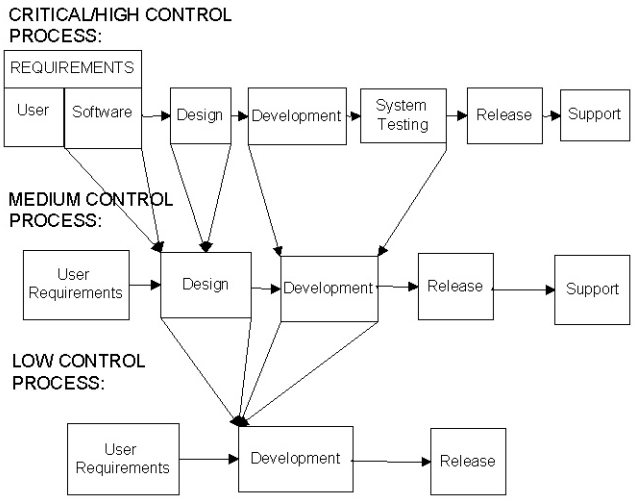
Integrated Project Planning = Characterization + Estimation + ???

Tim Kurtz
SAIC/NASA Glenn Research Center

In the beginning, there was the budget and the budget was huge. Then, there was the project and the project used the whole budget. Next came ISO 9001 and CMM processes and the budget seemed smaller. Now there is risk management and the budget seems too small. Money and planning…Spend too much on one and you will not have enough for the others. Spend too much too soon and you may not even finish. This article discusses ways to integrate and tailor development activities with estimation activities to prepare a sensible plan for a successful project with enough left over for risk management.

During the last two years, NASA Glenn Research Center received ISO 9001 certification. Meanwhile, the NASA IV&V Center funded the development of an application that ties COCOMO II estimation with tailoring of software development processes. The result is a tool, Ask Pete [1], that integrates estimation and process tailoring in a way that can be duplicated by any development organization. The tool combines two sets of questions and answers into one and eliminates the duplicates. The first set obtains information needed to estimate how long the project will take. The second set determines the level of control over the project. The Control Level, based on various factors, includes risks to personnel, equipment and the organization.



• Figure 1, Lifecycle Models

Determining the Control Level and Tailoring Development Processes

GRC identified four levels of control [2]:

- Low – typically small project with little or no integration, some testing, small development team.
- Medium – typically moderate to large project, some integration, normal testing, some damage to equipment possible, minor injury possible, little effort to meet deliveries

- High – typically large, innovative projects with part of the effort subcontracted, heavy integration and testing, damage to equipment possible, injury possible, nominal to aggressive effort to meet deliveries

- Critical – typically large or very large projects whose failure may cause death or injury, major damage to equipment and/or major embarrassment or financial impact on the organization

Requirements Phase Activities	Low	Medium	High	Critical
Authorization to proceed	☒	☒	☒	☒
Identify design/coding standards	☒	☒	☒	☒
Management Plan approval	☒	☒	☒	☒
Documented requirements	☒	☒	☒	☒
Requirements approval	☒	☒	☒	☒
Maintain Software Development Folder	☐	☒	☒	☒
Software Assurance reviews Management Plan	☐	☒	☒	☒
Implement Problem report and corrective action system	☐	☒	☒	☒
Peer review of requirements	☐	☒	☒	☒
Software Assurance reviews requirements	☐	☐	☒	☒
Peer review of plans	☐	☐	☒	☒
Implement Formal configuration management	☐	☐	☒	☒
Conduct Product Assurance Audits	☐	☐	☒	☒
Conduct Formal Reviews	☐	☐	☒	☒
Document approval of requirements and formal review	☐	☐	☒	☒
Conduct formal inspection of requirements	☐	☐	☐	☒
Customer approval of certification procedures	☐	☐	☐	☒
Conduct analyses of criticality and safety	☐	☐	☐	☒
Plan and schedule IV&V activities	☐	☐	☐	☒
Identify method for verification of safety critical functions and requirements	☐	☐	☐	☒

• Table 1, Requirements Phase Activities by Control Level

Software Documentation for:	Low	Medium	High	Critical
Management Plan:	□	□	□	□
Development Activities Plan	□	□	□	□
Verification Plan	□	□	□	□
Validation Plan	□	□	□	□
Organizational and Technical Interface Descriptions	□	□	□	□
Acquisition Activities Plan			□	□
Training Development Plan			□	□
Assurance Plan		□	□	□
Risk Management Plan		□	□	□
Configuration Management Plan		□	□	□
Delivery and Operational Transition Plan			□	□
Product Specification			□	□
Concept documentation			□	□
Requirements documentation	□	□	□	□
Design documentation	□	□	□	□
Version description		□	□	□
User's guide	□	□	□	□
Operational Procedures Manual	□	□	□	□
Procedures:				
Testing Procedures	□	□	□	□
Safety Assurance Procedures			□	□
Security and Privacy Procedures			□	□
Certification Procedures		□	□	□

