

# *System Component Tree - Level 3*

## *Study Template*

This template presents the MethodA System Component Tree (S-Tree) at Level 3. The template is meant primarily for use in explaining and understanding the universal (generic) form of the S-Tree. For building documents, the appropriate phase-specific template, with its detailed and appropriate guidance — for analysis, design, testing or maintenance — should generally be used instead. And any specialized information systems and infrastructure systems should take their guidelines from the appropriate kits in the Specializations / Information Systems / Infrastructure Systems volume. But when no more appropriate template is available, experienced users can relatively quickly build general system documentation (a *system document*) from the Work Template that corresponds to this Study Template.

For additional guidelines on using this template, see the Guide in this kit.

**For actual work, use the corresponding Work Template.**

## TABLE OF CONTENTS

<b>Executive Summary</b>	<b>2</b>
<b>0. Administration</b>	<b>3</b>
<b>1. Goals</b>	<b>5</b>
<b>2. Application</b>	<b>9</b>
<b>3. Technology and Infrastructure</b>	<b>26</b>
<b>4. Implementation</b>	<b>35</b>
<b>5. Cost – Resources</b>	<b>42</b>
<b>Appendices</b>	<b>46</b>

# Executive Summary

## 1. Goals

Synopsis of system goals, based on section “1.0 Overview – Highlights” below.

## 2. Application

Synopsis of the application, based on section “2.0 Application Architecture – Highlights” below.

## 3. Technology and Infrastructure

Synopsis of system technology and infrastructure, based on section “3.0 Technology Architecture – Highlights”) below.

## 4. Implementation

Synopsis of the system’s implementation, based on section “4.0 Implementation Highlights”) below.

## 5. Cost and Resources

Synopsis of the system’s costs, including anticipated maintenance, based on section “5.0 Cost ” below.

**A graphic (e.g. PowerPoint) presentation of the executive summary is strongly recommended.**

For more about how to write an executive summary, see the Documentation kit in the Enablers volume.

## 0. Administration

This section assists you in managing and overseeing the system's current life-cycle phase, including the parties involved, the work plan, execution tracking, configuration management and change control with respect to the documentation itself, approvals, quality assurance (QA), etc.

Note the difference between this section and section 4 (Implementation). Whereas this section deals with management of the current phase, section 4 describes the system's overall development plan (on beyond the current phase) and how the system will be operated and maintained. Once the current phase has been completed, section 0 loses its importance. Section 4, in contrast, is one of the system's main deliverables. (Section 0 is "reborn" with the start of the next phase, as a description of that phase.)

In the RFP phase, section 0 becomes the "Administration Section" for the RFP, with a special compulsory format. See the RFP kit in the Basics / Life Cycle volume.

**The precise format and level of detail for this section are left to the discretion of the project/organization management.**

### 0.0 Overview

An abbreviated description of the system's current phase (during which this document is being produced).

#### ☞ 0.1 Parties involved

- Main task-owner (current phase accomplisher)
- Other participants: QA, technical support (infrastructure), data security, consultants in other matters, etc.

### 0.2 Work Plan

- For complex systems/projects: GANTT chart, Pert/CPM
- For average to small systems/projects: a task list including deadline, task owner, cost, and deliverables.
- For all systems/projects: Ongoing execution tracking (vis-à-vis the chart or task list)

### 0.3 Tools and Work Procedures

A list of the tools and work procedures for the current phase.

### 0.4 Configuration Management and Change Control

A table of configuration management (that is, a list of changes) based on the following format and applying to the documentation of the current phase:

<b>Date</b>	<b>Version/Base</b>	<b>Section No.</b>	<b>Description of Change</b>	<b>Approval</b>

## 0.5 Approvals

A table of signatures, based on the following format, if any additional approvals are required beyond the signatures on the front page (see corresponding work template) and on the table above.

<b>Date</b>	<b>Name/Title</b>	<b>Representing (unit)</b>	<b>Comments</b>	<b>Signature</b>

# 1. Goals

## 1.0 Overview – Highlights

**Note:** This section will serve as input for the Executive Summary.

### 1.1 Client / Subject Expert

It is essential to check section 1.1 in its entirety against section 4.1, in order to avoid an excess of teams, a duplication of responsibilities, etc. This section (1.1) deals *solely* with the application expert and the user teams. Other parties involved in the project belong in section 4.1.

#### 1.1.1 Client / Main user

An organizational definition, emphasizing the management level.

#### 1.1.2 Subject expert(s)

A personal and organizational definition, emphasizing the professional (*but decision-making*) level.

#### 1.1.3 User teams

An “outer circle” of application experts, which may be necessary in large projects that involve various types of clients and stakeholders. See the note at the start of section 1.1.

## 1.2 Goals & Objectives

### 1.2.1 General goals

Goals at the organizational level: services that the organization provides, deliverables that the organization markets, positioning in the market, etc. Ways that the system serves the organization’s goals.

### 1.2.2 Specific objectives

Practical short-term objectives: improving service, increasing market share, preventing loss/damage, meeting regulatory and legal requirements, adhering to standards, etc.

### 1.2.99 Future objectives

Goals and objectives for implementation in future versions or delivery units.

☞ *Note: The version(s) or delivery unit(s) primarily dealt with in any MethodA document are considered to be “current,” regardless of whether the document deals with planning, development, or support. A “future” version or delivery unit is one scheduled for later than the current one.*

## 1.3 Problems

### 1.3.0 Summary of current problems

A few sentences, if no more is needed for summarizing the current situation and its problems. If a few sentences are not enough, then use the rest of this section to provide relevant details. If necessary, use section 2.1 as well (as explained there).

### 1.3.1 Problems that the system should/will solve

Problems that the system is designed to solve, sorted by level of damage (including lost income), affected party, criticality, frequency, severity, etc.

### 1.3.2 Problems that the system might/does cause

☞ *In theory, there is a clear difference between system problems as defined here and project risks as defined in section 1.6.2. In practice, however, a reduction in project risks helps reduce the problems that the system might generate when it goes into operation. See 1.6.2.*

#### Typical problems:

- Business-related problems that the system might/does cause in the organization's operation
- Problems for the users of the system
- Operational problems related to the system's own functioning

### 1.3.99 Problems for postponement

- Problems that the system should/will resolve in subsequent phases or delivery units.
- Problems that the system will not solve in the foreseeable future.

☞ *For help with your definitions and descriptions for this section, see the corresponding Extension.*

## 1.4 Organizational/Business Context

### 1.4.1 Organizational goals and strategy

The relationship between the system and the organization's overall strategy (master plan).

Consult the Strategic Planning – Master Plan for Computerization kit in the Overall Management volume.

### 1.4.2 Organizational chart and structure

A detailed description, in words and in a chart, of the part of the organization that will be using the system closely (and thus be the *client*) and will operate differently because of the system.

### 1.4.3 Implications for Organization and Methods

Implications of the system for Organization and Methods and for business processes within the organization.

☞ *Note: This section refers to overall / organizational / business implications. Section 4.7 below refers to practical changes in work procedures and instructions.*

## 1.5 Annual Work Plan

### 1.5.1 Budgetary/business approval

- **For IT departments:** A clear link to the approved annual work plan and to the IT budget.
- **For computer companies and software houses:** A signed/executed contract or work order from the client.

