

The Impacts of Process Structure, Communication, and Coordination on Software Project Estimation

Michael J. Douglas, Alan R. Hevner, and Rosann W. Collins

Information Systems and Decision Sciences

College of Business Administration

University of South Florida

Tampa, FL 33620

{mdouglas, ahevner, rcollins}@coba.usf.edu

1. Software Project Cost Estimation

The development and use of effective and efficient methods for estimating the costs of software development remain difficult challenges in software engineering. Many studies have shown wide discrepancies between estimates and actual costs and completion schedules of software development (e.g., [Vijayakumar 1997]). Project managers strive to make accurate estimations of cost and effort, but without good software cost estimation tools, the effectiveness of software project management is reduced. There is an urgent need for research on improved methods and tools for software project estimation.

COCOMO II [Boehm 2000] is used widely for software project estimation throughout the industry and represents the state-of-the-art in estimation methods and tools. In this paper, we present our ideas for extending the COCOMO II framework to include the selection of process structure, staff size, and staff loading on groups in the process structure. Based on the explicit modeling of costs for intra-group communication and inter-group coordination we propose a decision support system (DSS) to support a project manager's decisions on staff size, process structure, and staff loading. A brief case study illustrates use of the DSS. The paper concludes with the status of our research in progress and future directions.

2. Process Structures for Software Development

“If we can just put the right people in the appropriate positions, with well-understood responsibilities and clearly defined interfaces passing clean information from which effective decisions are made, then our project will be successful.” [Armour 2003]

A software development project's process structure is the mechanism to place the right people on the right tasks. While some projects take on very structured and disciplined process structures with formal handoffs across role boundaries, other projects may employ a process structure in which everyone works together in a single group with no formal role interfaces.

Clearly, project staff size will influence the need for high or low structure processes. A software project cost estimation model that allows managers to consider the effects of communication and coordination overheads with different staff sizes and process structures will provide important decision support information to project managers.

Although there are a plethora of possible process structures, our research has identified three basic structures that are commonly used in software development. Figures 1 to 3 show the One-Tier, Two-Tier, and Three-Tier process structures that a manager can select to best fit her specific project. The double-headed arrow at the top of each figure represents the comparison of the delivered system against the desired system. The single-headed arcs within the process indicate a coordination path in which verbal and written interactions and handoffs of system artifacts (e.g., specifications, designs, test cases) occur.

The one-tier model combines all the project staff and development roles into a single team that is responsible for all systems development. The two-tier process structure divides the staff into three development groups – Requirements/Design, Implementation/Unit Testing, and Integration Testing/Customer Acceptance. Finally, the three-tier process structure is the most complex with five groups employed in the software development life cycle. Our research focuses on the cost trade-offs between having staff in fewer groups with increased intra-group communication and having a larger number of groups with smaller staff sizes but with increased inter-group coordination.

