out Methods and Tools

Level and impact of tools and procedures to roll-out modified programs/applications and related data to operation environment.

Table A.17. Roll-out methods and tools

<table>
<thead>
<tr>
<th>Level</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>—</td>
<td>No defined approach for rollouts. Deployment is work intensive and depends on key staff.</td>
</tr>
<tr>
<td>–</td>
<td>Some documents about rollouts are made and in use. Some separate tools in use.</td>
</tr>
<tr>
<td>+/-</td>
<td>Roll-out is a well-defined process and it is followed largely. One dominant tool to perform roll-out and record roll-out status. No easy traceability and version status of rollouts.</td>
</tr>
<tr>
<td>+</td>
<td>Well-established work process for roll-out. Good version control and traceability.</td>
</tr>
<tr>
<td>++</td>
<td>Roll-out and version control has already long history in organization and is fully automated.</td>
</tr>
</tbody>
</table>

A.3.3 Product Factors

Functionality Requirements

Variety and complexity of the requirements and business rules, level of interfaces.

Table A.18. Functionality requirements

<table>
<thead>
<tr>
<th>Level</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>—</td>
<td>Virginal and complex application area, security critical big (thousands of FPs) multitier system for various, multicultural users. Many authorization levels for users. Some complex, algorithmic functions.</td>
</tr>
<tr>
<td>–</td>
<td>Various user groups and access levels to applications and data. Many interfaces with other systems. Some business rules require special application knowledge from developers and testers.</td>
</tr>
</tbody>
</table>
Some user groups with slightly different access control. Mostly simple business rules.

Only a couple of user groups, only some interfaces with other systems. All business rules are relatively simple.

Only one user group, all have same access control. No interfaces with other systems. Functionality is simple data movement to and from user (screens, reports).

**Reliability Requirements**

Severity of failures and impact of failures to users and operation.

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>——</td>
<td>Operation faults may endanger human lives or cause great economic or environmental losses, the application must recover without losing any data in any case.</td>
</tr>
<tr>
<td>—</td>
<td>Failures can cause major economic loss and image suffering, can lead to negative news in mass media.</td>
</tr>
<tr>
<td>+/-</td>
<td>Faulty operation can cause harm for some hundred users, can reflect negatively in operation of 2–3 other applications.</td>
</tr>
<tr>
<td>+</td>
<td>Failures can cause harm for some tens of users, but they can tolerate short operation breaks even daily. Some impacts in max one other application.</td>
</tr>
<tr>
<td>++</td>
<td>Failure has impacts only in some users. Maximally weekly or monthly operation period, error can be fixed without operational losses. No impacts on any other applications</td>
</tr>
</tbody>
</table>

**Usability Requirements**

Number of users, support for various skill levels of users, continuous operation, special requirements to attract users.

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>——</td>
<td>A very big number of different types of end-users all over the world, with different levels of experience at software usage, a high-level customization and help facilities required. 24 h/day, 7 days/week operation requirement.</td>
</tr>
<tr>
<td>—</td>
<td>2–3 different types of users with various skills and languages, requiring automated multilevel help function, the use of software during interactive customer service. 24 h/day operation requirement.</td>
</tr>
<tr>
<td>+/-</td>
<td>Limited number of regular users, who can be trained in advance. Mostly in back-office use, sometimes in direct customer service. Max 20 h/day operation.</td>
</tr>
<tr>
<td>+</td>
<td>Application for small number of users. Only in back-office functions. Operation in working hours is required.</td>
</tr>
<tr>
<td>++</td>
<td>Only few expert users or one team, all located at one site, not very frequent use.</td>
</tr>
</tbody>
</table>
Efficiency Requirements

Requirements for response and transaction processing time, differences in operational and computer load, transaction and data volumes.

Table A.21. Efficiency requirements

| +++++ | Very big volume of real-time transactions, big differences in operation load, need for simultaneous online and batch processing. Millions of records in database, many kinds of nonpredictable inquiry needs. |
| +++++ | Hundreds of simultaneous end-users in multiple sites, most of response time requirements critical, queuing in transaction processing causes operational loss for services. |
| +/+-  | Max one hundred simultaneous end-users. Response time requirements are flexible but critical for work efficiency, mostly only predefined inquiry needs. |
| +     | Simple database, straightforward, and predictable data requests from few simultaneous end-users. |
| ++    | Simple and small database, no simultaneous end-users or complex data requests, total number of transactions not more than tens per day. |

Maintainability Requirements

Stability of the environment, standardized code and component structures, clarity of architecture, pressure for changes.

Table A.22. Maintainability requirements

| +++++ | Very large strategic (target lifetime more than 20 years) software at a volatile business area with frequent changes of laws and standards and business rules. Also the maintenance speed is essential, logging and the defect messages must be clear, exact, and instructive for developers |
| +++++ | Large software (target lifetime from 10 to 20 years), frequent changes of laws or standards or business rules. Time to analyze defect messages, change the programs and test them is always some hours but not more. |
| +/+-  | Average size tactical (target lifetime from 5 to 10 years) software, monthly changes of laws, standards and business rules. Maintenance timing is reasonably flexible, a couple of days rather than hours, an application specific error log needed. |
| +     | Rather small rarely changing software (target lifetime from 2 to 5 years), no application specific diagnostics needed. |
| ++    | Temporary software (target lifetime less than 2 years) with no intention to enhance for new requirements. |

Portability Requirements

Adaptability and installability to different environments, openness of architecture and structural components, volatility of platforms and environments.
Table A.23. Portability requirements

--- Users of the software are located in many kinds of organizations, with various platforms (hardware, browsers, operating systems, middleware, data communication protocols, etc), various versions, and various upgrading frequencies.

− The software must operate on many different platforms (hardware, browsers, operating systems, middleware, data communication protocols, etc) and on several versions of them.

