

Instructions

1. Collection of requirements

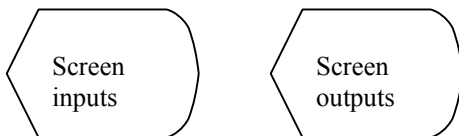
1.1. Obtain output reports

Request copies of all hard copy output from customer's current system.



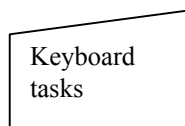
1.2. Capture input and output screens

Request screen printouts of all input and output screens.



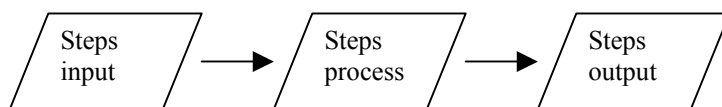
1.3. Document input tasks

Interview each employee having unique input tasks; list in order of input sequence the tasks including screen identifier, data element name, data element type, constraint conditions, and any special instructions.



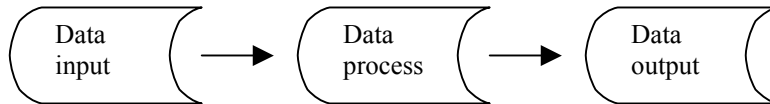
1.4. Determine processing steps

Categorize and index the steps required for processing input data to produce output screens and reports.



1.5. Map data flow

Trace the steps of movement of data from input through processing to output into a data flow diagram.



1.6. Collect user change requests

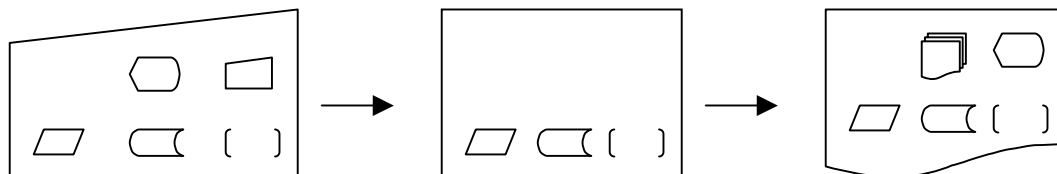
Solicit user requests to change the current system; user requests should be as explicit and detailed as practical.



2. Analysis of requirements

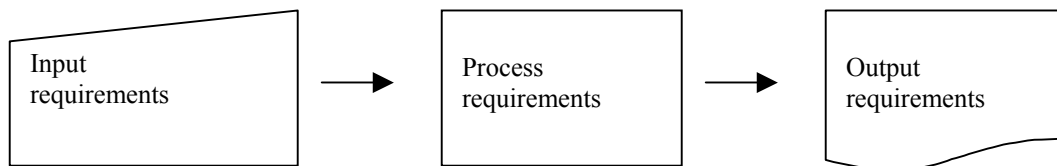
2.1. Collate and index requirements

Combine, sort, and index the current requirements and user requests from item 1. above.



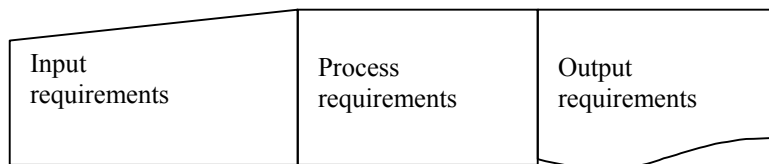
2.2. Produce requirements document

Edit and rewrite requirements for consistency.



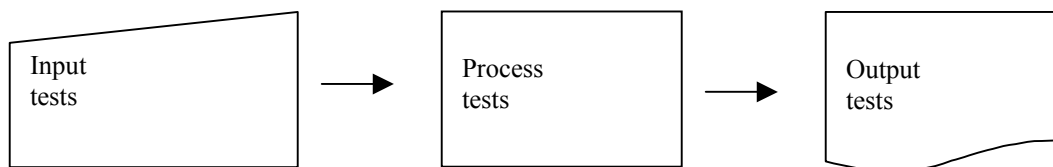
2.3. Approve requirements by user

Submit requirements to the customer for approval.



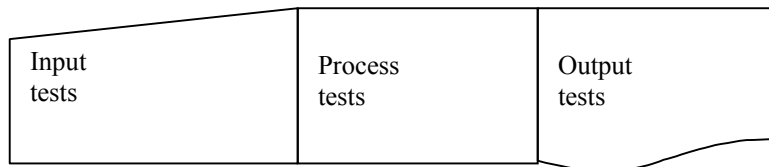
2.4. Produce test descriptions and cases

Produce a test description and test case for each indexed requirement.