Large projects require multi-year budgeting (authorization) in addition to the project budget for the current year. (For software houses this is a multi-year commitment by the client.)

### 1.5.2 Dependence on other systems

Other information and infrastructure systems, either appearing in the annual work plan or already in development, that the system depends on. Be careful not to overlap this section and sections 2.2.2, 2.22, and 3.33. This section is for *budgetary* dependence, a dependence arising from the organization's overall development plan (work plan).

## 1.6 Feasibility and Cost/Benefit

### 1.6.1 Risks – project feasibility

- Compare the system or project with similar projects inside and outside the organization
- An annual risk survey is mandatory in large and continuing projects.

See the Risk Analysis kit in the Enablers volume.

### 1.6.2 Cost/benefit – business feasibility

- Benefits (quantification of section 1.2 and 1.3)
- Competition, anticipated changes
- Costs (summary of section 5.0 or 5.5)

- Cost/benefit table or formula

For more information and discussion, see the Feasibility and Cost/Benefit kit in the Enablers volume.

## **1.7 Time Frame**

Major milestones and schedules at the “goal” level. The place for the work plan and lower-level dates is section 4, Implementation.

### **1.7.1 Major deliverables**

- The first version or delivery unit to be provided under this project, installed and working
- The next version or delivery unit to be provided under this project, installed and working
- End of the project, completion of the system

### **1.7.2 Drop-dead date**

The critical date after which the project is better abandoned because it will be pointless to install the system.

### **1.7.3 System lifespan**

The expected period of usage for the system before replacement of the system becomes necessary.

The default time frame according to MethodA is five years.

## **1.98 Open Issues**

A list of the open issues from sections 1.1–1.7 that must be resolved before the current version or delivery unit of the system can succeed.

- A comprehensive but succinct list
- A detailed description of each open issue or alternative, or a reference to the appropriate 1.X.98 section above.

## **1.99 Future Goals**

Any important points from sections 1.1–1.7 that will be implemented in future versions or delivery units of the system.

- A comprehensive but succinct list
- A detailed description of each future goal, or a reference to the appropriate 1.X.99 section above.



## 2. Application

The main objective in this section is to ensure that all aspects of the application, which is the system's essence, have been defined. The sequence of the sections (2.1, 2.2, 2.3, etc.) is to an extent arbitrary and is certainly not a prescription for the sequence in which components should be defined and built. Each project will find its own starting point and then move on to other components as appropriate, shifting among them systematically, gradually, and sometimes iteratively. Section 2.20, Cross-references, is very important. In addition to functioning as a summary and as a QA tool, it plays a constant role in maintaining balance and coordination among the various components throughout the system analysis process. There are formal methodologies along these lines, called *cross-reference analysis* or *requirements tracking*. Such methodologies are useful but very difficult to use without computerized tools.

Despite the somewhat arbitrary arrangement of this section, a number of clear subtopics are discernable within an application:

- **System scope** and (external and internal) **system boundaries**. *Sections 2.1–2.3.*
- **User interface** (User experience). *Section 2.4.*
- **Business rules** that govern system behavior. These rules are also known as the dynamic model or the algorithmic part of the system. *Sections 2.5–2.10.*  
The user interface and business rules, sections 2.4–2.10, together define how the system works and is worked with.
- **Information** (data) managed by the system, and supporting data used by the system. This aspect is also known as the *static model* or the *database*. It includes section 2.10 — a “two-headed animal,” as explained later in its own context — even though the dynamic model includes it too. *Sections 2.10–2.16.*
- **“Orthogonal” parameters** that cut across the previous subgroups and describe general properties of the application such as interfaces, data security, and performance.

The subtopics — and in particular the concepts of interface, business rules, and data — dovetail with most of the accepted systems-analysis methodologies, so that the Application section of this document can connect logically and physically with CASE tools and with other advanced software-engineering tools and methodologies.

☞ *For further details and explanations, see section 2 (Application) of the User's Guide.*

### 2.0 Application Architecture – Highlights

A chart or brief verbal explanation of the application's overall architecture.

**Note:** This section is input for the Executive Summary.

#### 2.1 Main Attributes

##### 2.1.1 Current situation

Any of three possibilities, from the easy to the weighty:

1. A reference to section 1.3 (Problems) if the level of documentation there — either in 1.3.0 or in 1.3 as a whole — is sufficient.
2. A brief survey of the current situation, if section 1.3 is insufficient
3. A thorough and wide-ranging survey in the form of an attachment named Attachment 2.1.1, “The Current Situation.”

For more information, particularly regarding the last two possibilities, see the Current Situation kit in the Enablers volume. See also the System Analysis kit.

### **2.1.2 Type of system**

State the type of system in terms of the following::

- Online transaction processing (OLTP), batch, realtime, internet, or a combination
- Centralized, distributed, or multi-tier
- New, re-engineered, adapted, added as a new delivery unit for an existing system, etc.

### **2.1.3 Constraints**

Legal and administrative constraints especially:

- Legal/Regulatory requirements (local, international)
- Company standards, policies, etc.
- Ethics and professional norms
- Industry requirements

Other constraints having to do with the technology, the application, personnel, costs, time frame, etc. should be covered *not* here but in the appropriate sections.

### **2.1.4 Glossary**

Definitions of a few particularly important terms used in the system. Do not repeat terms that are explained in the natural course of other sections (e.g., 2.10, 2.11, and 2.13).

## **2.2 Users & Interfacing Systems**

### **2.2.0 System Scope**

An overall description in the form of a chart, such as a context diagram, with a brief legend.

### 2.2.1 Users

- Inside the organization, according to organizational structure
- Outside the organization

### 2.2.2 Interfacing systems

- Inside the organization (information systems and infrastructure systems alike)
- Outside the organization (information systems and infrastructure systems alike)

**Note:** The purpose of mentioning external systems here is to fully establish the scope of the system. The place that details the system's interfaces with other systems, both internal and external, is section 2.22.

## 2.3 Internal Subsystems

This section is of central importance, particularly when the project may be subdivided into subprojects, delivery units, or work by parallel teams. The system analyst may already need to provide a preliminary view of this section in the early stages of Analysis (generalized analysis).

### 2.3.0 System layout

- General explanation in a diagram, e.g. DFD Level 0 with a brief legend
- Division into major subsystems
  1. Division into versions / delivery units / subprojects
  2. Ranking by complexity, precedence, and priority

### 2.3.1 Subsystem/Delivery Unit 1

Note that further breakdown and detailing of the subsystem will appear in sections 2.5 and 2.6.

In MethodA the first subsystem and if necessary the second (that is, 2.3.1 and 2.3.2) are customarily the system's internal management and control, along with cross-system mechanisms and services such as error handling.

- General explanation in a diagram, such as DFD Level 1 with a brief legend)
- Major functions and processes

### 2.3.2 Subsystem/Delivery Unit 2

Remember, further breakdown and detailing of the subsystem will appear in sections 2.5 and 2.6.

- General explanation in a diagram, such as DFD Level 1 with a brief legend)
- Major functions and processes

### 2.3.N Subsystem/Delivery Unit N

## 2.4 User Experience

### 2.4.0 Human-engineering guidelines

- A succinct description of the rules and principles for the system's human-machine interface (HMI)
- A reference to the object library and to the standard UI tools for building the user interface.