+/− Every version of the software must run on several versions of a certain platform (hardware, browser, operating system, middleware, data communication protocol, etc), the upgrading frequencies of the users are rather predictable.

+ The software must run on a certain platform (hardware, browser, operating system, middleware, data communication protocol, etc), for which the software is tested. Only one “latest version” of software is required. Some customers or user groups may use older versions, but they do not need to be interoperable with new version.

++ The software must run only on a certain platform (hardware, browser, operating system, middleware, data communication protocol, etc) in which upgrading process is completely manageable (for example, most of the mainframe environments). Several tens of similar applications are running on the same platform.

---------

A.3.4 People Factors

Development Environment Skills of Staff

Experience and knowledge of maintenance staff in development environment, tools, and platforms (design, implementation, testing, version control, operation, documentation, communication)

Table A.24. Development environment skills of staff

--- Development environment and tools are new for the whole maintenance staff. The average experience time is less than 3 months. Special expertise is difficult to get. Training needs are not satisfied.

− At least one responsible person has reasonable knowledge of environments (3 months to 2 years). Special knowledge is difficult to get. Training is partially available.

+/− At least one of some responsible persons has good knowledge of environments (several years). Average experience is 1–3 years. Special knowledge is largely available on request. Training is available on essential tools.

+ All responsible persons know well the environments and tools (2–5 years). Some persons can give hands-on support in tools. Training is available on all tools.

++ The software must run only on a certain platform (hardware, browser, operating system, middleware, data communication protocol, etc), which upgrade process. The whole maintenance staff knows all the tools very well (>5 years experience). Support available for the specific needs of the project. No need for training.

---------


**Application Knowledge of Staff**

Knowledge of the maintenance staff in the applications and interfacing systems (both the supplier and the customer).

**Table A.25. Application knowledge of staff**

| -- | The business area knowledge of maintenance staff is very small, less than 12 months. No expertise on interfacing systems. |
| - | The application experience is small on vendor side, and software knowledge is small on customer side. Maintenance staff has no special knowledge on interfacing systems. |
| +/- | Maintenance staff has quite good experience of the business area and application domain, 1–3 years in average. At least some people have good overall understanding of the application portfolio. |
| + | The business area and application domain experience is good both on the supplier and the customer sides. The experience is 3–6 years in average, some have >5 years experience. |
| ++ | Both the supplier and the customer representatives know the business area and application domain very well (in average >5 years), including the understanding of the business as total. Good understanding of application portfolio among the whole maintenance staff. |

**Networking Skills of Staff**

Level of team building and networking among maintenance staff, ability to cooperate with partners.

**Table A.26. Networking skills of staff**

| -- | Maintenance staff consists of new people, no mutual working history and experiences. Responsible persons have no common language. No connection with external stakeholders. |
| - | Part of maintenance staff has common working history, max 2 years. Management and experts have mutual communication and understanding problems. Ad hoc connections with stakeholders. |
| +/- | Maintenance staff has some year’s common working history (2–5 years). Mutual communication is open and works quite well. Cooperation with stakeholders is done when required. |
| + | Maintenance staff has long common working history (>5 years). No mutual communication problems between management and experts. Regular cooperation with stakeholders. |
| ++ | Maintenance staff has very long common working history (>10 years). No communication problems between management and experts. Stable and continuous cooperation with stakeholders, even when responsibilities are allocated to new staff. |
**Motivation and Responsibility of Staff**

Personal motivation to develop application and customer business

*Table A.27. Motivation and responsibility of staff*

<table>
<thead>
<tr>
<th>Score</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>—</td>
<td>Maintenance staff has no interest to develop application. Maintenance is considered as mandatory extra duty, which should be avoided. Easy to transfer responsibility to other staff members.</td>
</tr>
<tr>
<td>—</td>
<td>Maintenance staff members are not interested to develop application, expect some limited responsibilities that are explicitly allocated to them.</td>
</tr>
<tr>
<td>+/-</td>
<td>Maintenance staff members are performing maintenance activities according to plan and take the development responsibility.</td>
</tr>
<tr>
<td>+</td>
<td>Maintenance staff members are interested to develop application and take personal responsibility over the whole application area, as defined.</td>
</tr>
<tr>
<td>++</td>
<td>Maintenance staff is interested to develop customer’s business, like introduction of new technology, competitive position of applications, and new changes in interfacing systems. Real responsibilities are far over the minimal requirements defined in maintenance contract.</td>
</tr>
</tbody>
</table>

**Team Atmosphere**

Influence on working conditions, self-learning, professional career opportunities.

*Table A.28. Team atmosphere*

<table>
<thead>
<tr>
<th>Score</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>—</td>
<td>Maintenance staff feels that their work effort is highly underappreciated. Continuous lack of resources. No influence on daily work and working conditions. Unfair or unknown feedback on work.</td>
</tr>
<tr>
<td>—</td>
<td>Maintenance staff feels that their work is underappreciated, and leads to at least temporary resource conflicts and inadequate training. Only some influence on daily work at individual level, weak feedback on work results.</td>
</tr>
<tr>
<td>+/-</td>
<td>Maintenance staff feels that their work is moderately appreciated. Resourcing and training are quite adequate. Mostly good influence on daily work, sometimes resource conflict with continuous responsibilities and project duties.</td>
</tr>
<tr>
<td>+</td>
<td>Maintenance work and results are well appreciated. Resourcing and training are adequate. Each individual has good influence on daily working arrangements. Good feedback from work, fair awarding.</td>
</tr>
<tr>
<td>++</td>
<td>Excellent feelings about maintenance work among the whole staff and management. Resourcing and training are adequate. Good knowledge on all feedback from management and customer, awarding is fair. Full responsibility and self-control at individual level on personal working conditions and satisfaction of new professional requirements.</td>
</tr>
</tbody>
</table>
Literature