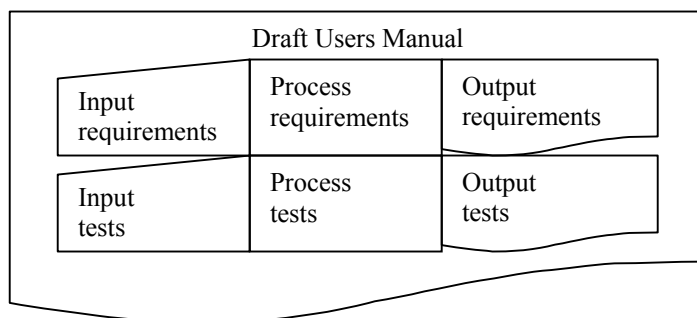
2.5. Approve test cases by user

Submit test cases to the customer for approval.



2.6. Combine documents into user manual draft

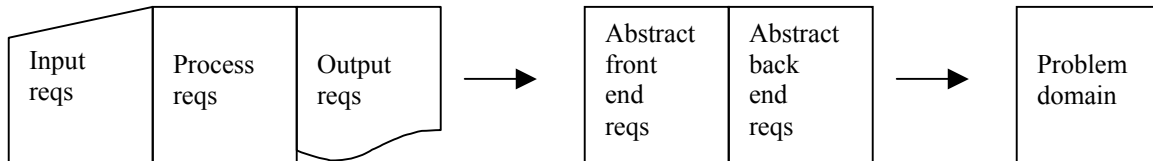
Interleave requirements and tests for input, processing, and output into a draft user manual.



3. Design based on analysis

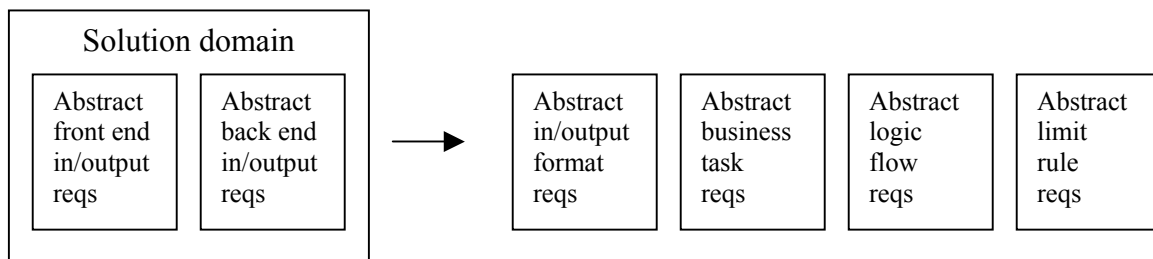
3.1. Generalize requirements

Abstract the requirements to the next higher logical level, repeating this process until the model is abstracted into only one component which is the problem domain.



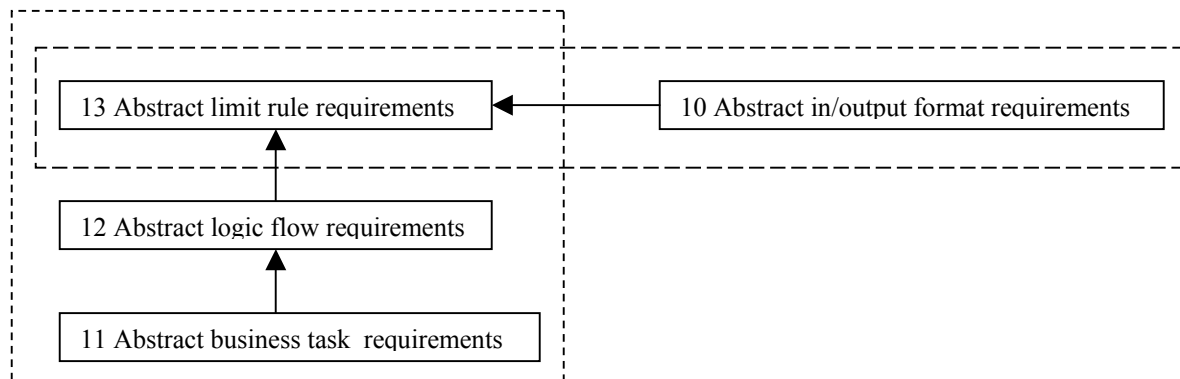
3.2. Determine the abstract solution domain

Decompose the solution domain into specifications of solution components.