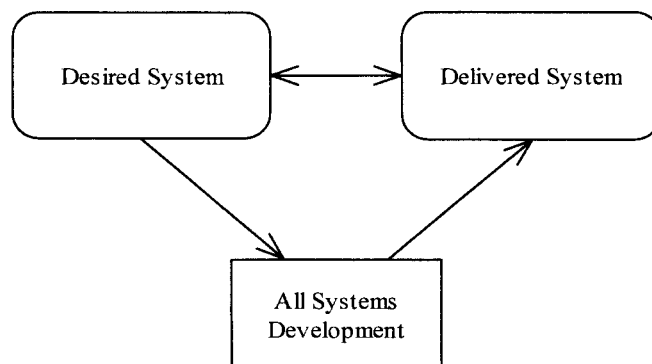


Figure 1: One-Tier Process Structure

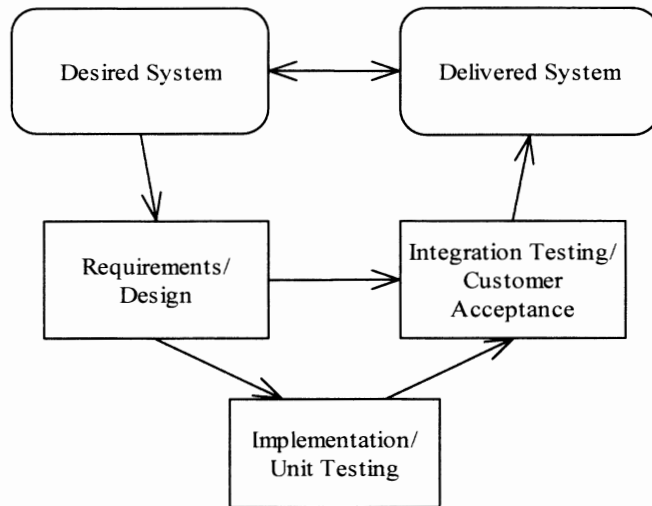


Figure 2: Two-Tier Process Structure

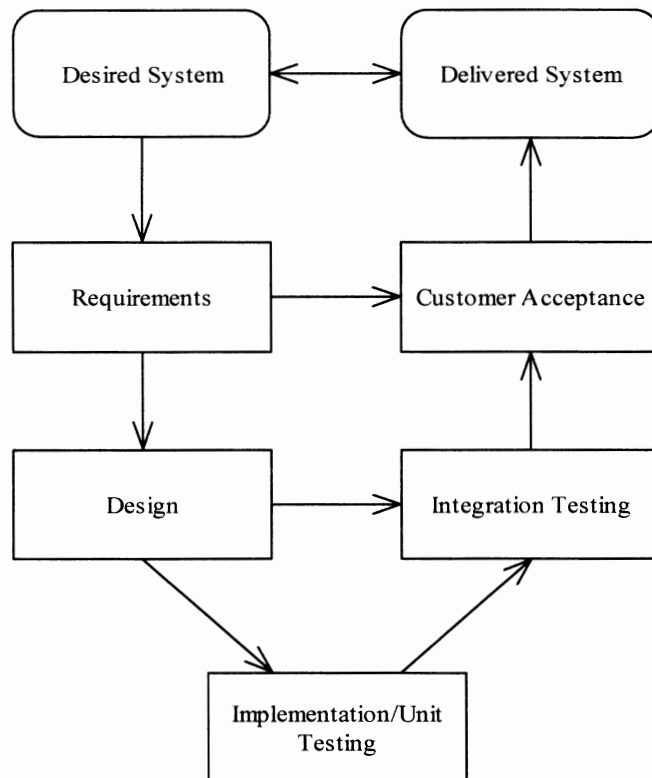


Figure 3: Three-Tier Process Structure

3. Communication and Coordination Costs

Most software cost estimation models view staff size as a dependent variable that is a function of effort and schedule. Little is made of the well-known fact that size impacts

communication costs both within and between software development groups [Brooks 1975]. Abdel-Hamid and Madnick [1991] model the impact of development group size on software development as *communication overhead*, due to the additional effort needed to communicate as groups become larger. Communication overhead costs are unavoidable, but can be managed by keeping group sizes small.

Inter-group coordination is a critical CMM Level 3 key process area [SEI 1995] that attempts to control communication and coordination by effectively and efficiently planning and managing the technical working interfaces and interactions between software development groups. Malone and Crowston [1994] define *coordination* as the management of dependencies between activities. In the software development context these dependencies include shared tasks, resources, goals, constraints, and designs.

There is a lack of techniques available to project managers for planning and managing this inter-group coordination. Especially problematic is how to include the effects of coordination on software cost estimation. COCOMO II includes a scale driver to capture the costs (effort and schedule) incurred in projects that require more coordination between development groups. This scale driver, *team cohesion*, measures

“the sources of project turbulence and entropy because of difficulties in synchronizing the project’s stakeholders: users, customers, developers, maintainers, interfacers, others. These difficulties may arise from differences in stakeholder objectives and cultures; difficulties in reconciling objectives; and stakeholders’ lack of experience and familiarity in operating as a team.” [Boehm 2000]

However, this single coefficient does not allow project managers to directly model differing coordination costs as they consider various process structures and staff assignment to groups.

In addition to team cohesion, COCOMO II introduces an effort multiplier for *schedule compression*. Required Development Schedule measures the schedule constraint imposed on the project team developing the software. The schedule compression effort multiplier is an initial attempt to model the changes in effort and staff required when the schedule changes.

The schedule compression effort multiplier does not consider coordination or communication costs. Table 1 shows seven different projects and the COCOMO II estimates for effort, schedule, and staff. Note that when the schedule is reduced to 75% of original, the effort and staff increase accordingly, but the staff increases about 90% in all cases. This implies that adding 1 additional person in the 2000 lines of code project is similar to adding 58 people in the 512,000 lines of

code project. We believe that if communication and coordination costs are included, the effort and staff estimate would increase much more than is estimated by COCOMO II.