ASMA, Sizing in Object-Oriented Environments, Australian Software Metrics Association, Victoria, Australia, 1994


Boehm, B. W., Characteristics of Software Quality, North Holland, Amsterdam, 1978


Boehm, B. W., Requirement-Led Project Management, Discovering David’s Slingshot by Robertson, Suzanne and James, Pearson Education, 2005


Crosby, P. B., Quality is Free, Mentor Books, New York, NY, 1979


Dekkers, C. A., Forselius, P., Increase ICT Project Success with Concrete Scope Management, in Proceedings of the Project Management Institute 2007 Asia Pacific (Hong Kong) and 2007 North American (Atlanta) congresses, 2007


Dumke, R., Foltin, E., Koepppe, R., Winkler, A., Softwarequalität durch Meßtools, Vieweg, Wiesbaden, 1996


Fetcke, T., The warehouse software portfolio, a case study in functional size measurement, Technical report no. 1999-20, Département d’Informatique, Université du Quebec à Montréal, Canada, 1999


Forselius, P., Discipline to Software Development – Learn to Distinguish Project Management from Other Management Levels, in: Proceedings of the 19th IPMA Global congress, New Delhi, India, November 2005


Gupta, R., Gupta, S. K., Object Point Analysis, in: IFPUG 1996 Fall Conference, Dallas, Texas, 1996


Herrmann, O., Kalkulation von Softwareentwicklungen, München Wien, 1983


Humphrey, W. S., Managing the Software Process, Addison-Wesley, Reading, MA, 1989


Jones, C., Geriatric Issues of Aging Software, March 17, 2007


Kearns, D. T., as quoted by Beth Enslow, American Programmer, (now Cutter IT Journal), 1992


McKinlay, M., The Top Ten (Or So) Reasons For Not Implementing EVM, IPMA/ICEC European Conference April 2006, Ljubliana, Slovenia


Myerson, B., Email posting on now-defunct CRI List-serv about Unified Modeling Language and IFPUG Function Point, June 16, 1999. Quantimetrics, South Africa.

Nagano, S., Ajisaka, T., Improvement of analysis model by removing improper parts based on functional size measurement, in: Proceedings of the 15th


Rule, G., “Small project’, ‘medium-size project’ and ‘large project’: what do these terms mean? PowerPoint presentation. Software Measurement Services Ltd. (SMS), copyright 2004-2005, 124 High Street, Edenbridge, Kent, United Kingdom, TN8 5AY.


Sneed, H. M., Software-Qualitätssicherung für kommerzielle Anwendungssysteme, R. Müller, 1983

Software Engineering Institute (SEI), The Capability Maturity Model Integration (CMMI®) for development version 1.2, Carnegie Mellon University, Pittsburgh, PA, www.sei.cmu.edu/cmmi, August 2006


Wittmann, W., Unternehmung und unvollkommene Information, Köln, Opladen, 1959


Index

90% finished syndrome, 202, 231

A
acceptance, 5, 85, 96, 98, 101, 102, 105, 106
the king’s road to, 85, 98, 103, 192
adaptive maintenance, 161
algorithms, 123
Analysis Method, 138, 139
application
base, 72
boundary, 70, 71, 80, 456
boundary diagram, 410
enhancement, 340
enhancement count, 352
Function Point count, 352
portfolio, 89
registry, 61
specification, 67
structuring, 66
APQC, 291, 318
architecture diagram, 71, 80, 416
architecture diagram example, 421
Assembler equivalent, 133, 134, 210
average FP component
complexity, 417

B
backfiring, 110, 133, 134, 135, 136, 215, 295
bad fixes, 220
Balanced Scorecard, 198, 266, 267, 268
implementation, 268
model, 267
baseline, 61, 62, 88, 89, 455
bathtub curve, 169, 418
acceptance, 292
benefits, 293
data quality, 294, 295, 296
database survey, 297
experiences, 293, 294, 295, 296
experiment, 146
external, 291, 296
implementation, 291
internal, 291, 296
management support, 292, 293
peer group, 296
process, 290, 292
providers, 296
surveys, 292, 294, 296, 303
time accounting, 69
benchmarking organizations, 317
links, 318

