

SECTION 5 – SMALL PROJECT METHOD

The small project method condenses the full-scale development method into five stages and a management layer. These stages represent a standard configuration of the tasks and documentation for projects not subject to the SDLC threshold criteria; the stages below form the base from which the project team may modify tasks and documentation as required in order to address the needs of their particular project.

A. Project Management

The project management includes activities that have been considerably simplified from the activities in the large development approaches.

General Management tasks include:

Define, document, and get appropriate sign-offs on project scope.

Manage personnel resources

- Set up project advisory group
- Manage day-to-day activities
- Manage risk actively
- Monitor key progress indicators

Maintain user relationship

- Establish sign-off process
- Maintain communications with project advisory committee
- Manage expectations

Manage issues, open points and change requests

- Identify and log items
- Assign responsibility
- Facilitate and monitor resolution process
- Report item status
- Escalate items to sponsor or advisory committee
- Communicate resolutions

B. Requirements Analysis

The small project method requires analysis activities in order to understand the requirements, including their structure and organization. This method combines activities from requirements gathering and analysis, reflecting the conditions of a small team of experienced personnel working with a single business process.

Analysis includes several high-level activities including the identification of user requirements and development of events, data, and process prototype.

User requirements analysis

This analysis is intended to understand what the new application must do to satisfy the needs perceived by its users.

General user requirements analysis tasks include:

- Planning and conducting interviews (may be either one-on-one interviews or Joint Application Development (JAD). For further information on JAD see the “References” in the VI REQUIREMENTS SPECIFICATION AND PROTOTYPE phase of the DEHNR SDLC. Determine what information to collect. Select interviewees. Schedule sessions.
- Where appropriate conducting external reviews. May include review best practices, site visits of comparable organizations, review of packaged software for additional functionality.
- Performing gap analysis by:
 - Documenting present business process
 - Designing/reviewing preferred business process
 - Highlighting system changes required to bridge the gap resulting from new workflow and organization, resulting from present operational difficulties, and resulting from present technical deficiencies.

Events analysis (optional)

This analysis is intended to model the external behavior of the application according to its business transactions and events.

General events analysis tasks include:

- Based on the scope of the project identifying a list of external events including transactions, alerts such as inventory levels, and sensor events such as time.
- For each event, determining stimulus, describing overall response and any generated events. Iterating as required. Annotating event-stimulus-response description with volumes, frequencies, and desired response times.
- For each major entity, creating an entity life cycle diagram (See data analysis below.) If required, creating an entity by event matrix. Verifying completeness of event model relative to the requirements, relative to present system, relative to data and process models.
- Validating event model with users.

Data analysis

This analysis is intended to model the application's data requirements, independent of implementation details and to provide a basis for designing the application's database and file structures.

General data analysis tasks include:

- Defining the scope of the data model.
- Defining entity types, type hierarchies, and a unique identifier for each entity type.
- Defining relationship types based on business rules and policies.
- Defining attribute types.

- Normalizing and verifying the completeness of the data model relative to the user requirements, present system, and event and process models.
- “Scrubbing” existing data.

Document and confirm requirements

Gather and model the requirements into a requirements specification. This specification captures the events, data, and process aspects of the system, as well as its quality attributes, all of which are key inputs to the design process.

Business Process Prototype

This analysis is intended to create the basis for user approval of requirements on *shown features* rather than written descriptions, to confirm that the specified level of usability can be achieved, and to ensure that innovative or high-risk features are functionally and technically feasible.

General prototyping tasks include:

- Defining scope of prototype
- Developing the prototype review plan
- Selecting the prototyping environment
- Designing the prototype. Pay particular attention to the overall decisions that affect the user interface metaphor, levels of user expertise to support, support and control of the work flow, hardware, and software platform (in particular hardware features such as large screens, scanners and other special-purpose devices), and implications for the execution architecture
- Selecting events, processes and data to be prototyped
- Building the prototype
- Applying the methodology prescribed by the prototyping tool
- Testing and reviewing prototype with selected users. Documenting reactions and requested revisions, and iterating as required.

Since the main goal of a prototype is to gauge the users' reaction to the future application, develop several sections of the application to provide data entry and retrieval capability, full navigation, etc.

C. Detailed Design

The Design stage of the small project method uses the requirements specification as the basis for system design. Design activities in the small project method are heavily influenced by the use of existing development and execution architectures. These activities reduce technical risk to a certain degree, reducing the need for architecture and several other technical design tasks. The activities in the design stage produce detailed designs of the user interface, database and processing for implementation in the Build Stage.

General detailed design tasks include:

- Designing dialogue features including the user interface and reports and documents.

- Designing the application architecture including the messaging and processing flow and the logical and physical database. The purpose of this task is to identify all programs in the application, their sequence, and how they communicate. Typical steps followed in this task include:

- Identifying client, server, traditional online, asynchronous and batch programs
- Identifying processing sequence and inter-program communication
- Identifying interfaces to other systems
- Designing program-to-program, internal program, and file-generation controls
- Identifying additional programs required for ancillary processes and iterations.

