|  |  |  |
| --- | --- | --- |
| Document ID  **ITSW104** | Title  **SOFTWARE DESIGN** | 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:** To design software in a technically sound and efficient manner and fulfill requirements identified by the systems analyst.

**Purpose:** To transform a set of system requirements (developed by the systems analyst) into programming instructions for a software product.

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

**Responsibilities:** The Software Designer is responsible for transforming system requirements developed by the systems analyst into programming instructions and then communicating the overall design approach.

**Procedure:**

### 1.0 SOFTWARE DESIGN – INTRODUCTION

1.1 The Software Designer transforms the system requirements and other design documents developed by the systems analyst (see ITSW103 SYSTEMS ANALYSIS) into instructions and specifications for programming a software product.

1.2 The Software Designer might use the following tools:

* A word processing program, for typing text;
* A paint or draw program, for creating graphics;
* A flow charting program, for documenting data flows;
* A source code control system, for controlling program revisions; 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 SOFTWARE DESIGN SPECIFICATION

2.1 The Software Designer shall write a description of the programming environment. The description should include instructions for:

* Locating the programming system (its directory/account structure);
* Accessing the programming system (startup, login);
* Applying for the required user codes;
* Locating programming libraries and tools;
* Checking components into and out of the software component library or source code control system[[1]](#footnote-1);
* Assigning tasks to programmers;
* Reporting and tracking problems (or bugs); and
* Returning completed work to the designer.

2.2 The Software Designer shall create a general design of the software required to fulfill the system requirements developed by the systems analyst. To do this, the designer must:

* Fully address each process, calculation, relation, and flow defined in the system requirements;
* Design software components that leverage the strengths of the programming tools and run efficiently in the physical environment; and
* Incorporate internal and industry-accepted standards of design.

2.3 The Software Designer shall review the general design with the systems analyst and make any required changes.

2.4 The Software Designer shall write detailed programming instructions for each component in the design. The following table shows the types of components and instructions found in a typical database software design.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Programs** | **Forms** | **Reports** | **Menus** | **Database** |
| program name | form name | report name | menu name | database name |
| description | description | description | description | description |
| screen layout | screen layout | screen layout | menu layout | table names |
| process logic | field edits | report layout |  | field names |
| messages | process logic | query logic |  | keys & indexing |
|  | messages | process logic |  | format rules |
|  |  | messages |  | edit rules |
|  |  |  |  | update rules |

2.5 The Software Designer shall create a catalog of messages used in the software. For each message, the Software Designer indicates the action the system users or administrators must take.

2.6 The Software Designer shall review the detail design with the systems analyst and make any required changes.

### 3.0 SOFTWARE DESIGN REVIEW

3.1 The Software Designer should use ITSW104-1 DESIGN REVIEW CHECKLIST as a guide to preparation before presenting the software design documents in a formal review to everyone who will be working on the project, including:

* The project manager;
* The systems analyst;
* Programmers;
* Technical writers; and
* Quality assurance analysts.

3.2 The software design should be reviewed for compliance with overall design objectives, including:

* A clear understanding of the user environment, requirements, and system analyst specifications;
* Use of best practices in software design, including effective design strategies, modularity, performance, and extensibility; and
* Clear process flows, data integration, and data models.
  1. Document ideas, comments, and concerns for possible investigation. Plan on spending as much time as necessary to answer any questions before turning the design over for programming. The more time spent in the early planning phases will help to create easier coding and save time later in the software programming phase.
  2. Users may request design changes during this or any other phase of the software development life cycle; users shall submit change requests in accordance with ITSW108 DESIGN CHANGES DURING DEVELOPMENT.

**Forms:**

* ITSW104-1 DESIGN REVIEW CHECKLIST

**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**

This ISO standard 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 (such as acquisition and operations); supporting (such as documentation and configuration management); and organizational (such as infrastructure and training). Each life cycle process is made up of activities, and each activity is further subdivided into tasks. The standard is based on ISO quality management principles.

The IEEE version of 12207 is closely aligned with, though not exactly the same as, its ISO counterpart. For more information, visit <http://www.iso.org/iso/catalogue_detail.htm?csnumber=43447> and/or <http://standards.ieee.org/findstds/standard/12207-2008.html>.

**Revision History:**

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

**ITSW104-1 DESIGN REVIEW CHECKLIST**

|  |  |  |
| --- | --- | --- |
|  | **DESIGN REVIEW** |  |
| **ID** | **ITEMS TO BE CONSIDERED** | **RESPONSE** |
|  | Have the work products to be reviewed been identified? |  |
|  | Has the type of review been selected? Alternatives include:   * Informal walk through by several team members * Technical review by project team members and stakeholders * Inspection by project team members(and perhaps others) |  |
|  | Have the goals of the review been established? |  |
|  | Has a moderator/facilitator been selected? |  |
|  | Has a review package been developed and distributed to the participants with ample review time? The review package should include at least the following:   * Work product to be reviewed * Related templates, guidelines, other background information * Forms with which to record defects, questions, issues |  |
|  | Has the software design been reviewed for compliance with overall design objectives including:   * A clear understanding of the user environment, requirements and system analyst specifications. * Use of “Best Practices” in software design including effective design strategies, modularity, performance and extensibility. * Clear process flows, data integration and data models. |  |
|  | Have results of the review been used to update the work product? |  |
|  | Have the goals of the review been reviewed to determine success? |  |
|  | Has the process been reviewed to identify any improvements? |  |

[This page intentionally left blank]

1. For information about the software component library, see ITSW109 SOFTWARE RELEASES AND UPDATES procedure [↑](#footnote-ref-1)