C
CAME tools, 550
case study
case study comparison, 531
case study COSMIC, 533, 534, 535, 536, 538, 539
boundary, 536, 537
data groups, 537
data groups, 499
data movements, 539, 540
functional processes, 501
layers, 536
trigger, 502
use case diagram, 493
use cases, 484
user requirements, 533, 534
user requirements, 483, 484
case study FiSMA
  KISS quick, 511, 512
  KISS quick results, 512
  measurement process, 510, 511
  results, 514
  user requirements, 510, 511, 512
case study IFPUG, 541, 542, 543, 544
data functions, 516, 517
measurement process, 516
requirements, 542
result, 522, 531
transactional functions, 519, 520
use cases, 519, 520
user requirements, 519
case study Mark II
  result, 524
case study NESMA
  transactional functions, 528
  use cases, 528
CER, 232
change metrics, 201
CHAOS Report, 233, 327
checklist, 35
estimation, 38
estimation assessment, 596
Experience Situation Analysis
  Model MT22, 599
FP count assessment, 591
FP count kickoff, 590
Function Point count, 354
Function Point count
  assessment, 355, 356
project post mortem, 591, 594
Chidamber and Kemerer, 244, 247
class metrics, 242, 243, 244, 248, 250
CMMI®, 14, 57, 58, 186, 261, 262, 263, 264
  homepage, 261
  level 2, 13, 187, 217, 227, 263, 278
  level 3, 27, 227, 263
  level 4, 263
  level 5, 263
  levels, 262
  manual, 261
CMMI®-DEV, 263
CoBRA method, 446, 447
COCOMO II, 67, 119, 144, 145, 215, 233, 447
  Application Composition
    Model, 145
cost drivers, 447
Early Design Model, 145, 448
embedded mode, 144
organic mode, 144
Post Architecture Model, 145
semidetached mode, 144
tool, 145
variants, 145
COCOMO-M, 167
code size adjustment factor, 135, 136
communication, 85, 96, 104
Compass benchmarking database, 35
Compass method, 296
Compass metrics, 198
competence center, 7, 8, 89, 93, 95, 96, 98, 99, 105, 190, 296, 339
  benefits, 96
Computer Measurement Group
  (CMG), 269
consultants, 101
continual improvement, 56
contract metrics, 329, 330, 331
  arbitration, 331
  examples, 331
conversion - see migration, 72
core metrics, 193, 197, 199
corrective maintenance, 161
COSMIC, 130, 269, 271, 412
  CFP, 371
  counting process, 369, 370
  CPM, 269
  data movements, 370, 371, 372
  elementary process, 371
goals, 369
  IFPUG FPM comparison, 375, 376
  Implementation Guide, 270, 369
  layers, 371, 372, 373, 374
  layers examples, 372, 373
  measurement model, 370
  MPM, 270, 367, 368
team, 368
  UML, 253
website, 368
  COSMIC Function Points, 167, 269, 270
COSMIC Method, 13, 38, 121, 270, 271, 333, 368, 369
  advantages, 369, 374
cost
  challenges, 233
  correcting factor, 234
  data, 232, 233
  excess, 233
  overrun, 233, 234
  per Function Point, 233, 234
cost estimating, 232
cost metrics, 232
COTS software, 72, 73, 74
counterarguments, 98
customizing, 72, 73
cybernetic control circuit, 47, 53, 199
cyclomatic complexity, 167

D
DACS, 297, 550
dashboard metrics, 292
DASMA, 25, 273
data collection, 89, 90, 92
data complexity, 222, 223
data conversions, 109
Data Point Method, 121
data quality, 418
Data Warehouse System (DWS), 433, 434
defect metrics, 217, 219, 220
  benefits, 217
  examples, 217
  ISBSG, 220
defect potential, 218
defect rates, 217, 219, 220
  examples, 218
  rule of thumb, 218
defect removal, 219, 220
defect severity levels, 219
Definitions
  actor, 323
  adaptive maintenance, 161
  aggregation/consolidation, 437
  application, 340, 341
  application boundary, 456
  backfiring, 133
  benchmarking, 290, 291
corrective maintenance, 161
  COSMIC BFC, 370
  COSMIC data movements, 369, 370
  COSMIC layers, 373
  COTS software, 72
counting scope, 456
customizing, 72
defect potential, 218
  DET, 345
  DI, 350
Dimensions, 437
drill down, 437
duration, 315
efficiency, 231
effort, 226
EI, 347, 461
EIF, 345, 458, 460
enhancement, 159, 455
EO, 347, 461
EQ, 347, 462
estimation process, 124
expansion rate, 133
expert estimation, 11
FiSMA BFC, 377, 378
FSM, 365
FSM Method (FSMM), 366
FTR, 349
functional size, 121, 323, 365
functional size measurement (FSM), 365
functional user requirements (FUR), 121, 323, 365
GSC’s, 350
Hierachy, 437
Hierarchy Dimensions, 437
ILF, 345
indicators, 181, 183
language level, 133
maintain, 458
maintenance, 159, 161, 164
measures, 180, 181, 182
metrics, 181, 182, 183, 186
new development, 455
OLAP, 437
PDR, 313
perfective maintenance, 161
process metrics, 225
product metrics, 208
productivity, 226
RET, 345, 346
rework, 228
rol up, 437
rule of thumb, 419
size, 208
Star Scheme, 435
system boundary, 342
system complexity, 222
time accounting, 68
user, 323
VAF, 350
VAFA, 352
value, 181, 182
degree of completion, 202
Delphi Method, 137
Desharnais study, 376
development costs, 330
devils square, 49, 50, 56, 222
direct metrics, 207
documentation, 27, 87, 99, 222
effort for, 222
documentation metrics, 222
DoD, 261, 262
DoD ARF dataset, 297
DWS
FP count, 434
ISBSG FP component proportions, 434
OLAP Design, 437
Star Scheme, 435
system boundary, 434
E
early estimation, 87
early Function Points, 409, 411
benefits, 416, 417
formulae, 411
early warning
indicators, 183
signs, 150
system, 29, 110
Earned Value Method, 110, 201, 202
efficiency, 198, 231, 314
PDR, 232
effort, 77, 226
distribution, 77, 79, 118
maintenance, 418
VW formula, 418
elementary process, 347
end-user efficiency, 100
end-user engagement, 327
enhancement, 159, 161, 164, 166, 168
effort, 163
NESMA standard, 163
supermarket analogy, 159
tasks, 165
enhancement project, 71, 72, 73, 109, 160, 166, 170, 340, 342, 343, 392
examples, 352
FP’s, 351, 352
ERP systems, 72, 73, 74, 443
ESA database, 67
ESA/INSEAD database, 297, 298
ESMIT survey, 198
estimation, 1, 9, 13, 14, 15, 22, 24, 26, 45, 71
acceptable, 58
acknowledgement, 124, 125
awareness, 89
bargaining, 32, 547
best case, 76, 137
challenges, 1, 90
checklist, 61, 62, 74
concepts, 8
conference, 36, 60, 66, 75
consciousness, 16
controllable, 30
controlling, 45, 46
cost drivers, 447
culture, 8, 32, 36, 37, 43, 88, 547
cybernetic control circuit, 53
documentation, 30, 47, 74, 75, 80, 138
documentation, 30, 47, 74, 75, 80, 138
early, 1, 31, 59, 120, 212, 310, 409, 410, 412, 417
effort for, 93, 99
equations, 126, 127, 128
errors, 31
experience, 60, 74, 76, 80, 89, 95, 96, 138
experience curve, 126, 127, 128
expert system, 129, 131, 132
factors, 14
goals, 56, 58, 85, 105, 107
heuristic methods, 137
honesty, 36, 37, 547
implementation, 57, 85, 87, 89, 98, 105, 106
Internet Links, 39
introducing, 56
logbook, 80
manual, 87, 109
methods, 118, 137
methods survey, 119
milestones, 59, 60, 61
minimum size, 366
most important themes, 8
motivation, 37
object, 13, 14, 19, 20, 21, 36, 66, 117
of costs, 118
of effort, 13, 15, 16, 117, 119, 120, 126, 128, 129, 130, 137, 138, 139
of packages, 72, 73
paradox, 5
parameters, 47, 48, 49, 56, 57, 76, 80, 566
planning, 31, 47
political, 32
pragmatic rules, 16
precision, 16, 18, 19, 23, 150, 151, 153, 358, 447
precision study, 152, 153, 154
preparation, 65, 66, 76
prerequisites, 31
process, 15, 30, 39, 76, 77, 117, 118
quality, 31, 50
repeated, 22, 29, 59, 61
right time for, 58
risk add on’s, 62, 76
risk-oriented, 58
risks, 11, 31
rules, 29
rules of thumb, 412
scenarios, 80
strategic, 47, 56
tool support, 59, 60, 73, 75, 76, 80, 547, 548
tracking, 23, 29, 31, 60, 61
training, 38
trimmed, 32
uncertainty, 11, 31, 548
validation, 76
worst case, 76, 137
estimation conference, 449
estimation coordinator, 96, 98
estimation methods
acceptance, 148
evaluation, 147, 149, 150, 151
estimation process
example, 124
estimation tools, 6, 7, 8, 26, 88, 89, 119, 123, 124, 131, 547
acceptance, 549, 572
assignment scope, 554
basic relations, 554
benefits, 549
comparison, 555
customization, 564
duration, 554
effort, 550
estimation improvement, 564
estimation modus, 561
Experience® Pro, 573, 574, 576, 577
for OO projects, 581, 582, 583
Function Point Workbench (FPW), 578, 580
hard data, 558
input parameters example, 562
integration, 552, 553
Internet resources, 584
KnowledgePLAN, 555, 556, 559
list, 550
Logiscope, 582, 583, 584
macro estimation, 554
micro estimation, 554
output example, 562, 563
overestimation, 564
planning support, 549
process, 553
production rate, 554
project portfolios, 564, 565, 567, 568
project portfolios example, 568
569, 570, 572
sensitivity analysis, 555
simulations, 555, 564, 567
simulations benefits, 565
simulations example, 565, 566, 567
size, 552
soft data, 559
soft data example, 560
Soft-Calc, 578
time accounting, 565
usage, 553
exchange of experiences, 88, 105
expansion rate, 133, 135
experience database, 76
expert estimation, 10, 11, 12, 137
external metrics, 207
F
Feature Point Method, 121, 123, 130
feed forward, 25, 62, 188
feedback, 53, 90, 91
feedback loop, 54, 56
Fetcke study, 376
financial metrics, 267
FiSMA, 15, 160, 271, 366
BFC, 379, 380, 381, 382, 383, 384, 385, 386
Index