Key:□□□□□□□□Required □ Optional□

• Table 2, Documentation Requirements by Control Level

Each Control Level is associated with a development lifecycle model based on the waterfall model (Figure 1). As the Control Level decreases from Critical/High to Low, the phases are combined. Development activities and documentation requirements also decrease. Table 1 lists possible development activities for the Requirements phase and identifies the ones applicable for each Control Level. Table 2 shows the documents required for each Control Level. Table 3 shows the Control Level matrix.

Project managers determine which Control Level is needed by selecting various responses from a matrix of project characteristics. Each of the factors associated with a characteristic is weighted and the weighted sum of all the characteristics determines the Control Level.

Using COCOMO II to Obtain an Estimate of Development Cost and Schedule

COCOMO II [3] uses an equation to calculate an estimate of how long it will take to develop software. The COCOMO equation is

$$PM = A \times Size^{E} \times \prod_{i=1}^{n} EM_i$$

where:
PM = nominal schedule effort in person-months
A = productivity constant (2.94 unless calibrated)
Size = code estimate (thousand Source Lines Of Code, KSLOC, or Unadjusted Function Point, UFP)
EM = Effort multipliers or cost drivers
E = Scaling exponent $= B + 0.01 \times \sum_{j=1}^{5} SF_j$
B = Scaling base exponent, 0.91 unless calibrated
SF = Scale Factors which account for economies or diseconomies of scale.

The information needed to use the equation are:

- Estimate of how much programming is required in Source Lines of Code (SLOC). If Function Points are used, they are converted to a SLOC estimate.
- Effort Multipliers
- Scale Factors

The Effort Multipliers are grouped into four areas:

1. Product Factors
    a) *Required Software Reliability (Determines critical Control*
    b) Data Base Size
    c) Product Complexity
    d) Develop for Reusability
    e) Documentation

2. Platform Factors
    a) Execution Time Constraint

b) Main Storage Constraint
c) Platform Volatility

3. Personnel Factors
    a) Analyst Capability
    b) Programmer Capability
    c) Personnel Continuity
    d) Applications Experience
    e) Platform Experience
    f) Language and Tool Experience

4. Project Factors
    a) Use of Software Tools
    b) Multisite Development
    c) Required Development Schedule

Scale Factors consist of five characteristics of the project:

- Precedentedness
- Development Flexibility
- Architecture/Risk Resolution
- Team Cohesion
- Process Maturity

Each of the Effort Multipliers and Scale Factors have a set of weighted responses. Project managers select one of the responses for each of the multipliers and factors. These values, when plugged into the equations provide the project estimate.

Integrating the Control Level Matrix with COCOMO II

Comparing the Control Level factors and the COCOMO II Effort Multipliers shows a similarity between the two. In fact, six of the 17 Effort Multipliers (italicized in the previous section) correspond to Control Level factors. The possible responses, though weighted differently for each application, characterize the same project factors. The two sets of questions and responses can be combined in a database or spreadsheet with the COCOMO II and Control Level calculations added. By eliminating the duplicates, the effort to perform both the project estimation and Control Level determination activities. An added bonus is that the outcomes can be tuned to provide the best combination of Control Level and effort estimate since changes in the responses that affect both will be automatically updated. This also eliminates the possibility that planning may result in inconsistent information used in estimating the project and determining the Control Level.

Using this integrated approach, the project manager would typically characterize a project by selecting the appropriate response for each of the 50 questions. This provides a baseline for planning the project. If the resulting control level is believed to be excessive, the initial responses for the control level questions should be evaluated. Similarly, if the estimate is too large, the COCOMO questions should be evaluated. If both the Control Level is excessive and the estimate is too large; then the questions that affect both should be looked at first.

Once the project has been tuned as much as possible, the project manager must determine if the budget allocated for the project is sufficient. If it will not cover the costs of development, implementing the necessary controls and providing resources for risk management then the following options are available

- committing additional funds
- re-scope the requirements of the project
- further tailoring of the controls
- canceling the project

If the project is re-scoped then the characterization questions should be reviewed again and changed to reflect the modified requirements. Further tailoring of the controls will require stricter surveillance to ensure the remaining controls provide the maximum benefit. Ultimately, the responses selected for each question and the results should guide the project manager in scheduling, staffing and creating the development plan.