☞ *For further helpful HMI-related information, see the corresponding Extension.*

For detailed, comprehensive computer interface information, see the HMI kit in the Enablers volume. See also the UI standards in the Standardization and Standards kit there.

### 2.4.1 Menu screens – Site Map

#### Main screen – main menu/home page

- Structure, functions, content/data, selection/links
- Layout, or reference to a prototype / concept model

#### Principal secondary screens (submenus)

- Structure, functions, content/data, selection/links
- Layout, or reference to a prototype / concept model

#### Screen hierarchy – site map

### 2.4.2 Work screens

For each work screen or work window:

- Structure, functions, content/data, selection/links
- Layout, or reference to a prototype/concept model

## 2.5 Processes

### Introductory note:

Sections 2.3, 2.5, and 2.6, taken together, describe the procedural (algorithmic) aspect of the system at various levels of decomposition (i.e. the system's dynamic model). The balance between them is important.

Section 2.3 presents the highest level of system decomposition (subsystems), and section 2.6 presents the lowest level (the system's most basic operations). Section 2.5 therefore encompasses all the intermediate levels: processes and subprocesses. In practice, the balance among these three sections can be achieved in a number of ways, (from the simple to the complex) as follows:

- a. For very simple systems: Sections 2.3 and 2.5 serve no practical purpose. Section 2.4- User Interface, on the other hand (and section 2.4.2 in particular), connects directly to section 2.6 (Transactions). These two sections essentially cover the dynamic model (the algorithmic part) of the system. See the Small Systems kit, in the Project Management volume.
- b. For simple or medium systems with a clear-cut division into subsystems: Sections 2.3 and 2.6 are the center of gravity. There is a direct connection between them (as well as between them and 2.4). Section 2.5 is optional, consisting of a reference to the others if it exists at all.
- c. For simple or medium systems that are not divided into subsystems: Sections 2.5 and 2.6 are the center of gravity. Section 2.3 provides only a succinct general explanation and immediately references section 2.5. In fact it is possible to understand the system directly from section 2.5, which relates to section 2.4 on the one side and to 2.6 on the other.
- d. For medium to complex systems: All three sections (2.3, 2.5, and 2.6) are important, as explained in the sections themselves.
- e. For complex systems that are developed in a number of versions or delivery units: Section 2.3 is the center of gravity (in the main system document) and refers to appendices 2.3.1, 2.3.2, etc., or to the subsystem documents (for the versions or delivery units) with their individual Application sections. Sections 2.5 and 2.6, like all the other Application sections of the main system document, describe only the components that are common to all the subsystems. For these types of systems see the MDU kit (Multiple Delivery Units) and the LSS kit (Large Systems and Subsystems) in the Project Management volume.

Whenever including a section 2.5, connect it clearly to section 2.3: Section 2.5 begins where 2.3 ends. The "roots" of component 2.5 are the "leaves" of 2.3 (in the subcomponents/subsystems of sections 2.3.1, 2.3.2, etc.). Process 2.5.X is the direct continuation of subsystem 2.3.X. Thus section 2.5.1 defines all processes that support cross-system mechanisms (see section 2.3.1).

### **2.5.0 General Index**

A list or table of all the system's processes. Each line of the list or table should include the following properties:

- Process name and code number (linking to section 2.3)
- Process type (e.g. online, batch)

- Level of decomposition (final, decomposable)

### **2.5.1 <name of first process>**

A detailed description of the first process, including:

- A textual description
- A chart (such as DFD level 2) that will later be decomposed

See the complete and detailed explanation of this section's structure in the template corresponding to the current phase: Analysis, Design, Testing, or Maintenance.

#### **2.5.1.1 <name of first subprocess>**

A detailed description of the first subprocess, including:

- A textual description
- A chart (such as DFD level 3) that is not subject to further decomposition
- A reference to the relevant transactions that make up the subprocess

For a detailed description of subprocesses, see the corresponding section in the Analysis document template.

### **2.5.2 <name of second process>**

A description of the second process, resembling section 2.5.1 above, at whatever various levels of decomposition are appropriate for the complexity of the process.

**And so on throughout the process tree...**

#### **Notes in conclusion:**

1. A process can be decomposed on several levels. It can end at level 3, as shown in 2.5.1.1; it can continue to a lower level, such as 2.5.1.1.1; or it can stop at a higher level, such as 2.5.1. It all depends on the complexity and intricacy of the process. In any event, the lowest level — the “leaf” on the process tree — will link to sections 2.4 and 2.11, either through section 2.6 or directly.
2. Decomposition is not necessarily symmetrical across the process tree. One branch may end at level 2 (e.g., 2.5.1) while a different branch ends at level 4 (e.g., 2.5.4.1.1).
- 3.

## **2.6 Transactions**

See the remarks at the beginning and end of section 2.5. A transaction is usually a further decomposition of a process. In some systems, the level of detail and decomposition in section 2.5 may make a lengthy section 2.6, or even any section 2.6 at all, unnecessary.

### 2.6.0 Index of transactions

A list or table of all the system's transactions. Each line of the list or table should include the following properties:

- Transaction name and code number
- Transaction name type (input, output, update, etc.)
- Grouping (by processes / screens / files)

### 2.6.X <transaction name>

A detailed description of the transaction, including:

- A flowchart
- Name, internal process, input, output
- A cross-reference to processes, screens, files, etc.

See the complete and detailed explanation of the section's structure in the template corresponding to the current phase: Analysis, Design, Testing, or Maintenance.

## 2.7 Modules (Programs)

This section is first used in the Design and Build phase. Its exact structure depends largely on the choice of development and maintenance tools, including configuration-management and version-control tools. The suggested structure is a basis for adaptation, by project management, to the relevant development environment and methodology. The section's introduction (replacing this text) should explain the chosen structure. In all cases, decomposition into modules will maintain the connection either with transactions (section 2.6) or — when section 2.6 is not necessary — directly with processes (section 2.5).

This section may be written with either a centralized or a distributed approach.

- The centralized approach documents all system modules here, including modules that implement screens and interfaces.
- The distributed approach uses this section, in direct continuation from sections 2.5 and 2.6, for modules that implement processes and transactions. But modules that implement screens and interfaces are documented in their own context (sections 2.4 and 2.22).

**Project management will decide which approach to use.**

☞ *For further information and for the suggested Programs document structure, see the corresponding Extension.*

Another source of assistance is the study and work templates of the Programs document, located in the Design and Build kit, inside the Life Cycle volume.

### 2.7.1 Source modules

A list or table of all the system's modules. For each module, list the following properties:

- Module name and code number
- A reference to the Programs document (see Appendix 2.7.1: List of Source Modules, below)
- A reference to a library

### 2.7.2 Executable modules

A list or table of all the system's modules. For each module, list the following properties:

- Module name and code number
- A reference to a library
- Link to the source programs

## 2.8 Control Procedures

This section is for system control procedures (also known, depending on the specific operating system, as scripts, operating system commands, etc.). The documentation here, as in section 2.7, should be based on references to the relevant libraries.

