|  |  |  |
| --- | --- | --- |
| Document ID  **ITSW103** | Title  **SYSTEMS ANALYSIS** | Print Date  **mm/dd/yyyy** |
| Revision  **0.0** | Prepared By  **Preparer’s Name / Title** | Date Prepared  **mm/dd/yyyy** |
| Effective Date  **mm/dd/yyyy** | Reviewed By  **Reviewer’s Name / Title** | Date Reviewed  **mm/dd/yyyy** |
|  | Approved By  **Final Approver’s Name / Title** | Date Approved  **mm/dd/yyyy** |

**Policy:** All software products developed by the company must be designed to meet certain specific needs of potential users.

**Purpose:** To create:

* Documents that define the functions the system will perform;
* An acceptance testplan, which describes how to test those functions; and
* A beta test plan, which describes how to conduct a user test of those functions.

**Scope:** All software products and updates released by the company.

**Responsibilities:**

The Systems Analyst is responsible for developing and communicating the overall design approach that addresses the customer requirements, as well as producing the test plans.

**Definitions:** Systems analysis - Work that involves applying analytical processes to the planning, design, and implementation of new and improved information systems to meet the business requirements of customer organizations; phase of the SDLC in which the current system is studied and alternative replacement systems are proposed.

Systems development life cycle (SDLC) - A method for developing information systems, made up of five main stages: analysis, design, development, implementation, and evaluation. Each stage is further comprised of several components (for example, the development stage includes programming, debugging, testing, and documenting).

**Procedure:**

### 1.0 SYSTEMS ANALYSIS - INTRODUCTION

1.1 Systems analysis is a process of refinement. The analyst begins with a general model for a software product and gradually develops and refines it until it fulfills the requirements identified by users and subject matter experts. Good methods, patience, and attention to detail save time in the long run.

1.2 The Systems Analyst might use the following tools:

* A word processing program for typing text;
* A paint or draw program for creating graphics;
* A flow charting program or CASE tool for documenting process flows; and
* A central database for storing specifications, charts, and images.

NOTE: If possible, scan sample documents to disk and store them with the specification files.

### 2.0 SYSTEMS ANALYSIS - REQUIREMENTS

2.1 The Systems Analyst interviews users and subject matter experts to determine the functions the software must perform.

2.2 The Systems Analyst researches how potential users are currently performing these functions.

For example, if the software is a word processing system, the Systems Analyst researches how potential users are currently designing and creating documents. Research methods may include manual or paper systems, competing software systems, or industry standards the software must meet. See references A and B.

2.3 The Systems Analyst creates a document that specifies the functional requirements of the software.

2.4 The Systems Analyst reviews the requirements with users and subject matter experts and makes any necessary changes.

### 3.0 SYSTEMS ANALYSIS - INFORMATION FLOWS DOCUMENTATION

3.1 The Systems Analyst creates a high-level diagram that shows the information the software system manages and the way in which this information flows through the system. The diagram should contain all of the components required to support the information flows, including:

* Inputs;
* Outputs;
* Processes;
* Calculations; and
* Relations.

3.2 Large projects should be divided into subsystems with each subsystem individually designed. The Systems Analyst reviews the diagram with users and subject matter experts, and then makes any necessary changes.

3.3 The Systems Analyst writes a detailed description of each of the flows in the diagram, and of each input, output, process, calculation, and relation.

3.4 The Systems Analyst reviews the detail descriptions with users and subject matter experts, and then makes any necessary changes.

### 4.0 SYSTEMS ANALYSIS - ACCEPTANCE TEST PLAN

4.1 After defining and documenting the system requirements, the Systems Analyst creates an acceptance test plan. When programming for the project is complete, the Systems Analyst uses this test plan to verify that all elements of the software work as designed.

4.2 The Acceptance Test Planincludes:

* A list of the test team members, their titles, and their duties. Members of the test team typically include:
  1. Quality Assurance Manager;
  2. Systems Analyst;
  3. Software Designer;
  4. Users; and
  5. Subject Matter Experts.
* Instructions for setting up the software after it is released from development.
* Descriptions of the data required during the test. For example, a description of the data required to test an order entry database might include sample customers, parts, and orders.
* A step-by-step test procedure that specifies exactly what to do and what the expected results are.

### 5.0 SYSTEMS ANALYSIS - BETA TEST PLAN

5.1 In addition to the acceptance test plan, the Systems Analyst also creates a beta test plan. After the software has been programmed and tested internally, a selected group of users performs a beta test. The beta test plan explains how to coordinate the beta test.

5.2 The beta test plan describes how to:

* Select user sites – the sites selected should accurately represent the product’s target market.
* Install the software at the beta sites. Specify any installation procedures that differ from those used for the official release. For example, the beta release might need to be installed manually, while the official release contains an automated installation procedure.
* Train beta testers on the software. For example, provide written training materials or on-site training for the beta testers.
* Support the beta release. Specify the individuals responsible for answering beta testers’ questions. Include individuals from analysis, design, programming, and documentation.
* Report problems found during testing. Give each test site a packet that includes sample problem reporting forms and contact names and phone numbers.

### 6.0 SYSTEMS ANALYSIS - REVIEW

6.1 The Systems Analyst should present the systems analysis documents in a formal review to all interested people in the company, including:

* The Software Designer;
* Programmers;
* Technical Writers;
* Members of upper management; and
* Marketing and sales representatives.

6.2 Document ideas, comments and concerns for possible investigation. Plan on spending as much time as necessary to answer questions before turning the analysis over for software design. The more time spent in the early planning phases will help to create an easier coding specification and save time later in the software design phase.

**Forms:**

* None.

**References:**

* 1. **ISO/IEC 12207:2008, “SYSTEMS AND SOFTWARE ENGINEERING – SOFTWARE LIFE CYCLE PROCESSES”**
  2. **IEEE 12207-2008, “SYSTEMS AND SOFTWARE ENGINEERING – SOFTWARE LIFE CYCLE PROCESSES”**

ISO 12207 describes the major component processes of a complete software life cycle and the high-level relations that govern their interaction. It establishes a software life cycle architecture based on two principles, modularity of processes and responsibility for processes. There are three process classes in the ISO software life cycle: primary; supporting; and organizational. Each life cycle process is made up of activities and each activity is subdivided into tasks. ISO 12207 is based on ISO quality management principles.

The IEEE version of 12207 is closely aligned with, but not an exact duplicate of, its ISO counterpart.

For more information, visit the ISO web site at <http://www.iso.org> or the IEEE web site at <http://www.ieee.org/>.

**Revision History:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Revision** | **Date** | **Description of Changes** | **Requested By** |
| 0.0 | mm/dd/yyyy | Initial Release |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

[This page intentionally left blank]