Table 1: COCOMO II Schedule Reduction Multiplier

Lines of Code	Original Estimate			75% of Original			Increase in Staff
	Effort	Schedule	Staff	Effort	Schedule	Staff	
2000	6.7	6.7	1	9.581	5.025	1.9	91%
8000	31	10.9	2.8	44.33	8.175	5.4	94%
16000	66.4	13.9	4.8	94.952	10.425	9.1	90%
32000	142.2	17.8	8	203.346	13.35	15.2	90%
64000	304.8	22.6	13.5	435.864	16.95	25.7	90%
128000	653.2	28	22.7	934.076	21	44.5	96%
512000	3000	46.8	64.1	4290	35.1	122.2	91%

A COCOMO II extension, CONstructive RAD schedule estimation MOdel (CORADMO), has been developed to address the shortcomings in COCOMO II when alternative processes are used. CORADMO introduces five new drivers that adjust both effort and schedule. But CORADMO still does not model the impact of having different staff sizes, therefore ignoring important communication and coordination costs. The model described in our research is more parsimonious than CORADMO in that no additional cost drivers are needed to model the impacts of change. Rather the coordination and communication costs are calculated directly for the staffing assignments. As the staffing changes, there is an immediate change to both effort and schedule.

4. A Decision Support System for Software Project Estimation

Figure 4 illustrates our approach for this research-in-progress. Inputs to our decision support system include data on specific software project characteristics and the COCOMO II outputs on project effort, schedule, and the work breakdown structure. Then we analyze the impacts of the process structure and team size as they effect communication and coordination costs in the project. The project manager will use the DSS to find the most effective mix of staff size, process structure, and staff loading in development groups.

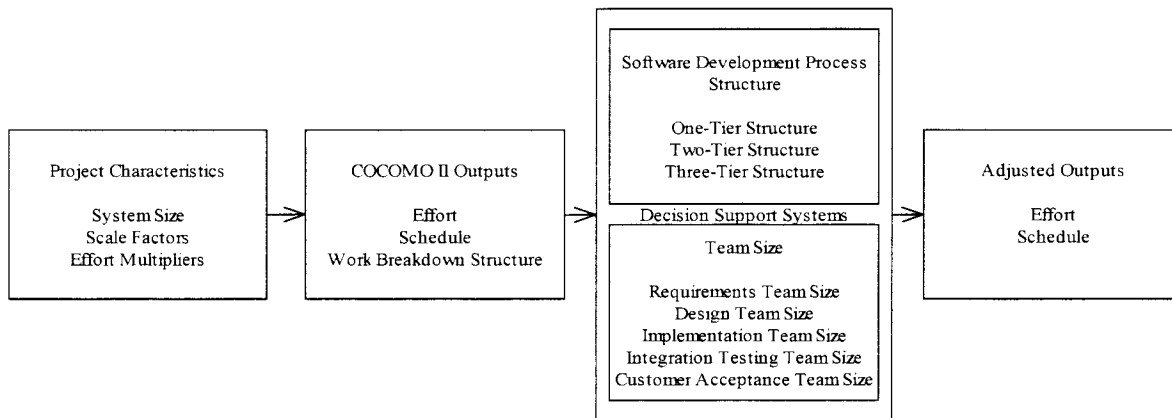


Figure 4: Project Estimation Approach using the Extended Models in a DSS

We have developed mathematical models to calculate communication and coordination costs for each of the three process structures based upon staff size and the assignment of staff to development groups. The DSS user can vary these inputs to find the best mix for her project and to minimize development costs and schedules. Here, due to space limitations, we will provide a summary of the model for the two-tier process structure (refer to Figure 2).

For each of the three development groups in the two-tier process structure (i.e., Requirements/Design (RD), Implementation/Unit Testing (IT), and Integration Testing/Customer Acceptance (TC)), the communication overhead is a function of staff allocated to that group, s . The number of communication paths in each group is $(s * (s - 1))/2$. This value represents the number of communication paths in the network connecting any two staff members. Communication overhead for the group is then defined as $d * (s * (s - 1))/2$, where the overhead coefficient $d = 0.001248$ [Abdel-Hamid and Madnick 1991].

Coordination costs are calculated along the connecting arcs between the Desired System and the Requirements/Design group (C_{DR}), Requirements/Design and Implementation/Unit Testing groups (C_{RI}), Requirements/Design and Integration Testing/Customer Acceptance groups (C_{RT}), Implementation/Unit Testing and Integration Testing/Customer Acceptance groups (C_{IT}), and the Integration Testing/Customer Acceptance group and the Delivered System (C_{TS}). The details of these equations are rather extensive and are presented in full in [Douglas 2003].

