

Guideline

For

Tailoring the Project

**Revision History**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Ver. No.** | **Date** | **Comments** | **Prepared By** | **Reviewed By** | **Approved By** |
| Ver. 1.0 | 4th March,  2009 | Initial Draft | Jyotsna Bareja | Abhishek Rautela | Mr. Sudhir Saxena |
| Ver. 1.1 | 8th November , 2010 | Update in Tailoring Process | Neha | Rohitash | Mr. Sudhir Saxena |
| Ver. 2.0 | 16th August, 2011 | Formatting of the contents and Update in Introduction & SDLC Model | Rahul Raj | Dhananjay Kumar | GM Dua |
|  |  |  |  |  |  |

Table of Contents

**Version History2**

**1.0 Introduction4**

1.1 Purpose4

1.2 Scope4

1.3 Glossary4

**2.0 Description of the Tailoring Process4**

**3.0 Software Development Life Cycle Model6**

3.a Waterfall Model6

3.b RAD Model6

3.c Incremental Model7

3.d Prototype Model7

3.e Spiral Model8

3.f Joint Application Life Cycle Model8

3.g Agile Model8

**4.0 Tailoring Life Cycle Models9**

**5.0 Phases under operational process10**

1. **Introduction**

The definition of a software process is embodied in how a software organization does business; that is, how management and engineering practices are implemented to support software development and/or maintenance. This view of software process assumes that an organization has a set of building blocks that defines the general way it does business and that some subset of those building blocks is implemented for each software project.

The building blocks that define how a software organization does business are called the organization's standard software process. The subset of the building blocks tailored to and implemented for each project is referred to as the project's defined software process. Then, the project's software development plan is developed based on the project's defined software process and used to manage the software project

**1.1 Purpose**

The main purpose of this document is to provide a guideline for tailoring the Software Life cycle as described in the SDLC Process to fit the specific characteristics of a project. Tailoring is to be done in a thoughtful and disciplined manner. Therefore, all projects in NST are to follow this guideline for tailoring the approved Software Life cycle models.

**1.2 Scope**

This is applicable for all types of projects that are carried out at NST.

**1.3 Glossary**

**Tailoring** – Tailoring is a process for selection of the appropriate Software Life cycle model and modification of the same to fit the specific requirements of the project when the projects are not following all the steps of the chosen life cycle model.

**SLC** – Software Life Cycle

**SDLC** – Software Development Life Cycle

**SMLC** – Software Maintenance Life Cycle

**JAD** – Joint Application Life cycle

**RAD** – Rapid Application Life cycle

1. **Description of the Tailoring Process**
   1. Depending on the requirements as expressed in Scope and/ or other documents where the requirements are stated, the Project Manager will select the appropriate life cycle from the Software Life cycle Process.
   2. If it is seen that the project cannot be carried out with any specified model in the SDLC Process, then the appropriate life cycle procedure will be described in the relevant section of the Project Management Plan and the same is to be approved by SEPG.
   3. If the new Life cycle is not approved by SEPG, then the Project Manager or any authorized person will make the changes as required and get approval from SEPG.
   4. For an existing life cycle model, there may be some phases that may not be applicable for a particular project. The Project Manager or any authorized person may pick up the applicable phases and mention the same clearly in the relevant section of the Project Management Plan.
   5. The tailoring shall match the requirement as expressed in the scope/ contract document.
   6. After choosing a particular life cycle, later if it is seen that another life cycle may be more appropriate, the Project Manager may change the existing life cycle after assessing the impact on delivery schedule. If the delivery schedule is impacted as expressed in scope/ contract document, the same will be intimated to the client for their approval.
   7. The description of all the life cycles as a whole stating what life cycle is to be selected at what circumstances is given in the sections given below. This is stated as a guideline only and the Project Manager will choose as appropriate as possible. However, after choosing the appropriate life cycle, tailoring is to be done as required.
      1. Types of Projects: The Projects at NST can be broadly classified into two types based on the software Life Cycle:
         1. Life cycle Projects that follow the Software Life cycle
         2. Maintenance Projects that follow the Software Maintenance Life cycle (SMLC)

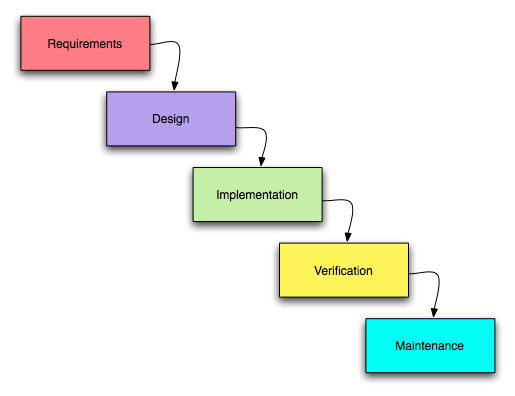
However, both of these life cycles (called SLC as a combination) are defined in SDLC Process.