Designing logical and physical databases. Typical steps followed in this phase include:

- Review of existing databases and distribution approach.
- For each entity type, choose the data management software (for example, relational DBMS, flat file)
- Identify tables and for each table: 1) Define all columns. 2) Define primary key.
- 3) Evaluate row length and adjust as needed.
- Design foreign keys and referential integrity rules
- Design logical database file structures.
- Review design with appropriate management and staff
- Design physical database: relational database management system (RDBMS) structures.

- Designing the approach to testing ensures that all the time and resources required for testing will be available during the test stage of the small project path. Typical steps in this task include:

Identifying the required testing stages (from unit test through acceptance test)

For each testing stage, determining

- Objectives
- Requirements against which to test
- Who will execute?
- Environment
- Approach to regression testing (testing to assure changes in one unit of the system does not affect other units.)
- Approach to creating, maintaining and sharing test data
- Tool requirements

- Confirm tool availability

D. Building System

This stage produces tested executable code, procedures, training and the system test components. The project team writes and checks the code, then prepares test data and expected results that will fully explore all of the operational conditions that an individual module will face. The team then runs the unit and phase tests, checking the results against those that were expected, and corrects any errors. In parallel with these activities, the team creates the user procedures and training, and prepares to begin the Test stage.

General Building tasks include:

- Developing and documenting systems performance, security, control and operations procedures with an identification of anticipated users, drafting of documentation, and development of help materials.

- Generating and coding work units. To produce executable code for each designed work unit, review the specifications and complete the programming work units. Desk-check the code, developing any unit-testing routines, compiling until correct, and updating the development status.
- Conducting unit testing to ensure that the programming work units properly perform all required functions. Execute each test cycle. Check results. Verify actual to expected results. Make changes, and retest as necessary.
- Performing code reviews. (optional) To identify where the program does not conform to specifications and to ensure that all code is written and documented according to project standards. Assign reviewers and arrange meetings, document the results, clear the points, notify reviewers of changes, and update the development status.
- Preparing system tests. To ensure that the system test will be thorough finalize dates and assign people, confirm requirements such as hardware time requirements, prepare the test data, finalize the test conditions, and define the test cycles. System testing for small projects refers to the integration and user acceptance tests.

E. Testing

In this stage, the project team is responsible for testing the new system and training the personnel in preparation for production. Test the configurations promoted from programming and review the results. Then conduct an integration test and monitor the system's performance, making changes as necessary. Investigate and correct discrepancies in the results. The objective in this phase is to ensure users and operations personnel know how to use and run the system and to verify before conversion that the new system contains all required functions that work correctly.

General Testing tasks include:

- To verify that the programs in the new system communicate properly with each other review the test plans (These plans should be developed with substantial user involvement including preparation of the test data.), verify that resources are available, develop and execute test cycles, verify results, and obtain sign-off.
- Selecting/promoting the system configuration (e.g. selecting and organizing the components of a system.) To promote the system configuration define the configuration, confirm the availability of the system components, verify that the target environment is ready, and select the method to be used to promote the system (i. e. move the system from the test environment to the production environment.) Promote the configuration and verify that the promotion has been successful.
- To verify that the new system can operate effectively in a full production environment perform the system quality and integration tests. Follow the same general test steps listed above.
- To ensure users and operations personnel know how to use and run the system confirm the training session agenda, conduct the training sessions, evaluate the training sessions, and establish a continuing training program.
- To ensure that the system is ready to be rolled out: Conduct a final quality assurance review that reviews the tests and the roll-out plans.

F. Rolling-out system

The small project method assumes the roll-out occurs for an initial site, with subsequent efforts used to execute and manage roll-outs to multiple sites. A successful roll-out involves more than just using the new programs and procedures. The work environment must reflect only the new system; remnants of the old system must be removed when it makes sense. The project team must review and analyze the system to verify that it performs the functions stated in the design. Finally, the project team must document, for future implementation, any changes or enhancements identified during the initial production cycles that would make the new system more responsive to user and business needs. The small project method defines the roll-out process for an individual site. Repeat the activities in this stage for each distributed site and once for a central site.

General Roll-out tasks include:

- To verify that operations personnel can effectively support the new system and to ensure user acceptance of the new system, perform a readiness test. The general steps are the same as those used in earlier testing tasks.
- To convert the site, develop conversion plan, verify that prerequisites and resources are in place, review conversion process with conversion team, resolve open issues, convert files and “scrubb” data where necessary, verify results, remove old system, establish user liaison, develop a maintenance manual, and transfer responsibility.
- To ensure that the new system is functioning as designed provide support during initial production, monitor progress, record and control variances, make adjustments where necessary. Document all change requests.

End of Section 5