Tailoring a Model to a Project

Process models typically define:

- Phases of development
- Inputs to each phase
- Milestones
- Outputs from each phase
- Deliverable documentation

Acquisition standards define effort and deliverable products as expected by the customer. Rarely will a software project conduct all activities and produce all documents specified in the process model or the acquisition standard. For this reason tailoring is absolutely essential. The current standards such as IEEE 12207 removes waterfall as the required model, removes any preference for functional decomposition versus object orientation, separates documents from deliverables, improves compatibilities with tools, provides clearer requirements for software reuse, supports the use of management indicators, puts an emphasis on software supportability, and improves the link with systems engineering.

An example set of documentation may include:

- System Requirements Document
- System Design Document
- Software Development Plan
- Master Test Plan
- User Manual

Why Tailor?

- Acquisition Standards are developed to address "generic" software systems development
- Standards must be comprehensive enough to cover any software project
- Each specific software program need only comply with a subset of the acquisition standards
- Tailoring is the process of "fitting" the acquisition standards to a specific project
- Tailoring reduces software development cost and schedule
- Requests for Proposals (RFPs) will often specify tailoring as a program requirement

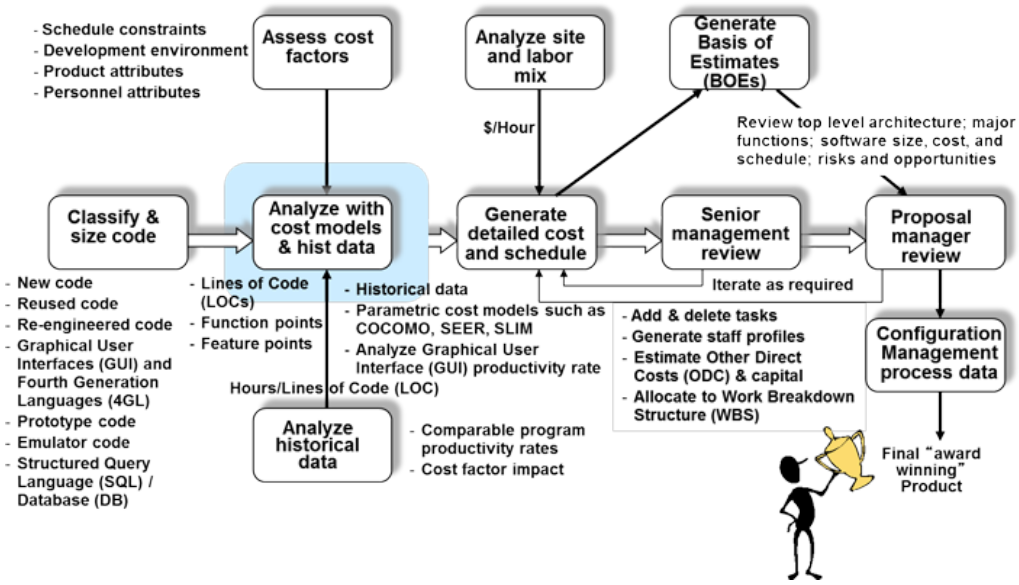
How to Tailor a Process for a Project

- Comply with customer specified requirements on tailoring
- Select the best process model for the project at hand
- Delete or modify steps in the process model
- Delete, modify, or combine contract deliverable requirements list (CDRL) items, sections, or paragraphs
- Use available computer based tools which automate the tailoring process for specific acquisition standards
- Emphasize the aspects most appropriate to the project such as:

Project Characteristic	Tailoring Example
Non-real time system	Few control aspects required in the design model
Data intensive application	Increased emphasis on information modeling
Simulator/Training application	Users Manual not required
Evolutionary development	May combine preliminary & detailed design for builds > 0
Evolutionary development	Most documentation effort is during build 0; appendix for builds > 0
Developer = Maintainer	SW Programmer's Manual replaces design docs after Build 0
"Man Rated" applications	SW must be failproof; increased emphasis on testing
Secure applications	Additional security engineering design & development activities

Software Estimation Process

The software estimation process is based on a number of inputs, the process to attain the desired outcomes. Careful planning is highly regarded. The following graphic shows a flow chart of the Software Estimation Process.



The software manager must include all the following items (often as a percentage of the software build) in the software cost estimate:

- Support to Systems Engineering Requirements Analysis and Design
- Planning and preparation to setup Software Development Environment
- **Software Requirements Analysis**
- **Software Design, Code, and Unit testing**
- **Software Integration and Test (SWIT)**
- Support to the Independent Test Teams
- Software Configuration Management
- Management, clerical, and software processes including metrics and the Software Engineering Process Group (SEPG)
- Operating System, Tools, and Commercial off the Shelf (COTS) software for the Development System, Administration, and Maintenance
- Software Licenses and COTS software for the Target System

Typically, only the Software Requirements Analysis; Software Design, Code, and Unit testing; and SWIT are included in the basis of estimates. The computer tools such as COCOMO, SLIM, and Price-S also do not include these other activities. To develop a quality estimate, you should read and analyze the Request for Proposal (RFP), Statement of Work (SOW), System Specification, Concept of Operations (CONOPS), existing software documents, any other available pertinent documents looking for:

- Functionality
- Number and types of interfaces
- Quantitative Performance Requirements including:
    - Data rates
    - Response time
    - Throughput
    - Accuracy
    - Retrieval time
    - Spare capacity
    - Update time
    - Storage capacity
- Special Requirements including:
    - Security
    - Self-diagnosing
    - Fault tolerance
    - User friendly
    - Language such as C++, Java, or HTML
    - Architecture such as DoDAF, n-tier, or J2EE
    - Reuse
    - Degree of customer involvement