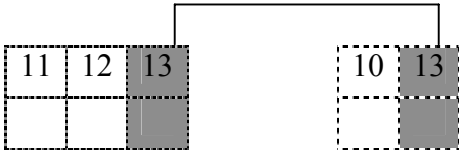
3.3. Map solution components into classes and attributes

Translate on a one-to-one basis the solution components into attributes, showing attributes within solid lines and inheritance as an arrow from child attribute pointing to parent attribute, and into classes of attributes, showing classes within dotted lines.



3.4. Map classes and attributes into tables and columns

Translate on a one-to-one basis attributes and classes into columns and tables, showing common parent attributes in tables as linked by lines.



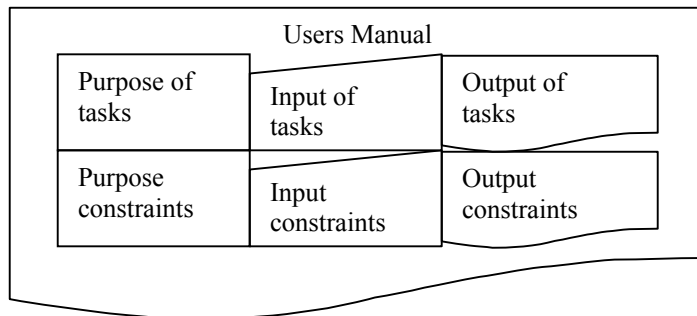
3.5. Map task actions into logic switches

Translate on a one-to-one basis the task actions on tables into logic switches.

Column 13 row	Task actions		
	A	B	C
1001	a	o	c
1102	a	b	o
1203	o	b	c

3.6. Produce finished user manual

Recompile the draft user manual based on further specifications of tasks in the solution domain.



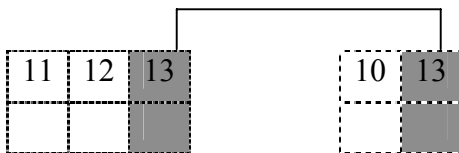
4. Implementation of design and unit testing

4.1. Produce relational database table classes

Name tables according to the computer software systems and components.

Table 1: Formats

Table 2: Tasks



4.2. Produce relational database column attributes

Name table columns according to the computer software items or units for each system or component.

Table 1: Formats

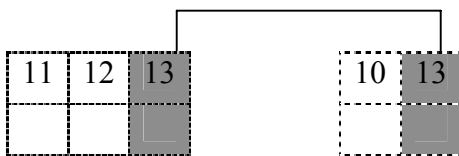
Table 2: Tasks

11 Task column

10 Format column

12 Flow column

13 Rule column



4.3. Produce relational database logic switches by task

Define task names and specify logic switch position according to logic switch specification.

Task A: From column 13 row(s) insert the data “a” into column 12 of table 1.

Task B: From column 13 row(s) insert the data “b” into column 11 of table 1.

Task C: From column 13 row(s) update the data “c” into column 10 of table 2.

Column 13 row	Task actions		
	A	B	C
1001	a	o	c
1102	a	b	o
1203	o	b	c

4.4. Test input task logic flow

Perform test cases for correctness of tasks based on logic switch assignment.

If task A is not “o” then insert “a” into column 13 row values 1001 and 1102 in Table 1.

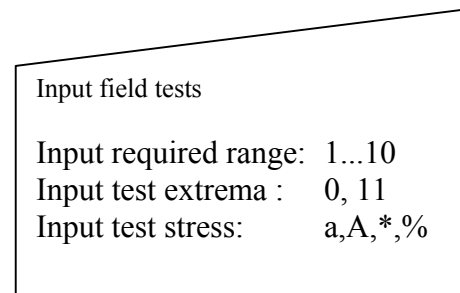
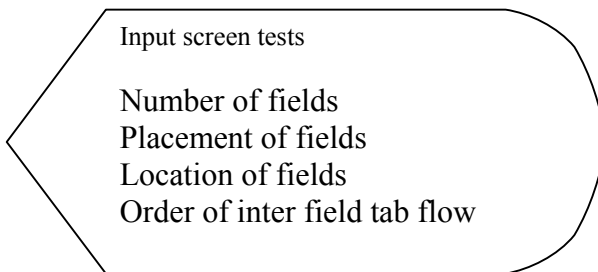
If task B is not “o” then insert “b” into column 13 row values 1102 and 1203 in Table 1.