This section provides a static description of the procedure library. Like section 2.7, it is used especially in and after the Design and Build phase. How to run the procedures is covered in sections 2.4 (User Interface) and 4.4 (System Operation).

## 2.9 Subroutines (Common Objects)

### 2.9.1 Local subroutines

A list or table of all local (system) subroutines — subroutines written as part of system development and belonging exclusively to the system. In each line of the list or table, include the following properties:

- Subroutine name and code number
- A brief description of the subroutine
- A reference to a library

#### 2.9.1.X <subroutine name>

A detailed description of the subroutine, including:

- A flowchart
- A “where-used” link to other components that use the subroutine
- A source library and an executable library



For a detailed description, see the corresponding section in the Analysis kit template.

### **2.9.2 Global subroutines**

A list or table of all global (organizational) subroutines — subroutines shared with other systems within the organization. In each line of the list or table, include the following properties:

- Subroutine name and code number
- A brief description of the subroutine
- A reference to a central library

### **2.9.3 External subroutines**

A list or table of external (sectoral, cross-organizational) subroutines — often supplied by a third party. For each subroutine (in a table, each line), include the following properties:

- Subroutine name and code number
- A brief description of the subroutine
- A reference to a central library

## **2.10 Coding Tables**

Note the difference between this section and section 2.11. This section refers to “central tables” that contain code numbers and parameters intended for system parametrization. Section 2.11 refers to “tables” in the relational-database sense, namely data files. Private (module-level) coding tables that belong exclusively to the system and are modified during operation are treated as files and belong in section 2.11. For further explanation, see the Analysis template in the System Analysis kit.

Rely as much as possible on standard definitions at all three levels of coding tables — local, global, and external. See the Data Dictionary and Tables kit in the Enablers volume.

### **2.10.1 Local coding tables**

A list or table of all local (system) coding tables — those written as part of system development and belonging exclusively to the system. For each coding table, specify the following properties:

- Table name and code number
- A reference to a library

#### **2.10.1.X <coding table name>**

A detailed description of the coding table:

- Internal structure

- A “where-used” link to other components (if any) that use the table

For a complete and detailed description of the component’s structure, see the template corresponding to the current phase: Analysis, Design, Testing, or Maintenance.

### **2.10.2 Global coding tables**

A list or table of global (organizational) coding tables — those that are shared with other systems in the organization. For each coding table, specify the following properties:

- Table name and code number
- A brief description of the coding table
- A reference to a central library

### **2.10.3 External coding tables**

A list or table of external (sectoral, inter-organizational) coding tables — those that are shared by other organizations and often supplied by a third party. For each coding table, specify the following properties:

- Table name and code number
- A brief description of the coding table
- A reference to a central library

## **2.11 Logical Data Structure**

### **2.11.0 Overall – Data modeling**

1. Schemas:
  - Global schema: ERD, DSD, or Class Diagram
  - Secondary schemas (partial views)
2. Index/table of logical file. For each logical file, include the following properties:
  - File name and code number
  - Key field
  - General description

### **2.11.X <name of logical file>**

A detailed description of the logical file:

- Internal structure: key field, other fields
- Volume and volatility

- A “where-used” link to the other components that use this file

For a complete and detailed description of the component’s structure, see the template corresponding to the current phase: Analysis, Design, Testing, or Maintenance.

## **2.12 Physical Data Structure - Database**

### **2.12.0 Overall model**

- Global schema: Physical ERD or Storage Diagram
- Secondary schemas (partial views)
- Index of physical files — table, main properties

### **2.12.X <name of physical file>**

A detailed description of the file:

- Internal structure: key field, other fields
- Volume and volatility
- Frequency of access
- Links to logical files

See the complete and detailed description of the component’s structure in the template corresponding to the current phase: Analysis, Design, Testing, or Maintenance.

## **2.13 Data Items**

Data items (fields) are found in screens, reports, and data files, and internal and external interfaces. Data items exist at four different levels: private (module), local (system), global (organizational) and external (sectoral, inter-organizational). This section deals with the last three levels; it does not address the private (temporary) data items that are used by a specific module and are not found in any interface, incoming message, or outgoing message.

Rely on standard definitions as much as possible at all three levels — local (between modules within the system but not outside the system), global (between systems within the organization but not outside the organization); and external (between organizations). See the Data Dictionary and Tables kit in the Enablers volume.

### **2.13.0 General index**

A list or index of all the data items and fields that are not private (that is, from the local level up). For each field, provide the following properties:

- Field name and code number
- Type: local (system), global, external
- Basic structure: type (numeric, alphanumeric, etc.), size

- Link to data dictionary

This method, which assembles all the data items in one table or index while distinguishing between their types, is suitable for small and medium systems where no more than fifty data items are of interest. A different system, more suitable for medium systems and larger, is presented below.

### **2.13.1 Local fields**

A list or index of all the local fields (those that appear only in internal interfaces between modules, inside the system and under its control). For each field, provide the following properties:

- Field name and code number
- Basic structure: type (numeric, alphanumeric, etc.), size
- Reference to a data dictionary

### **2.13.2 Global fields**

A list or index of all the global fields (those shared with other systems inside the organization). For each field, provide the following properties:

- Name and code number
- Basic structure: type (numeric, alphanumeric, etc.), size
- Reference to a data dictionary

### **2.13.3 External fields**

A list or index of all the external fields (those shared with other organizations). For each field, provide the following properties:

- Name and code number
- Basic structure: type (numeric, alphanumeric, etc.), size
- Reference to a data dictionary

For more details of the fields that describe the data items of various types above, see the template corresponding to the current phase: Analysis, Design, Testing, or Maintenance.

## **2.15 Reports (and Queries)**

### **2.15.0 Index of reports**

A list or table of all the system reports (or of the queries that generate them). In each line of the list or table, include the following properties:

- Report name and code number
- Primary recipient(s)

- Category (report group)

**Further detailing of reports may be by either of two methods:**

- By groups
- Straight down to individual reports, without grouping

### **2.15.X <name of group of reports>**

#### **2.15.X.0 Index**

#### **2.15.X.N <name of report in group X>**

### **2.15.X <name of report>**

A detailed description of the report, including:

- Report heading (Title)
- Detailed format (columns, summaries, breaks)

For how to describe a report's structure fully, see the section corresponding to this one in the Analysis kit template.

## **2.16 Input (Forms)**

**Note:** This section is unnecessary in most current IT systems. "Forms" are now system screens and belong in subsection 2.4.2. Nevertheless, this section may be used for one of the following reasons:

- The forms reside outside the system, i.e., they are filled in manually and then, at a later stage, entered into the system.
- It is essential to present all system inputs together (like the reports, above).

If the decision is made to use this section, then it should resemble section 2.15 (Reports) in level of detail.

### **2.16.0 Index of inputs**

A list or table of all the system inputs. In each line of the list or table, include the following properties:

- Input name and code number
- Primary source(s)
- Category (input group)

**Further detailing of inputs may be by either of two methods:**

- By groups
- Straight down to individual inputs, without grouping

## **2.16.X <name of group of forms>**

### **2.16.X.0 Index**

### **2.16.X.N <name of form in group X>**

## **2.16.X <name of form>**

A detailed description of the form, including:

- Form's heading (Title)
- Detailed format (columns, summaries, breaks)

For how to describe a form's structure fully, see the section corresponding to this one in the Analysis kit template.