Leveraging Project Characterization to Make Decisions

The questions used to characterize a project and determine the COCOMO II estimates and Control Level form a solid basis for integrating other planning decisions. For example, NASA uses criteria to determine the need for Independent Verification and Validation (IV&V). The criteria consists of a matrix of 17 questions with weighted responses for each. The weighted sum of the responses determines the need for IV&V or Independent Assessment. The majority of these questions are already addressed in the original 50. Only three additional questions need to be added for a project manager to incorporate the IV&V criteria into the planning process.

Additional value would be obtained by using responses to trigger risk area warnings. Many of the control level questions are based on risks to the project, organization, equipment and personnel. Adding triggers to the responses can result in a set of outputs that would identify these risk areas and provide a start for risk management and risk identification activities. Special controls could be identified and implemented when these risk areas are triggered.

• Table 3, Control Level Matrix

Factor	1	2	3	4	5	Weight
Resourcing						
Software cost annualized over life of project (including all civil servants and contractors)	$100K - $500K	$500K – 1M	$1M - $2M	$2M - $20M	$20M and up	2
Organizational Complexity						
Project development location	Own Group	Several Groups, most at GRC	More than 2 other sites	More than 3 other sites	Numerous sites	1
Customers	Self	Other in own Directorate or one customer group, low number of users	Within GRC	Within NASA	Entire Industry or multiple industries	1
Developers Software experience level	All team qualified and experienced	Most team members qualified and experienced	Half the team qualified and experienced	Few team members experienced	Experienced staff not available	1
Technical Complexity						
Test Risk	No testing required	Minimum Testing	Standard testing required	Integrated Testing	Major testing effort required (e.g. IV&V)	3
Degree of Innovation	Well proven, known to GRC	Proven with some GRC experience	Proven, but new to GRC	Partially proven with some pioneering	Pioneering	1
Software development tool availability	All software development tools already purchased/in-house/familiar	Majority of the software development tools are purchased /in-house/familiar	Software development tools must be identified and purchased and learned	Majority of software development tools must be obtained, remainder	All software development tools must be developed	2

	with	with		developed		
Interdependencies of deliverables	Simple standalone	Some Integration	Integrated	Highly integrated	Fully integrated	2
Safety Implications * (see NASA STD 8719.13A, NASA GB-1740.13 & NHB 1700.1 (V1-B))						
Potential Damage to carrier vehicle, major equipment, or system itself	No damage	Small/minor damage to equipment, or to system itself, mission still possible	Repairable/ recoverable damage to system and little or no damage to any related or surrounding systems	**Loss of system, and/ or damage to any critical surrounding systems or carrier vehicle**	**Loss of carrier vehicle (e.g. space craft, aircraft, major satellite)**	3
Potential Injury to personnel	No injury	Minor injury	Injury	**Severe injury or temporary disability**	**Loss of life or permanent disability**	3
Business Implications						
Consequence of Failure	Minor loss of Customer Confidence	Unsatisfied Customer	Damage to GRC Reputation	Damage to NASA reputation	**Significant impact to USA**	3
Schedule Pressure	No time pressure	Little effort to meet milestones	Nominal effort to meet milestones	Aggressive effort to meet milestones	Time critical	3

Critical Control

|----------Low control software ----------|
|---------- Medium control software----------|
|---------- High control software ----------|

References

1. Ask Pete Web Site, http://tkurtz.grc.nasa.gov/pete

2. NASA Glenn Research Center, *LeR-M2.6.4 Glenn Research Center Software Development Manual*

3. Boehm, et al. 2000. B. Boehm, *Software Cost Estimation with COCOMO II,* Prentice Hall PTR, Upper Saddle River, N.J., 2000

About the Author

Tim Kurtz works for SAIC as a Software Product Assurance Engineer for the Risk Management Office (RMO) at Glenn Research Center in Cleveland, Ohio. RMO performs Safety and Mission Assurance functions for NASA microgravity experiments. Prior to working for SAIC, he was responsible for overseeing software surveillance for the Defense Contract Management Command (DCMC) office at Wright-Patterson AFB in Dayton, Ohio. DCMC is responsible for providing contract administration for DoD contracts, including software quality and engineering oversight.

NASA Glenn Research Center
21000 Brookpark Rd., MS 501-4
Cleveland, Ohio 44135
Voice: (937) 254-8101
FAX: (216) 433-5270
Email: Tim.Kurtz@grc.nasa.gov
Internet: http://tkurtz.grc.nasa.gov/spa