1. **Software Development Life Cycle Model**

The Life cycle Projects can follow different kinds of the Life cycle Models depending upon the project. The description of the models and the phases within are described in the SDLC Process.

1. ***Waterfall Model or Linear Sequential Model***

This model is best suited when the user requirements are stable and known at the early stages of the life cycle. Each step is to begin when the earlier step is closed.

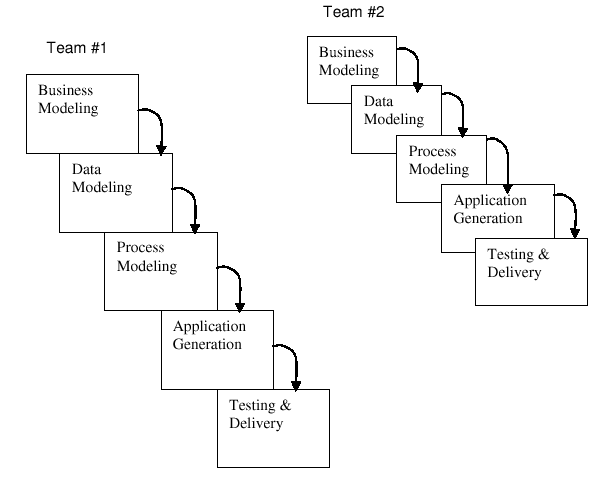


*Waterfall model*

1. ***RAD Model***

This model is “high speed” adaptation of the Waterfall model in which rapid life cycle is achieved by using a component based construction approach using “RAD tools”. This model is best applicable for projects that have a limited scope and the requirements are well understood. This is primarily applicable for information

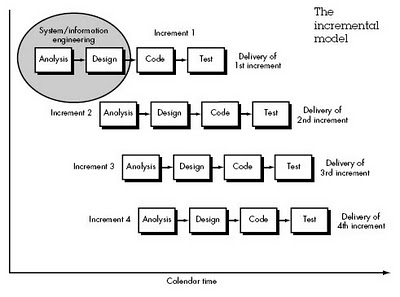
System applications.



*RAD model*

1. ***Incremental Model***

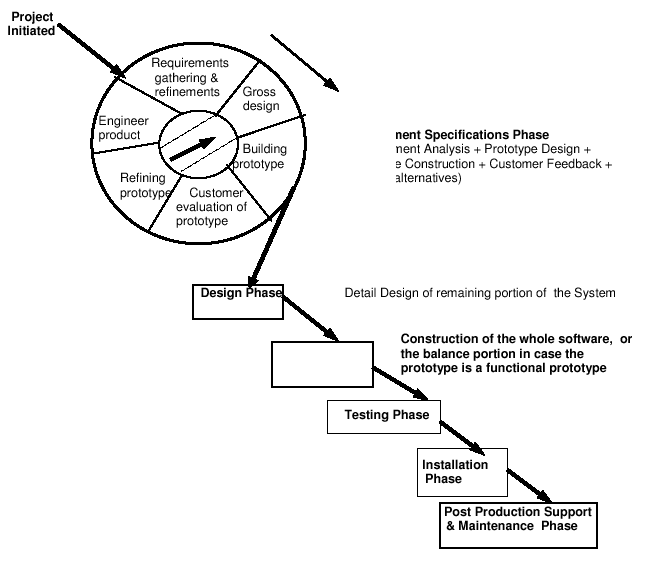
The model is applicable for projects in which all the requirements are not clear at the initial stages or there are resource constraints. In such situations the core product can be implemented with fewer staff that is available and the additional staff is added to the work on the next increment.