## **2.19 Information Security & Privacy**

### **2.19.0 Overall – Highlights**

- An overall definition of the system's information sensitivity and of its security level or business confidentiality level
- The overall security model
- Conformance to standards and to the organization's security procedures

### **2.19.1 Exposure and risks**

- Identification of risk/exposure
- Probability, severity, and level of damage

☞ *For further information, see the corresponding Extension.*

### **2.19.2 Security measures**

Details of the security measures (or requirements) in the system, unless it is sufficient to mention a standard or an organizational infrastructure.

Suggested security measures are:

- User identification: passwords, and authorizations
- Compartmentalization of system access
- System entry and exit log
- File-access management and logging
- Other security measures

☞ *For further information, see the corresponding Extension.*

### **2.19.3 Security administration**

System security administration tools and measures, such as:

- User administration: assignment of passwords and authorizations
- System entry and exit log
- Compartmentalization definitions
- Security reports, e.g. attempts at unauthorized access

### **2.20 Cross-references**

- Users / Subsystems
- Users / Processes
- Processes / Logical files
- Logical files / Physical files
- Files / Data items
- Subsystems / Interfaces
- Other relevant cross-references

### **2.21 Workload, Performance, and Capacity**

Specify system workload, performance and capacity. Typical parameters to include:

- File size and volume
- Volatility of data
- Number of workstations and users
  1. Aggregate
  2. Concurrent
  3. Identified (users), unidentified (surfers)
- System Throughput:
  - Transaction rate and response time
  - Batch processing (Jobs)
  - Turnaround times
  - Number of messages (per time unit)
- Network workload and performance
- Other metrics

☞ *For further information, see the corresponding Extension.*

## 2.22 Interfaces & Links

### 2.22.0 Index of interfaces and links

A list or table of all system interfaces and links. In each line of the list or table, include the following properties:

- Interface code number
- Interface classification code
- Grouping (link to processes / screens, or to files)

### 2.22.X <name of interface>

A detailed description of the interface, including:

- Name, logical code, and physical code
- Type of interface: data transfer, messaging, service request / approval, link to another application
- Responsibility and source/destination: sender or receiver, push or pull
- Operation: batch, online, queuing (store & forward)
- Service/action requested
- Description of transferred data
- Internal protocol/control record
- Protocol for interface implementation
- Standards used
- Volume and frequency

For further information and the detailed description of an interface, see the Message and Data Transfer kit in the Enablers volume. See also the section corresponding to this one in the Analysis kit template.

For relevant standards, see the Standards and Standardization kit in the Enablers volume.

## 2.23 Special Requirements

The above Application sections may not cover all requirements and system features. This section is for the rest, such as —

- **Flexibility / Resilience:** Tolerance for the addition of changes and for expansion
- **Portability:** Cross-platform data and code transferability
- **Languages:** Local language support (multilingualism)



For relevant standards, see the Standards and Standardization kit in the Enablers volume.

- Other requirements

## **2.98 Open Issues and Alternatives**

A compilation of the main open Application issues that must be resolved for the current version or delivery unit of the system. The compilation should include:

- A comprehensive but concise list
- A detailed description of open issues/alternatives

See the Alternatives Analysis kit in the Enablers volume.

## **2.99 Future Requirements**

Any important information from sections 2.1–2.23 that should be implemented in future versions or delivery units of the system, but not in the forthcoming one.

- A comprehensive but concise list
- A detailed description of future requirements if necessary; otherwise a link to the relevant section 2.X.99

## 3. Technology and Infrastructure

This section is divided into three parts, as follows:

- a. Sections 3.1–3.19 (not all numbers are in use): A description of the servers (main computers) that support the system (hardware, software, and tools), including front-end servers, application servers, web servers, back-end servers, mainframes etc. that are part of the system, etc.
- b. Sections 3.20–3.29 (not all numbers are in use): A description of the various client computers that can work with the system.
- c. Sections 3.30–3.32 (not all numbers are in use): A description of the communications network(s).

If sections 3.1–3.19 (“a” above) refer to various types of servers (main computers), use one of the following options:

- Division by type of resource (recommended):
  - 3.1 Hardware
  - 3.2 Data Storage
  - 3.10 Operating System
  - etc.

If there is more than one server, *subdivide* each resource according to type of server:

- 3.1.1 Server-A Hardware
- 3.1.2 Server-B Hardware
- 3.10.1 Server-A Operating System
- 3.11.2 Server-B Database
- etc.

This option is the one used below.

- Division by computer/server configuration (generic):
  - 3.1 Computer A
  - 3.2 Computer B
  - 3.3 Computer C
  - etc.

Each type of computer/server will be *subdivided* according to type of resource:

- 3.1.1 Server-A Hardware
- 3.1.2 Server-A Data Storage
- 3.2.10 Server-B Operating System

- etc.

For sections 3.20–3.29 (the client computers), MethodA recommends the following division:

- 3.20 Hardware
- 3.21 Infrastructure Software
- etc.

In the special case of more than one type of client computer, client computers may be divided by generic configuration, as in the second option for central computers.

**In relatively simple cases, section 3 should cover not only generic/technical specifications but also exact configurations and quantities. In more complex cases, section 3 should focus on generic/technical specifications and section 4.9 should specify configurations, quantities, and their locations. See section 4.9 (Configurations).**

### 3.0 Technology Architecture – Highlights

A diagram of the system’s architecture and central technological components of the system: computers, layout, and network. Support the diagram with a textual description detailing the main points of the technology in use.

**Note:** The content and complexity of the diagram may affect the choice of how other technology components will be documented (see options 1 and 2, above).

This section’s content is input for the Executive Summary.

#### 3.1 Main Hardware

- Definitions of all server hardware that supports the system, including central computers, back-end computers, etc., and not including computers dedicated to clients or to the network.
- Model
- Memory, including cache
- Auxiliary devices
- Modularity and upgrade possibilities

For relevant standards, including safety standards, see the Standards and Standardization kit in the Enablers volume.

#### 3.2 Data Storage

Definition of the system’s central data storage.

- Online storage – hard disks (RAID)
- Removable storage (diskettes, tape cartridges, CD burner)
- Backup and transfer facilities
- Archival

For relevant standards, including safety standards, see the Standards and Standardization kit in the Enablers volume.

### 3.3 Peripherals

- Monitors
- Printers
- Plotters
- Etc.

This section covers peripherals for central hardware. Peripherals for client stations belong in sections 3.20–3.29.

### 3.4 Special Equipment

Any special equipment connected to the central computers (not to a client station), including:

- OCR
- Scanners
- Bar-code scanners
- Etc.

### 3.5 Consumables

Supplies such as cartridges and tapes, diskettes, CDs, paper, etc.

### 3.9 Passive infrastructure

☞ *For details, see the corresponding Extension.*

#### 3.9.1 Main site

- Structure: Computer room, telecommunications center
- Power supply and air conditioning
- Cables and communications infrastructure
- UPS
- Backup devices

### 3.9.2 Backup site

Parallel to section 3.9.1.