advantages, 120, 333, 336
application areas, 329, 332, 333
benefits, 326, 328, 329
boundaries, 358
disadvantages, 335
evaluation, 334, 335, 358
goals, 336
history, 325, 326
implementation, 356, 357
Internet resources, 324
prejudices, 336, 337
problems, 121, 122
variants, 405
Function Point Prognosis, 59, 338, 410, 414, 417
formulae, 410, 411, 412
from documents, screens and files, 412
from files, 412
from ILFs, 389
from transactions, 409, 410, 412
Function Points, 89, 133, 211
adjusted, 52, 344
automatic counting, 212, 213, 555
average, 414, 415, 417
conversion formulae, 306, 376, 388
counting, 211
counting experiences, 25
EI complexity, 349
EO and EQ distinction, 462
EO/EQ complexity, 349
files, 343, 344, 345, 346, 347
files complexity, 345, 346
files complexity example, 347
governmental regulation, 268
installed, 71
object-oriented, 122
one file model, 309, 414
OO PDR, 251
OOA, 242, 251
OOD, 242
packages, 72, 73, 449
database, 272, 377, 386
manual, 380
measurement process, 379
FiSMA Method, 123, 124, 130, 271, 376, 386
scope, 379
team, 387
FP components, 343
complexity, 344
proportions, 413, 415
FP counting scope, 341
Fraunhofer IESE, 273, 446
FSM, 260, 268, 274, 366
Function Point components, 447
Function Point coordinator, 97, 98
Function Point count, 24, 60, 62, 72, 76, 87
assessment, 355
assessment checklist, 355, 356, 591
checklist, 354
DET example, 424, 425, 426
documentation, 353, 354
effort for, 170, 332, 356, 357
EI example, 422, 423, 425, 426
EIF, 422
EIF example, 422
EO example, 425, 426
EQ example, 427, 428, 429
FTR example, 427
ILF example, 421
logbook, 416
logbook contents, 355
logbook example, 587, 588, 589
precision, 152, 358
prerequisites, 338, 339
repeated, 337, 340
tools, 354
types, 339, 340, 343
Function Point counter, 97, 98
Function Point estimation, 409, 410
Function Point Method, 9, 66, 118, 215, 323, 325, 326
price evaluations, 329, 330
primary intent, 347
reuse, 24, 327, 328, 449
risk assessment, 333
test cases, 24
transactions, 343, 347
transactions AUDIO set, 309, 414
transactions complexity, 348, 350
transactions example, 348, 349
UML, 253
unadjusted, 343
usage, 121, 122
use case, 242, 252
functional requirements, 47, 50
functional size, 339
Functional Size Measurement (FSM), 258, 259, 260, 324, 344
Functional User Requirements (FUR), 369
functionality, 13, 323, 340, 341
G
GI Interest Group on Software Metrics, 273, 274, 410
goal conflicts, 32
Goal Question Metric (GQM), 264
GQM, 185, 264, 265, 266, 268
effort, 265
example, 265, 266
phases, 265
tools, 265
GSC’s, 51, 52, 53, 325, 326, 344, 446
GSC’s comparisons, 416
H
Halstead’s metrics, 223, 224
homeostasis, 54
I
IEEE, 120
standard 1045 LOC counting, 120, 136
standards, 180, 181
IFPUG, 38, 40, 269, 274, 324, 325
case studies, 275
Case Study 3 OO environment, 122, 241, 251
CFPS, 96, 268, 274, 275
CFPS exam, 275, 276, 277
COSMIC comparison, 375, 376
counting overview, 454
counting rules, 453
CPM, 274, 276, 277, 325, 350, 358, 453, 459
CSMS, 275
enhancement, 159
maintenance, 159, 160, 161
NESMA comparison, 389, 390, 391, 392
IFPUG FPM, 13, 38, 51, 123, 333, 341, 358, 371, 420
goals, 324
IFPUG rules
adjusted FP’s, 468
application boundary, 456, 457
application count, 480
Complex Processing, 474, 475
counting scope, 456
Data Communications, 470
data function types, 458
DET, 459, 465
DET for EO and EQ, 467
DI, 468, 470
Distributed Data Processing, 470, 471
EI, 461
EI classification, 468
EIF, 458, 460
End-User Efficiency, 473
enhancement, 455, 478, 479
EO, 461, 466
EO and EQ classification, 468
EO and EQ primary intent, 466
EQ, 462, 466
Facilitate Change, 477
files classification, 460
files FP’s, 460
FTR, 465
FTR for EO, 467
FTR for EO and EQ, 467
GSC’s, 468, 469
Heavily Used Configuration, 471, 472
ILF, 458, 459
Installation Ease, 475, 476
maintenance, 480
Multiple Sites, 477
new development, 455, 479
omissions, 463
Online Data Entry, 472, 473
Online Update, 474
Operational Ease, 475, 476
Performance, 471
primary intent of transactions, 462
processing logic, 463
processing logic of transactions, 463
RET, 459
Reusability, 475
Transaction Rate, 472
user view, 456
VAF, 468
VAF, 479
VAFB, 479
implementation process, 87, 89, 93, 106
Indicative Function Point
  Counting, 389
indicators, 183, 184, 185, 209
indirect metrics, 207
inheritance metrics, 245
interfaces, 70, 72, 73, 77, 78
internal metrics, 207
IPMA, 125
ISBSG, 15, 53, 270, 277, 278, 280, 299
  benchmarking analysis, 303
  benchmarking database, 67, 278, 280, 281, 290, 296, 299, 303
  benchmarking database r10, 229, 230, 251, 278, 299, 303, 419
  benchmarking database r5, 415
  benchmarking database r6, 419
  benchmarking database r8, 213, 270, 450, 451
COSMIC projects, 374
data collection form, 303
development methods, 305
development platforms, 305
development techniques, 305