If task C is not “o” then insert “c” into column 13 row values 1001 and 1203 in table 2.

Process tests	Column	Task actions		
	13 row	A	B	C
	1001	a	o	c
	1102	a	b	o
	1203	o	b	c

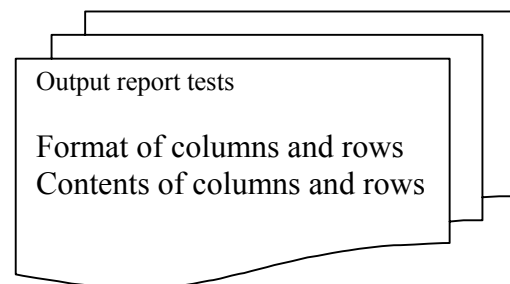
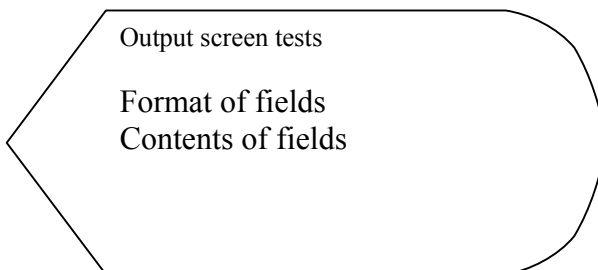
4.5. Test screen inputs

Perform test cases for boundary and stress conditions of screen inputs.



4.6. Test screen outputs and report outputs

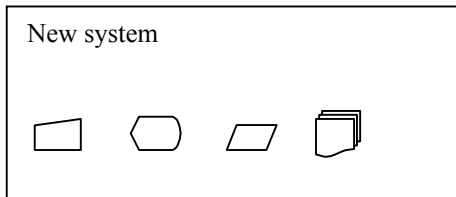
Perform test cases for screen outputs and report outputs.



5. Acceptance testing and user approval

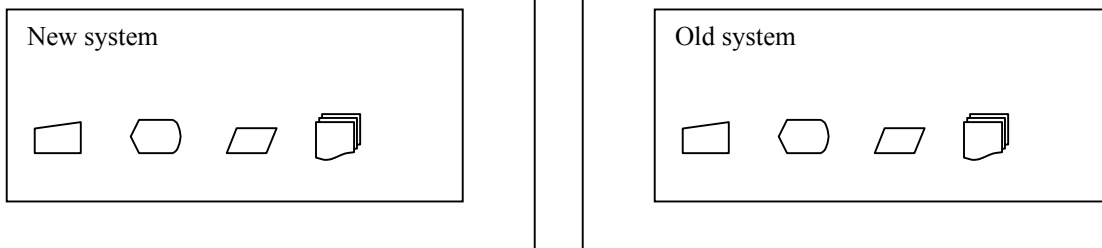
5.1. Perform system integration testing

Demonstrate to the user that all pieces of the product work together as a system.



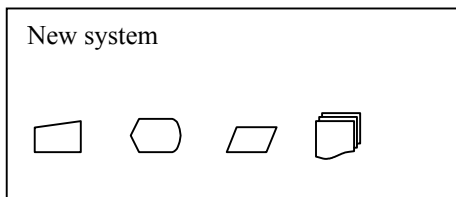
5.2. Operate new system in parallel with old system

Run the new system in parallel with the old system to mirror in the new system all input and output of the old system.



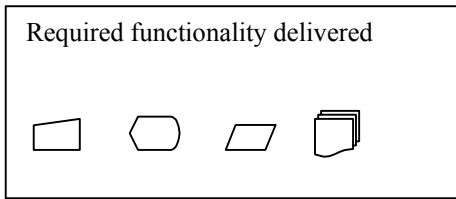
5.3. Approve systems test by user

Obtain approval by the user that all test cases passed.



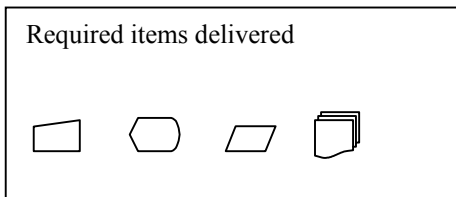
5.4. Verify delivery of required functionality

Determine that the product to be delivered meets the functionality of all requirements.



5.5. Validate delivery of required items

Determine that the product pieces to be delivered are present and complete.



5.6. Approve delivered product by user

Obtain final approval sign off by the user, including yearly maintenance agreement.