The project manager would use the estimation DSS with different staff levels and different staff loadings onto the three development groups to find the mix with the least costs and the shortest schedule for the two-tier process structure. Staffing constraints would include a

minimum and maximum staff level and the allocation of at least one developer to each group. This procedure could be automated to consider all feasible staff levels with all combinations of staff loading into the three development groups for each staff level. This would result in finding the best staffing profile on the two-tier process structure for the needs of the project. Comparing costs and schedules across the three process structures would also result in the selection of the best process structure for the project.

5. Case Study

We demonstrate our new software cost estimation approach on a simple software development project. Consider a project where the estimated software size is 40 KSLOC. Default COCOMO II effort multipliers and scale factors are applied to produce an estimated total effort of 169.9 person-months. However, COCOMO II by default does not include the planning and requirements phase of development. The effort and schedule required to build requirements must be added to COCOMO II estimate. To conduct the planning and requirements phase, the effort estimate is increased by 6-8%. Making straightforward assumptions, the new effort from COCOMO II including requirements is 181.8 person-months. COCOMO II estimates that 9.5 people are required and the project will take 19.2 months.

Based on these project data and the COCOMO II outputs, we analyzed project effort and schedule from the three process structures under several staffing profiles. The results are shown in Table 2.

One-Tier Process Structure Estimate

The estimate for the one-tier process structure is 192 person-months. The difference between the COCOMO II estimate and the one-tier estimate is due to communication overhead. COCOMO II's 9.5 staff estimate is rounded to 10 people to result in a schedule of 19.2 months. However, if the staff level is increased to 13 people, the effort increases to 199.5 person-months, but the schedule is reduced to 15.3 months. Compared to the schedule reduction computed by COCOMO II, the one-tier process structure estimate shows the need for more effort and duration with the additional consideration of communication overhead.

Table 2: Case Study Results for a 40 KSLOC Software Development Project

Estimation Model	Effort in Person-months	Staff Size	Schedule in Months	Project Staff Loading
COCOMO II	181.8	9.5	19.2	N/A
Estimate 1 with One-Tier Process	192.0	10	19.2	One development team
Estimate 2 with One-Tier Process	199.5	13	15.3	One development team
Estimate 1 with Two-Tier Process	185.0	10	31	3 in Requirements/Design, 4 in Implementation/Unit Testing, 3 in Integration Testing/Customer Acceptance
Estimate 2 with Two-Tier Process	198.0	21	16	6 in Requirements/Design, 9 in Implementation/Unit Testing, 6 in Integration Testing/Customer Acceptance
Estimate 1 with Three-Tier Process	184.0	10	52	2 in Requirements, 2 in Design, 2 in Implementation/Unit Testing, 2 in Integration Testing, 2 in Customer Acceptance
Estimate 2 with Three-Tier Process	199.0	24	16	3 in Requirements, 5 in Design, 9 in Implementation/Unit Testing, 6 in Integration Testing, 1 in Customer Acceptance

Two-Tier Process Structure Estimate

The estimate for the two-tier process structure is 185 person-months. If ten staff are used; three staff in the RD group, four staff in the IT group, and three staff in the TC group; the calculated schedule is 31 months. If the staffing loads are increased to six people in the RD group, nine in the IT group, and six in the TC group for a total of 21 staff, the total effort increases to 198 person-months, but the schedule is reduced to 16 months. This type of trade-off analysis is supported effectively in the new cost estimation model.

Three-Tier Process Structure Estimate

The estimate for the three-tier process structure with ten staff is 184 person-months. Two staff are in the Requirements group, two are in the Design group, two are in the Implementation/Unit Testing group, two are in the Integration Testing group, and two are in the

Customer Acceptance group. The calculated schedule is 52 months. If the project staff is increased to 24 and allocated to groups as shown in Table 2, the total effort increases to 199 person-months, but the schedule is reduced to 16 months.

Case Study Summary

Studying the data in Table 2, we can see the trade-offs inherent in this simple case study for decisions on process structure, staff size, and staff loading in development groups. The shortest schedule is produced by the one-tier process structure with 13 staff. However, having a single development group may not be feasible. For example, the software development organization may require software validation by a separate testing group. Another effective alternative is the two-tier process structure with 21 staff and a 16 month schedule.