development type defect density r8, 451
development type PDR r8, 451
development type phase proportions r8, 450
duration, 315
duration and effort, 316
duration and size, 316
Early estimate checker, 302, 420
effort, 79, 310, 311
effort ratios, 311, 313
enhancements r10, 166
Estimation Workbook, 434
FP component ratios, 308, 310, 311
FSMM’s, 367
goals, 279
language generations, 304
M&S database, 278, 303
members, 279
mission, 279
new development projects, 307
PDR, 314, 317
PDR by branches, 314
PDR by development platform, 315
PDR by languages, 315
phases, 312
Practical Project Estimation, 278, 440
precision study, 154
programming languages, 304
project size, 305, 306
publications, 278, 280, 330
r8 effort formulae, 311
Reality Checker, 146, 419, 567
regional distribution, 309
Software Metrics Compendium, 166, 226, 230, 278
surveys, 296, 297
team productivity r6, 419
team size, 316, 317
team size formulae, 317
The Benchmark, 278, 280, 303
ISBSG database
application hints, 301, 302
enhancement projects, 166
ISO, 280, 281, 282
ISO Framework for Measurement, 258, 259
ISO SQuaRE, 216
ISO standards, 261, 263, 281
benefits, 258, 281
ISO/IEC
12207 Software life cycle processes, 143, 259
14756 Measurement and rating of performance, 259
14764 Software Maintenance, 160
15504 IT Process assessment, 58, 259
15939 Software measurement process, 259, 262, 266
17024 Conformity Assessment, 125
19761 COSMIC-FFP, 121, 123, 270, 325, 367, 368
20926 IFPUG 4.1, 119, 121, 274, 325
20968 Mark II, 121, 325, 387
24570 NESMA Method, 121
29881 FiSMA 1.1, 121, 123, 325, 377
9126 Quality Attributes, 51, 52, 53, 129, 216, 217, 259, 260, 351
FMS standards, 365, 366, 367 standards, 365
ISO/IEC 9126 Quality Attributes, 582, 583, 584
IT metrics, 90, 98, 100
initiative, 85
organizations, 105
program, 85, 89, 93
IT work unit, 231
ITMPI, 318
IWSM, 105, 273
K
Key Performance Indicators (KPI), 263
KLOC see SLOC, 119
knowledge transfer, 86
L
language level, 123, 133, 134
legacy systems, 28, 135
LOC see SLOC, 119
M
MAIN, 105, 282, 283
maintenance, 28, 88, 159, 161, 164, 166, 167, 353, 392
adaptive, 161, 164
correction, 161, 164
costs, 159
effort after delivery, 169
effort estimation, 159, 167, 170
effort estimation study, 170, 171
effort estimation tool, 170, 171, 172, 173, 174, 176
effort for, 168, 170, 333
FiSMA standard, 160
IEEE standards, 161
measured effort, 168
measurement, 161
metrics, 168, 169
parameters, 167
perfective, 161
preventive, 164
standards, 160
tasks, 165, 170
UKSMA/ISBSG standard, 164
maintenance projects
study, 168
management support, 88, 106, 190, 192, 410
Mark II FP
advantages, 387, 388
complexity factors, 388
CPM, 387
Mark II Method, 38, 121, 123, 130, 152, 269, 387, 388
McCabe, 224, 557
Complexity Design Metric, 123, 224
cyclomatic complexity, 224, 225
measured data, 108
measurement, 9, 15, 22, 23, 26, 29, 183
activities, 24
benefits, 188
efficiency, 266
effort for, 92
implementation, 101, 102, 105, 106
of effort, 17, 23, 24, 25, 26, 28
of size, 9, 10, 12, 16, 22, 109, 112, 117, 119, 126
planning, 353
prerequisites, 190
requirements, 260
tracking, 150
measurement program, 188
measurement scaffolding, 25, 27
measurement system, 267
measures, 182
characteristic factors, 194, 196
examples, 182
metrics, 183, 186, 187
benefits, 187, 189
categories, 184
characteristics, 180
contracting, 189
data, 188
goals, 187, 188
overview, 184
relevance, 179
survey, 198
the Aspirin Metric, 203
the Pizza Metric, 203
viewpoints, 179
metrics database, 95, 96, 186, 190
benefits, 197
implementation, 196, 197
metrics initiative, 185, 186, 193
costs, 188
implementation, 190, 191, 192, 193, 209, 210
implementation strategy, 209, 210
prerequisites, 190, 193, 194
Metrics News, 274, 410
metrics organizations, 268, 274, 282, 283, 285
metrics standards, 61, 257, 258
overview, 259
metrics survey, 198
metrics system, 183, 186
contents, 199
starter set, 200
structure, 198, 199
migration (conversion), 61, 72, 73, 75, 136, 352
MIT, 20
MOOSE, 244
motivation, 102
motivational system, 102, 103
MT22
application knowledge staff, 608
communication mechanisms, 604
contracting procedure, 601
Development environment
skills of staff, 607
documentation methods and
tools, 603
efficiency requirements, 606
functionality requirements, 604
maintainability requirements, 606
motivation, 609
networking skills staff, 608
number of stakeholders, 601
organizational culture, 602
portability requirements, 607
priority setting and control of
changes, 601
productivity factors, 599
release and version policy
factors, 600
reliability requirements, 605
resource and availability for
future needs factors, 600
roll-out methods and tools, 604
source code edition methods
and tools, 602
team atmosphere, 609
testing methods and tools, 603
usability requirements, 605
Multiplicator Method, 147