This section is for use when the “Backup devices” item in 3.9.1 does not suffice and when the system demands a complete backup site. But remember that generally, when such a comprehensive infrastructure is needed the solution resides at the overall organizational level and not at the level of the individual system. Accordingly, this section details only the additional demands that the system makes on the organization’s central backup.

### 3.9.3 Safety requirements

All the safety requirements, including safety standards for the system, may be grouped here instead of being distributed on a per-section basis.

## 3.10 Operating System

A list of all the operating systems, by type of central computer and server.

- Name and version
- Manufacturer and vendor (dealer)
- Alternative operating system
- Concise description (optional)

☞ *For details and further information, see the corresponding Extension.*

## 3.11 Database Management System (DBMS)

All the types of databases and the database management systems for various types of information, by type of central computer and server.

### 1. General

- Name and version
- Manufacturer and vendor (distributor, dealer)
- Alternative DBMS (name, version, manufacturer)
- A concise description (optional)

### 2. Data (Content) management features

### 3. Metadata management (Data Dictionary features)

☞ *For details and further information, see the corresponding Extension.*

☞ *Also see the Extension for the Data Dictionary.*

## 3.13 Develop & Maint. Tools

Tools such as —

- Editors and Publishers
- Graphic tools: Visual design of the system (user interface)
- Programming languages, including script/code generators
- Report and help-screen generators
- Preparation and management of testing
- Configuration-management software
- Library and workspace management
- System analysis and design
- Auxiliary software

For each tool, specify name, version, manufacturer, vendor, and (in brief) essential features.

☞ *For details and further information, see the corresponding Extension.*

### 3.14 Third-party Software

Details of all third-party (off-the-shelf) infrastructure software and application software for central computers and servers, except operating systems, databases, and development, operating, and production tools (which were covered in sections 3.10–3.13).

Definitions of these tools should make mention of the tools that the project or the organization already has.

#### 3.14.1 Utility software

- Search engines
- Messaging and notification management
- Data security, e.g., data encryption between server and user  
This should be consistent with the requirements in section 2.19.2. Note which requirements are met by the utility software.
- Data-compression software
- Online help management
- Connectivity between servers

#### 3.14.2 Application Software

Software including, for example:

- GL (accounting)
- Statistics

- Graphics

## 3.15 Operation Tools

### 3.15.1 Production for operators and for production managers

- System startup/shutdown
- System operation tracking, including notification of exceptional events
- Backup and restore
- Recovery
- Running periodic procedures: statistical reports and “clean-up”

### 3.15.2 Management and control tools for system managers

- Identification of online users
- System-security monitoring (ISS)
- Urgent messaging
- System performance metrics and monitoring tools
- Statistical analysis and usage-tracking software
- System refresh

Definitions of these tools should make mention of the tools that the project or the organization already has.

☞ *For details and further information, see the corresponding Extension.*

## 3.20 Client Hardware

- Computer (minimum required processor)
- Connectivity and expansion potential
- Data storage
- Monitor requirements: size, resolution
- Telecommunication components:
  1. Modem, network interface card, bandwidth
  2. LAN connection
- Physical environment (if not a standard office environment)
- Special equipment
- Peripherals
- Backup devices

### 3.21 Client Infrastructure

- Operating system, including —
  - Drivers
  - Utility software
  - Emulator
- Database
- Browser
- Plug-ins, any software complementing the server software

**Note:** Check sections 2.5.1, 2.9, 2.19, and 3.10–3.15 for anything that requires complementary software at the client computer.

### 3.22 Client Application

Software packages for direct use by the end-user.

- Office productivity suites/programs
- Local report generator
- Discussion groups / Forums
- Newsletters
- Chat
- Statistical analysis and presentation software

**Note:** Check sections 2.5.2, 2.9, 2.19, and 3.10–3.15 for anything that requires matching software at the client computer.

☞ *For details and further information, see the corresponding Extension.*

### 3.30 Private/Local Area Network

The connection to the organization's LAN, including the intranet and connectivity to the extranet.

☞ *For details and further information, see the corresponding Extension.*

For relevant standards and standardization considerations, see the Standards and Standardization kit in the Enablers volume.

### 3.31 Private / Wide Area Network

- Connection to the organization's WAN, including connectivity to a public network/Internet
- VPN
- Extranet



**Note:** In many cases, section 3.30 is sufficient and this section is unnecessary because the organization's internal network is an LAN integrated with a WAN or MAN (metropolitan area network).

For relevant standards and standardization considerations, see the Standards and Standardization kit in the Enablers volume.

☞ *For details and further information, see the corresponding Extension.*

### 3.32 Public Network

- Connection to the public Internet
  1. Link to an ISP: dial-up (modem), ISDN/ADSL, dedicated line
  2. IP addresses
  3. Network interface card
- Link to knowledge bases
- Link to other public network
  - Depending on economic sector — international associations, consortium, etc.

For relevant standards and standardization considerations, see the Standards and Standardization kit in the Enablers volume.

### 3.33 Interfacing Technologies

Any associated technology with possible implications for the system.

- Back-end and other computers that will work closely with the system but are not included in components 3.1–3.19
- Associated operating systems and DBMS
- Associated networks

### 3.98 Open Issues and Alternatives

A compilation of the main open Technology issues that must be resolved for the current version or delivery unit of the system:

- A comprehensive but concise list
- A detailed description of each open issue/alternative

☞ *Use the Alternatives Analysis kit in the Enablers volume.*

### 3.99 Future Technologies

Support for future technologies:

- Hardware and central equipment
- Operating systems and databases

- Infrastructure software
- Multimedia
- Advanced browser versions
- Development tools
- Telecommunications, integration with telephony
- Support for advanced peripheral equipment (of clients), for example:
  1. Cellular devices
  2. PDAs

## 4. Implementation

### 4.0 Implementation Highlights

A summary of the implementation methodology and of the work plan for development and operation.

**Note:** This section will serve as input for the Executive Summary.

### 4.1 Parties Involved

Include the names, titles, and contact information of the following offices and individuals.

#### 4.1.1 Management

- Project steering committee
- Project administration

#### 4.1.2 Professional teams – Development teams

- Analysis and Specification team
- Design and Build team
- User Interface (prototype/model) team
- Testing team
- Etc.

**Note:** Application experts and user teams involved in the project are defined in section 1.1.

#### 4.1.3 Technical Support

- Infrastructure, operating system, databases, communications
- Quality assurance
- Human engineering and user interface
- Information security
- Procurement, budget, and cost-tracking
- Help desk, call center

#### 4.1.4 Vendors and other outsiders

- Computer companies and software houses
- Consultants and specialists

☞ *For further information that may be useful in your definitions and documentation here, see the corresponding Extension.*

## 4.2 Work Plan

### 4.2.0 Development Methodology

The chosen methodology for development and project management: sequential, ID/RAD, SDU, ADU, MDU, etc.

For explanation, see the appropriate kits in the Management / Project Management volume.