6. Research Directions

This research-in-progress has the potential to make contributions in several important areas. We are developing a comprehensive software estimation model that integrates the costs of intra-group communication and inter-group coordination. In addition, the selection of the most effective process structure during project estimation is a new contribution. We have developed a functioning prototype of a decision support system (DSS) that implements the new estimation models. The prototype implementation consists of eight thousand lines of C# code in Microsoft Visual Studio .NET 2003. The developed application has user friendly interfaces that can be used effectively to select staff size, staff loading, and process structure for a given project. A sample input screen from the DSS prototype is shown in Figure 5.

The new estimation model and prototype DSS are being evaluated on project data from secondary sources. Initial results show clear improvements in project estimation via explicit consideration of communication and coordination costs. Empirical validation of the model will be performed with real IT project managers. Introducing the project managers to the DSS tool in real software development organizations will provide valuable insight into the model's validity and generalizability.

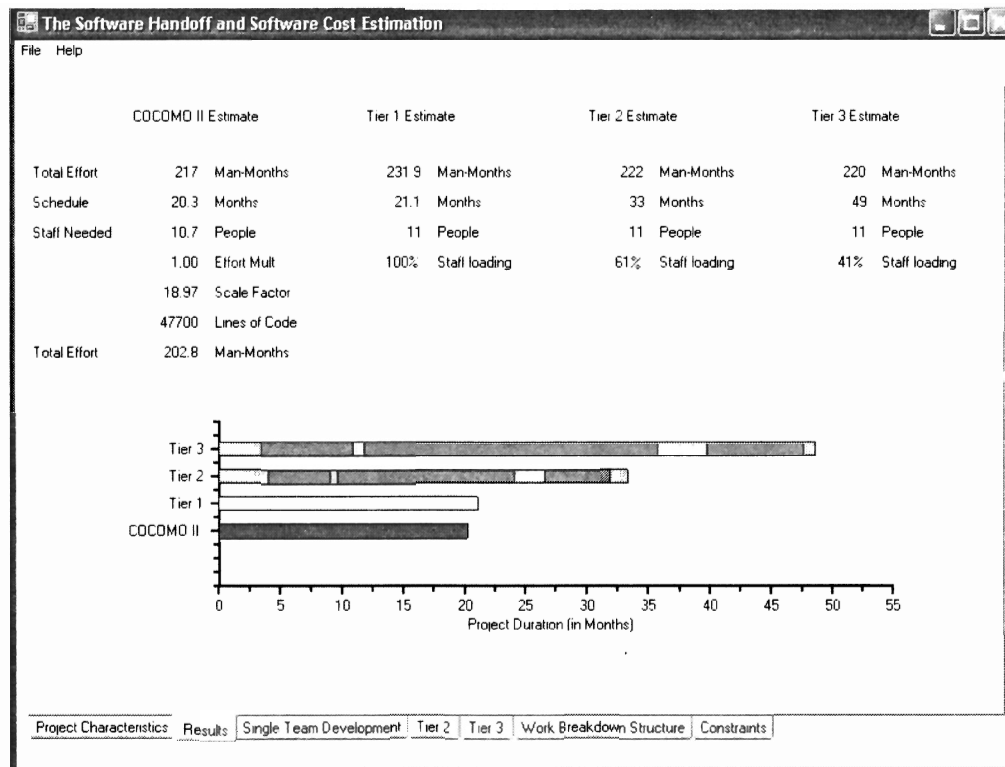


Figure 5: Screenshot of DSS Tool - Simulation Results

References:

- Abdel-Hamid, T. K. and S. E. Madnick (1991). *Software Project Dynamics: An Integrated Approach*. Englewood Cliffs, New Jersey, Prentice Hall.
- Armour, P. (2003). "In the Zone: The Need for Flexible Roles." *Communications of the ACM* 46(5): 15-19.
- Boehm, B. W. (2000). *Software Cost Estimation with COCOMO II*. Upper Saddle River, N.J., Prentice Hall PTR.
- Brooks, F. P. (1975). *The Mythical Man-Month*. Reading, MA, Addison-Wesley.
- Douglas, M. (2003). *Communication and Coordination Costs in Software Project Estimation*, Doctoral Dissertation (in progress), Information Systems and Decision Sciences, University of South Florida.
- Malone, T. W. and K. Crowston (1994). "The Interdisciplinary Study of Coordination." *ACM Computing Surveys* 26(1): 87-119.
- Software Engineering Institute (1995). *The Capability Maturity Model: Guidelines for Improving the Software Process*. Reading, Mass, Addison-Wesley.
- Vijayakumar, S. (1997). "Use of Historical Data in Software Cost Estimation." *Computing & Control Engineering Journal* 8(3): 113-119.