N
NASA, 232
SEL, 22, 61
NASA Benchmarking
Clearinghouse, 318
NASA/SEL dataset, 297
NESMA, 38, 269
Enhancement Function Points, 163, 393
Enhancement Guide, 163, 392
IFPUG FPM comparison, 389, 390, 391, 392
Test Function Points, 163
NESMA Method, 123, 130
new development project, 71, 339
FP’s, 351
northernSCOPE, 271, 272

O
Object Point Method, 121
Object Points, 243
object-oriented metrics, 241, 242, 243, 245, 250
example, 246, 247
relevant, 243
OLAP Design
DET count, 438, 439
EI count, 438
EIF count, 436
EO count, 438, 439
EQ count, 438, 439
FP component proportions, 436
FP count, 436, 439
FTR count, 439
ILF count, 438
RET count, 438
OO Metrics Suite, 244, 247
OOD metrics, 244
OTOBOS, 198
overestimation, 8, 33, 35
efficiency, 34
factors, 34
productivity, 12
overtime work, 18, 23, 28, 32, 67, 70, 295

P
parametric estimation equations, 143
participation, 105
PDR, 57
business type, 230
conversion to productivity, 229
Finnish database, 229, 230
ISBSG June 2002, 230
Percentage Method, 77, 79, 118, 137, 141, 142, 313
ISBSG database, 143
perfective maintenance, 161
Performance Enhancement Program (PEP), 296
Pi Times Thumb Method, 35, 137, 138
planning metrics, 201, 203
PMBOK, 45, 125
portfolio planning, 410
preventive maintenance, 167
problem complexity, 222
process capability, 232
process improvement, 57
parameters, 201
process maturity models, 262, 264
process metrics, 202, 207
elements, 207
product metrics, 203, 207
examples, 207
productivity, 89, 198, 200, 211, 419
analysis, 418, 419
baseline, 193
comparisons, 227, 228
higher, 231
in Europe, 200
in USA, 200
lower, 230
measurement, 89, 226, 227
measurement example, 227
of enhancements, 89
of projects, 89
PDR conversion, 313
productivity metrics, 200
acceptance, 228, 419
productivity rates, 227
project
cancellation, 70
complexity, 48
controlling, 46, 53
crisis, 111
leader, 4
objectives, 49, 65
plan, 21
planning, 9, 12, 14, 23, 24, 27, 90
portfolios, 568
post mortem, 59, 60, 61, 62, 70
productivity, 81
requirements, 65, 66
risks, 21, 99
size, 47, 71, 79
Project Delivery Rate (PDR), 57
project duration, 235, 236
COCOMO II formulae, 235
ISBSG r10, 235
ISBSG r8 formulae, 235
project management, 56, 60
effort, 57
effort for, 75
manual, 47
operative, 45, 46
strategic, 45, 56
project metrics, 203
report example, 188, 189
Index

project size
  classes, 213
  GUFPI-ISMA/ISBSG, 213
Putnam Formula, 143

Q
quality, 200, 215, 216
quality assurance, 56, 99
  measures, 51
quality attributes
  external, 216
  internal, 216
quality metrics, 198, 200, 207, 216, 217