### 4.2.1 Overall Development Plan

- Main milestones and deliverables
- General work plan: High-level GANTT chart

### 4.2.2 Detailed Plan

- Detailed GANTT chart(s) and detailed schedule
  - Link to the project teams' detailed task lists and work plans
- ☞ *Use the Project Management Techniques kit and other kits in the Management volume.*

## 4.3 Next Phase

Whatever is to be done “immediately,” during the next quarter:

- Main objective/deliverable – content
- Action team (recommended)
- Work plan: Task list

☞ *Use the Time Management kit in the Enablers volume.*

## 4.4 Ongoing Operation

Note the difference between this section and the System Operation Manual. In early development phases (Inception and Analysis), this section (4.4) contains system operation requirements. The requirements are gradually transferred to the System Operation Manual and this section evolves into a reference to that manual. The System Operation Manual is the full and binding documentation of system operation. (In many systems the System Operation Manual is a “live” electronic document integrated with the system monitoring / production tool.)

The system operation requirements (and later the System Operation Manual) will include these topics among others:

- System operation: organization/individuals and responsibilities
- Support and supervision: organization/individuals and responsibilities for technical support, client, computer-center management

- Times of operation (shifts)
- Resources
- System startup/shutdown
- Taking the system off line for preventive maintenance, reorganization, etc.
- Error control
- Backup and restoration
- Emergency and underequipped operation
- Security and safety

☞ *For details and further information, see the corresponding Extension.*

## 4.5 Documentation

**Note:** This section is nothing but an index and references. Its purpose is to briefly summarize the status of the system documentation and project library and to serve as a quick reference to their locations. The top-level documentation is the system document (Analysis, Design, Maintenance, etc.) — which is this document! — and it includes references to appendices and to other specialized documents.

### 4.5.1 Operational documentation

Document Title	MethodA Section number	Reference to Documentation	General Status
System Document (Maintenance)	---	Library and document name	
User Manual	4.7.4	Library and document name	
Operation Manual	4.4	Library and document name	
Service, operation, & maintenance contracts	4.6	Library and document name	
Manufacturer's Documentation	---	Library and document name	

### 4.5.2 Development process documentation

Document Title	Kit/Template	Doc. Reference	General Status
Inception Document	System Inception	Library/file name	
(Generalized or Full) Analysis Document	System Analysis	Library/file name	
RFP	RFP	Library/file name	
PEM, Vendor	RFP	Library/file name	

Document Title	Kit/Template	Doc. Reference	General Status
proposals			
System Development Contract	Contracts	Library/file name	
Design Document (Detailed Analysis)	Design & Build	Library/file name	
Testing Document	Testing		
Test Results	Testing		
Correspondence			
Minutes and Reviews		Appendices of the documents	

## 4.6 Service and Maintenance

☞ *For details and further information, see the corresponding Extension.*

See also section 4.6 in the Maintenance Document – System Document.

### 4.6.1 Help desk – Call center

- Staffing, procedures, and operation (work hours, response time)
- Resources
- Records: follow-up files
- External assistance

**Note!** This section can be written (and implemented) at various levels, from a simple list of requirements for adding the system to an existing support center, on through a detailed Appendix 4.6.1 (at the end of the document, with the other appendices) when the existing center needs to be changed and enlarged, and on up to the construction of an “independent” support center to be developed as a separate (infrastructure) project with its own complete life cycle and S-Tree. In all cases, a good support center is an integral part of the system, with a presence in the system’s Application, Technology, and Implementation sections. Remember: A good system is a self-supporting system.

### 4.6.2 Application maintenance

- Configuration management and change control: tools and procedures
- Task owner
- Content revision (Internet systems)
- Software revisions (code)
- Changes in structure of files and tables

### 4.6.3 Infrastructure and technology maintenance

- Configuration management and change control
- Hardware and software vendors, external service providers — with type of service provided. Note that the agreements/contracts themselves are covered in section 4.5.1.
- Call time and service time
- Preventive maintenance
- Internal tracking by the organization's operation and production unit

### 4.6.4 Ongoing Implementation

Ongoing training, user-manual updates, ongoing marketing, etc.

**Note:** Management of the system's "Ongoing Implementation" can be dealt with in this section, in centralized fashion, for everything in section 4 (Implementation), or alternatively distributed throughout the section (for example, in section 4.7 with respect to integration into the organization, etc.). Do one or the other and avoid repetition.

### 4.6.5 Everyday costs

Tracking of everyday costs, including contracts with vendors

**Note:** Management of the tracking and control of everyday costs can be dealt with in this section or in section 5.2. Do one or the other and avoid repetition.

## 4.7 Deployment

☞ *See the corresponding Extension.*

### 4.7.1 Deployment

- Internal: orientation/commissioning plan, training system, and user motivation enhancement.
- External: marketing, distribution and advertising, expanding customer base.

For details and additional information, see the Deployment/Commissioning kit in the Enablers volume

### 4.7.2 Migration and Conversion

Objects such as —

- Files, tables, and interfaces
- Programs and procedures (software code)

Most conversions belong in the development process and in the project's work plan. This section is only for data (files and tables) that must undergo conversion/cutover as the system is installed and run in.

#### 4.7.3 Organization and methods

- Integration into the organization's procedures, everyday work, and functioning
- Anticipated problems in deploying the system
- Reassignment of workers
- Changes in worker roles and responsibilities
- Impact on the organization's procedures
- Work during the transition period (conversion)

The principal implications for Organization & Methods — the required changes, etc. — should be defined, documented, and approved in section 1.4 (and 1.4.3 in particular). This section, 4.7.3, deals with the practical implications of these changes and with the actions that must be undertaken on the ground to achieve deployment.

#### 4.7.4 User's guide

☞ *For details, see the corresponding Extension.*

### 4.8 Robustness and Reliability

#### 4.8.1 Testing plan

A summary of the system tests:

- Functionality testing: Screens, processes, input, reports, etc.
- Technology testing: Hardware availability and performance, platform software, communications, etc.
- Implementation testing: Conversion plans, commissioning plans, servicing and maintenance, etc.

☞ *For details and further information, see the corresponding Extension.*

See also the Testing kit in the Basics / Life Cycle volume.

**Note:** This section (like other sections of the S-Tree) “matures” during the course of the project. During Analysis it will contain a brief list of the basic system tests required. Later, during Design and Build, it will refer to Appendix 4.8.1 (Testing Plan), which elaborates on the description of the system tests. And finally, in Testing, it will refer to the full Testing document.

#### 4.8.2 Availability and survivability

- Criticality of the system, by dates, times, types of users, etc.



- Uptime
- Mean Time Between Failures (MTBF)
- Mean Time To Recover (MTTR)
- Underequipped operation
- Compliance with standards

## 4.9 Configurations

Section 3 (Technology) provides a generic rather than a quantitative definition of the required technology and infrastructure and can adequately cover relatively simple cases. It is this section, 4.9, that provides a “bill of materials” for precise configurations at the various sites where the system is to be installed. But in relatively simple cases, section 3 may suffice and section 4.9 may be unnecessary. And in all cases, workstations (client PC’s) are to be covered in components 3.20–3.22, not here.

### 4.9.0 List of configurations (installations)

#### 4.9.1 Development and Testing configuration

- Hardware, software, system libraries, work spaces, development tools

#### 4.9.2 Main (central) configuration — primary server

- Specification of configuration/contents
- Installation plan

#### 4.9.X Additional site configuration / distributed configuration X

- Specification of configuration/contents
- Procurement and installation plan for configuration type X

## 4.98 Open Issues (and Alternatives)

A compilation of the major open Implementation issues that must be resolved before the current version or delivery unit of the system can succeed.