*Incremental model*

1. ***Prototyping Model***

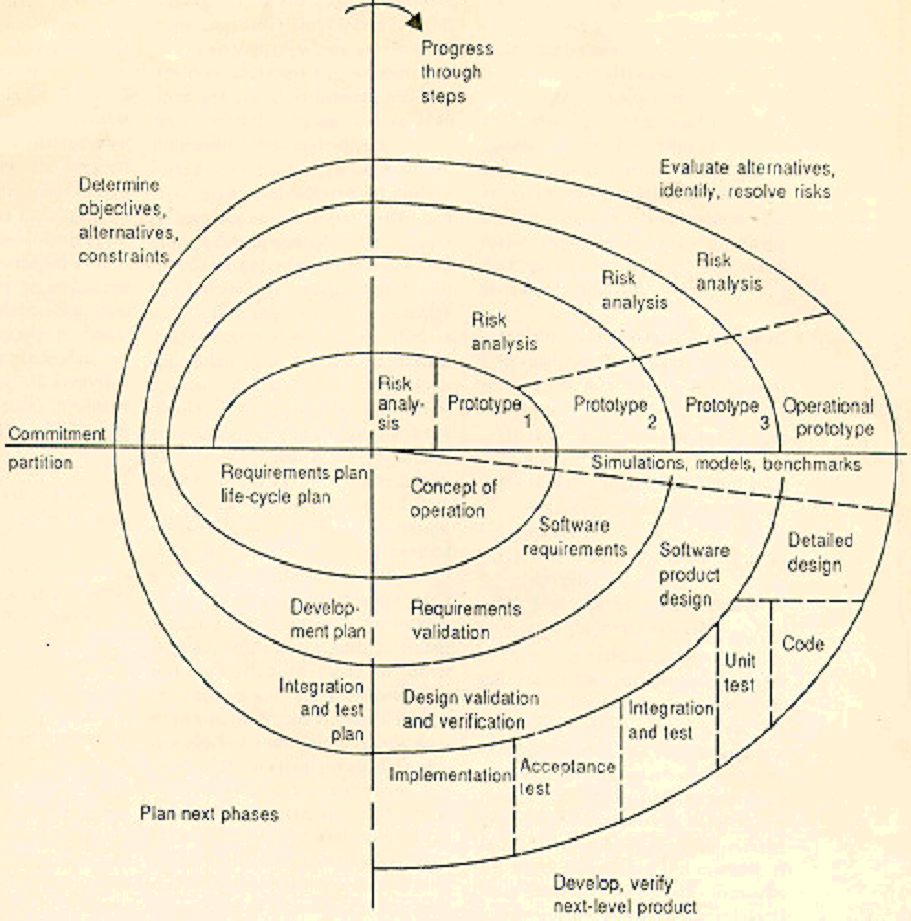
In cases where the customer defines a set of general objectives for software but does not identify detailed input, processing or output requirements or in cases where the developer may be unsure of the efficiency of an algorithm, the adaptability of an opening system, or the form that the end product may take.

In these cases, a Prototyping Model is the most appropriate.

*Prototype model*

1. ***Spiral Model***

This model couples the iterative nature of prototyping with the controlled and systematic aspects of the Waterfall model.

It is applicable to the life cycle of large-scale systems and software. Because the software evolves as the process progresses, the developer and customer better understand and react to risks at each evolutionary level.

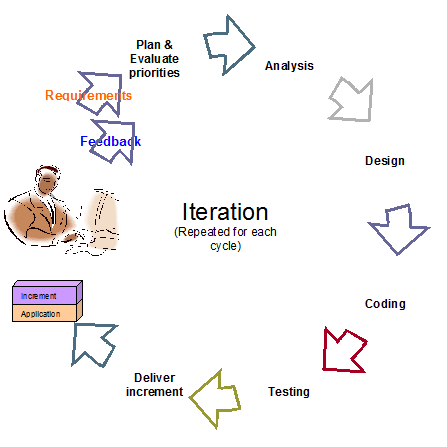
*Spiral model*

1. ***Joint Application Life cycle Model***

This model is applicable in situations where the life cycle team, customer management and user groups’ work together to build a product.

1. ***Agile Software Development Model***

**Agile software development** is a group of software development methodologies based on iterative and incremental development, where requirements and solutions evolve through collaboration between self-organizing, cross-functional teams. It promotes adaptive planning, evolutionary development and delivery, a time-boxed iterative approach, and encourages rapid and flexible response to change. It is a conceptual framework that promotes foreseen interactions throughout the development cycle.

****

*Agile model*

1. **Tailoring life cycle models**

On a project with tight deadlines where the user interface is of paramount importance, the project manager might select the waterfall approach augmented with rapid prototyping in the early phases to ensure getting the user interface right. Some of the shortcomings of the waterfall model may be mitigated with effective use of function change control and prototyping as a deliverable in the early stages.

Life cycle and methodology choices are not completely separated from those of the implementation environment. Fixed implementation choices, including hardware platform, or SOW, terms like fixed price, can affect life cycle and methodology choices.

Life cycle and methodology choices are also influenced by the amount and type of involvement from users. If intensive, accelerated requirements or analysis techniques such as joint requirements planning or joint applications design are desired, a stream-lined waterfall approach may be best. As mentioned, large and complex applications with known requirements usually follow a Waterfall Life Cycle.

However, if portions of the application involve new and untested techniques or technology, prototyping under some form of rapid application development approach is probably best. A methodology can be incorporated into any of the above-mentioned life cycles. If the problem is a unique, one-of-a-kind sort that has never been automated before, rapid application development may again fit best.

Based on the tailoring guidelines and the specific needs of the project, the operational process may be tailored as a part of the PP. Such tailored process would require SEPG approval.

1. **Phases under Operational Process**

The operational process is a tailored SDLC process based on Statement of Work and the project needs.

The process is divided into a number of phases. Usually a phase represents a stage in the evolution of software and has associated project deliverables.

Each phase has a set of deliverables, which need to be delivered at the end of the phase. A deliverable can be defined as an output or a combination of outputs realized from an activity or set of activities. These sets of deliverables can vary depending on the SOW and Project Plan. Depending upon the scope of the project, the process may contain some or all the phases.