SOFTWARE ESTIMATION CHECK LIST

1.1 Introduction

This document is a checklist to be used when doing estimations. After estimations are done use this checklist to ensure that there are no areas that have been left out.

1.2 Risk Factors

The major factors that increase complexity of systems:

1. Size (number of pages/screens, reports, tables, functions, features, processes)
2. Complexity based on past efforts (how familiar is the development team with the problem?)
3. Degree of structure, definition variability (how well defined is the system, is it expected to change?)
4. Are the interfaces stable? (eg., User interface, Database etc. are not changing)

1.3 Checklist

Sr.No.	Elements	Done (Y/N)
1.	Has the basis of the estimate (scope and functionality) been clearly identified. These could be any documents, meetings etc. received from the client.	
2.	Have the tasks been 'broken down' into small enough ones that span a range from ½ a day to 2 days (average)? This is contingent on the level of detail available for estimation. I.e. if you are estimating from high-level requirements, this rule can be relaxed. If you are estimating low-level tasks, it must apply.	
3.	Have the hardware resources been identified and documented. Is any additional or special h/w required for the project? Are workstation and server configurations clearly identified?	
4.	Have the people who will work on the project been identified and documented? Are any special skills required? Are they documented?	
5.	What database (if any) is to be used? Is special h/w or s/w required to support it?	
6.	Have all 3rd party components been identified. Name, version and vendor.	
7.	Has communication time been added	
8.	Has time been allocated to make releases?	

9.	Has the time for prototyping been taken care of?	
10.	Have all review / rework time been taken care of.	
11.	Has time for code optimization been allocated?	
12.	Has the total time spent been broken up for different roles and resources	
13.	Has project start up time been taken care of	
14.	Has the % for Design, Coding & Testing been calculated	
15.	Has the project management time been taken care of	
	a. Scope Management	
	b. Contract Management	
	c. Metrics Analysis	
	d. Risk identification, Mitigation and Management plan	
	e. Quality plan	
16.	Has the % of overall padding been calculated	
17.	Have you considered bought out components been included? (e.g. vbx's, ocx's etc.)	
18.	Have the global assumptions been added	
19.	Has a buy -vs- build option for control/software been evaluated (wherever necessary)	
20.	Has the testing strategy been defined?	
21.	Has the test data generation task been added?	
22.	Has the test script definition taken in to account?	
23.	Has any new learning/reading/R&D work identified and estimated?	
24.	Has time for technical architectures, once the f-spec is complete been estimated?	
25.	Has Project inertia been accounted for? Start up times, common modules. The first few modules take longer to implement. Common modules take a while to be 'spec'ed' and tested. On small projects (1-2 calendar months), entire estimates should be inflated to account for inertia.	

26.	If there is an on-site and off-site phase, have we accounted for knowledge transfers between team members? Are the people going on-site to be independent? If so, do they have the necessary skills to handle all the work?	
27.	Have we checked the units on each line item (hours, days etc.)? Are they consistent throughout the estimate?	
28.	Has the time required for seeking support from personnel outside the project (Could be within or outside the company, tech support for 3rd party controls.) been taken care of	
29.	Has the effort for configuration Management Identified?	