R
r9 effort, 79
re-design, 449
re-development, 450
  ISBSG FP components r8, 450
  ISBSG r8, 450
re-implementation, 449
Relation Method, 139
requirements creep, 11, 21, 22, 35, 59, 62, 72, 76, 235, 337, 340
resistance, 12, 36, 86, 101, 102
re-test, 450
re-use, 449, 450
rework, 449
risk assessment, 548
risk awareness, 36
ROI, 449
rules of thumb, 414
SEI, 261, 262, 264, 266
size
  functional, 209
  technical, 209

S
SLIM, 144, 146
SLOC, 119, 120, 133, 136, 210, 211, 245
  counting, 136, 211
  paradox, 211
problems, 119, 120
  usage, 120
SMLab, 274
software boundary, 21
Software Maturity Index, 168
software measurement, 90, 91, 92, 94, 100
software measurement database, 274
software metrics, 179
Software Process Improvement (SPI), 261
software product quality, 100
software quality metric, 186
source code metrics, 209
SPEC, 284
SPI, 261
SPICE, 58, 265, 278
SPR, 133
standard software
  implementation, 74
  implementation duration, 74
  implementation effort, 74
Star Scheme
  EI count, 436
  EIF count, 435
  EO count, 436
  EQ count, 436
  FP count, 439
  ILF count, 435, 436
structural complexity, 222, 223
students’ thesis award (DASMA), 273, 297, 446
SWEBOK, 12, 13
system boundary, 341, 342, 355, 443
system complexity, 222, 223
system metrics, 242, 250

T
TDI, 344
technical complexity, 222
The Benchmarking Center, 318
Three Point Method, 137, 138
time accounting, 68, 69, 226
tool expertise, 547
tool support, 6, 7, 95, 148, 193, 196, 265, 356
TPC, 284
tracking metrics, 201, 203
tree metrics, 242, 243, 250
U
UKSMA, 38, 152, 163, 387
UKSMA/ISBSG
   Maintenance standard, 164
UKSMA/ISBSG Defect
   Measurement Manual, 220
underestimation, 8, 32, 33, 154
costs, 35
effort, 33, 34
factors, 33
of size, 154
time, 35
UQAM, 412
use case metrics, 247
Use Case Points, 248, 249, 404
environmental complexity
   factor, 404
technical complexity factor, 404
user comfort, 148
user requirements, 21, 260
user satisfaction, 327
user view, 211, 212, 323, 342, 345
V
VAF, 344
   experiences, 416
W
Web development, 439
   architecture, 441
   architecture diagram, 442
   COSMIC, 439, 441
   DET count, 445
   EI count, 443, 444, 445
   EIF count, 444
elementary processes, 443
   enhancement, 440, 441
   EO count, 444
   EQ count, 444, 445
   estimation, 441
   estimation challenges, 441, 443, 444
   expert estimation, 441, 443
   files count, 443
   FP component proportions, 444
   FP count example, 444
   FP counting, 441, 443, 444
   FTR count, 445
   IFPUG FPM, 440, 441
   ILF count, 444
   measures of usability, 442
   productivity, 440, 441
   productivity ISBSG report, 440
   transactions count, 443
   VAF, 446
Web Object components, 447, 448
Web Objects, 447
WEBMO method, 447, 448
Weights Method, 139, 140, 141
Wolverton Method, 147
Carol Dekkers and Manfred Bundschuh have written an excellent book, which should be added to the collections of all software managers and software metrics workers throughout the world.

Measurement and estimation of software projects has been extremely difficult for both technical and sociological reasons. The technical reasons include scores of poorly-defined and incompatible metrics, gaps or “leakage” from historical data, and the rather sparse collection of accurate benchmarks available to the general software community.

The sociological reasons center around the adversarial relationships between followers of rival metrics and measurement practices. For many years the “lines of code” metrics users have been at odds with the “function point” metrics users. Several other forms of measurement such as earned value, balanced scorecards, and goal-question metrics also have supporters and tend to ignore other forms of metrics.

In recent years the situation has become even more complex. As of 2008 there have been at least 24 function point variants, 5 methods for counting lines of code, and perhaps 15 other forms of measurement such as use case points, story points, object-oriented metrics, and others too numerous to cite.

Dekkers and Bundschuh navigate this tricky area with clarity and objectiveness. All of the major metrics variants are discussed and explained, and their pros and cons are noted.

The book also discusses the organizations that are trying to eliminate competition among the rival metrics camps and achieve some kind of consensus on what needs to be measured and how to go about it. Although there is still antagonism among the various rivals, this new book by Dekkers and Bundschuh is likely to be useful in leading to common goals and mutual understanding of what the various metrics were trying to accomplish.

Prior to the publication of this book, there was no easy way for followers of various metrics to learn about the other possibilities. While there are many books that discuss IFPUG function points, COSMIC function points, goal-question metrics, balanced scorecards, and all the others, this is the first book to try and show all of the major metrics in one volume.

This new book is a worthy companion to older books such as Barry Boehm’s Software Engineering Economics, Steve McConnell’s Software Estimation, Richard Stutzke’s Estimating Software-Intensive Systems, Roger Pressman’s Software Engineering - A Practitioner’s Approach, Steve Kan’s Metrics and Models in Software Engineering, and my own books Estimating Software Costs and Applied Software Measurement. All of these books attempt to show the synergistic relationships among wide-ranging topics, as does this new book by Dekkers and Bundschuh.

Capers Jones
Chief Scientist Emeritus
Software Productivity Research LLC