- A comprehensive but succinct list
- A detailed description of each open issue or alternative, or a reference to the corresponding section 4X.99.

☞ *Use the Alternatives Analysis kit in the Enablers volume.*

## 4.99 Future Plans

Any important information from components 4.1–4.9 that entails further implementation in future versions or delivery units of the system.

## 5. Cost – Resources

In all the following sections, and especially in 5.1, 5.2, and 5.5, use the following kits:

- Cost Calculation in the Enablers volume
- Cost Estimation in the Enablers volume

### 5.0 Cost Overview

A concise presentation, at management level, of the system’s cost on the basis of section 5.5.

**Note:** This component, or a summary of section 5.5, is input for the Executive Summary.

### 5.1 Set-Up Cost (Development and Installation)

A summary of development and installation costs (set-up costs) for all anticipated versions and delivery units of the system.

#### 5.1.1 First/upcoming version or delivery unit

In the Inception, Analysis, and Design documents: the very first version or delivery unit of the system.

In the Maintenance document: the first version or delivery unit that has not yet emerged from development.

#### **Development cost (including testing and fixes)**

- Direct costs (of the project): Internal and external personnel, expenditures
- Indirect costs (to other projects, to the organization): personnel, expenditures?

#### **Installation and Deployment costs**

- Direct costs (of the project): Internal and external personnel, expenditures
- Indirect costs (to other projects, to the organization): personnel, expenditures?

#### 5.1.2 Additional versions and delivery units

The best possible estimate of the cost of further development cost, including additional versions and delivery units within the specified time frame (see section 1.7). Use either of the following methods:

- a. Combining all the anticipated versions and delivery units within the specified time and providing the following breakdown under the heading “5.1.2, Additional versions and delivery units”:
  1. Development cost (including testing and fixes)

- Direct costs (of the project): Internal and external personnel, expenditures
  - Indirect costs (to other projects, to the organization): personnel, expenditures?
2. Installation and deployment costs
- Direct costs (of the project): Internal and external personnel, expenditures
  - Indirect costs (to other projects, to the organization): personnel, expenditures?
- b. Listing each anticipated delivery unit. Section 5.1.2 will then decompose to the following subsections:

#### **5.1.2.1 <first delivery unit>**

Subdivide costs further according to direct and indirect development, installation, and deployment, as described above.

#### **5.1.2.x <subsequent delivery unit>**

Subdivide costs further according to direct and indirect development, installation, and deployment, as described above.

## **5.2 Ongoing Costs**

Ongoing costs such as operation, maintenance, and repairs, for the time frame established in section 1.7. The default time frame according to MethodA is five years.

### **5.2.1 First (upcoming) version or delivery unit**

- Direct and indirect operating costs
- Direct and indirect preventive maintenance and repair costs

Both types of cost — operating and maintenance/repairs — should include not only personnel of all kinds, computers, software, communications, space, etc. but also phone/fax, consumables, office supplies, travel, etc.

If the organization runs a centralized operation facility, the system's operating cost can be evaluated as the "delta-cost" to the existing organization-wide operation. Maintenance, on the other hand, is usually performed by a dedicated unit and accordingly maintenance costs are "autonomous."

### **5.2.2 Additional versions and delivery units**

The best possible estimate of operation and maintenance costs for all additional versions and delivery units within the specified time frame (see section 1.7).

MethodA’s minimum requirement in this section is to present together all the operation and maintenance costs for all anticipated versions and delivery units within the specified time frame, as in option (a) of 5.1.2 above. If this section can be broken down as in option (b), so much the better.

### 5.3 Cost by Configuration

All the installation and deployment costs from section 5.1, and all the operating and maintenance/repair costs from section 5.2, broken down by configuration and matched to the format of section 4.9 to present cost per configuration. For each configuration, future versions should be taken into account.

Configuration

Version

Installation and deployment costs

Operating and maintenance costs

**Note:** This component does not add to the costs in components 5.1 and 5.2; it merely extracts from them the cost by relevant system configurations, particularly for systems that are installed at different sites and/or at different times. All these costs are already included in sections 5.1 (Installation and Deployment) and 5.2 (Operation and Maintenance).

### 5.4 Price List

A table like this one, detailing the prices of relevant equipment and services:

Item	Unit price	Quantity	Total cost
Hardware components			
Third-party software			
Communications components			
Storage & maintenance			
Adding a report			
Adding file/page			
Manager’s fee			
System analyst’s fee			
Programmer’s fee			
General consulting			
Management consulting			
Technical consulting			

### 5.5 Cost Summary

#### 5.5.1 Cost of ownership

- Nominal cost
- Capitalized cost

### **5.5.2 Cost scheduling**

- By fiscal year
- By projected revenues/benefits

## **5.98 Open Issues (and Alternatives)**

A compilation of the main open Cost issues that must be resolved before the current version or delivery unit of the system can succeed.

## **5.99 Anticipated Future Costs**

Information concerning future costs that might crop up during the system's life span, but only *later than* the time frame discussed in section 5.2; for earlier costs, refer to section 5.2 and in particular to 5.2.2.

## Appendices

As a rule, any S-Tree section can have a corresponding appendix, bearing the same title and number, that expands upon its contents. The body of the document (i.e., the trunk of the S-Tree) should not be longer than 100 pages, suitable for reading at one sitting and with convenient references to the appendices.

**As a matter of principle MethodA does not use appendices numbered A, B, etc. Each appendix is numbered for a specific component in the S-Tree.**

Some typical appendices are:

### **Appendix 1.6.1: Risk Analysis and Feasibility**

☞ *Use the Risk Analysis kit in the Enablers volume.*

### **Appendix 1.6.2: Cost/Benefit Analysis**

See Appendix 1.6 (Feasibility and Cost/Benefit) in the S-Tree Appendices kit.

### **Appendix 2.7.1: Source Modules**

This appendix is to be used where a document specifying the structure and logic of every program is required.

In most cases, however, program specification will be developed as part of the physical module during the Design, Build, and Code phase using modern code development tools. The physical modules, in turn, will be located in the module library and referenced by section 2.7.1 in the body of the document, making Appendix 2.7.1 unnecessary.

#### **2.7.1.X <name of source module>**

- Textual description
- Flowchart
- Link upward to transactions (and processes)

☞ *For details of the structure, see the corresponding Extension.*

Also potentially helpful are the study template and work template for the Programming document, in the Design and Build kit, within the Life Cycle volume.

### **Appendix 4.2: Work Plan**

1. Detailed GANTT charts
2. Task lists

- For teams
- For individuals

3. Performance / utilization of resources

**Appendix 5.1: Estimated Set-Up Cost**

Details of the tables and calculations that were the basis of the costs in section 5.1

**Appendix 98: Open Issues – Alternatives Analysis**

See the Alternatives Analysis kit in the Enablers volume.

**Appendix 99: Future Requirements**

**Appendix X.Y**

Additional detail for any section that is becoming cumbersome, e.g., Appendix 2.21 (Workload, Performance, and Capacity) for detailed performance and capacity tables.