A Model for Estimating Agile Project Process and Schedule Acceleration

Dan Ingold, Barry Boehm, Supannika Koolmanojwong, Jo Ann Lane
Center for Systems and Software Engineering
University of Southern California
Los Angeles, USA
{dingold,boehm,koolmano,jolane}@usc.edu

ABSTRACT
Accelerating development schedules is increasingly important in a competitive world. Reduced time-to-market is a key response to competitive threats in the commercial sphere, and rapid response in deploying military systems may save lives in a geopolitical environment characterized by rapidly emerging and ever-changing physical threats. Agile/lean development methodologies show promise in providing the desired schedule acceleration, but it can be difficult for planners to determine the effects of these factors on schedule duration, and to make appropriate choices to optimize project performance. The Constructive Rapid Application Development Model (CORADMO) attempts to quantify the effects of key schedule drivers, and thus enable planners to estimate the relative schedule that will result from varying these parameters.

Categories and Subject Descriptors
D.2.8 [**Metrics**]: Complexity measures, Performance measures, Process metrics, Product metrics, Software science
D.2.9 [**Management**]: Copyrights, Cost estimation, Life cycle, Productivity, Programming teams, Software configuration management, Software process models (e.g., CMM, ISO, PSP), Software quality assurance (SQA), Time estimation

General Terms
Management, Measurement, Economics, Human Factors

Keywords
Agile, CORADMO, estimation, lean, modeling, rapid development.

1. INTRODUCTION
Accelerating development schedules is increasingly important in a competitive world. Reduced time-to-market is a key response to competitive threats in the commercial sphere, and rapid response in deploying military systems may save lives and deter adversaries in a geopolitical environment characterized by rapidly emerging and ever-changing physical threats. Agile/lean development methodologies show promise in providing the desired schedule acceleration, within certain problem domains and organizational characteristics [1]. However, we have found that many projects experience slower schedules by jumping into agile methods without awareness of their pitfalls. These include making easiest-first, hard-to-refactor architectural commitments, choosing unscalable or incompatible off-the shelf products, accepting unsuitable on-site customer representatives, teambuilding insufficiently, or assuming low personnel turnover.

The Constructive Rapid Application Development Model (CORADMO) attempts to quantify both the positive and the negative effects of key schedule drivers, and thus enable planners to estimate the relative schedule that will result from varying these parameters. CORADMO is a derivative of the revised Constructive Cost Model (COCOMO II) [6], which was calibrated against larger projects that were typically optimized to reduce cost. In contrast, the goal of projects using agile/lean techniques is often to compress schedule. Further, COCOMO II generates unreasonably high duration estimates for projects of fewer than two person-years of effort, and does not explicitly consider rapid development techniques.

The original CORADMO described in Chapter 5 of [6] operated as a post-processor to adjust the cost and schedule estimates coming from the standard cost-optimized COCOMO II estimates. The COCOMO II schedule model estimates the project duration D in calendar months as 3.67 times roughly the cube root of the estimated effort in person months (PM). Thus, a 27 PM effort would result in an estimated duration of 3.67 * 3 = 11 calendar months, and an average staffing level of 27/11 = 2.45 people. Such a small team minimizes communications overhead and optimizes effort, but 11 months is excessively long for a competitive or a much-needed product.

As we were calibrating COCOMO II, we were also seeing time-competitive early-agile projects completing 27-PM projects in 5 months by putting an average of 5.4 people on the project. In some well-jelled, domain-experienced Rapid Application Development (RAD) organizations, they could often put 9 people on a 27-PM project and finish in 3 months.

This motivated the development of CORADMO. Its COCOMO II post-processor used a nominal square-root relationship between PM and D, completing a 27-PM project in 5.2 months with an average team size of 5.2 people. It then adjusted the nominal schedule and the originally-estimated effort by applying some schedule acceleration-deceleration factors such as component reuse, asset prepositioning, process streamlining, collaboration technology, early architecture and risk resolution, and RAD personnel-team capability.

ICSSP'13, May 18-19, 2013, San Francisco, USA.

Table 1. Schedule Accelerators and Rating Factors

Accelerators/Ratings	Very Low	Low	Nominal	High	Very High	Extra High
Product Factors	**1.09**	**1.05**	**1.0**	**0.96**	**0.92**	**0.87**
Simplicity	Extremely complex	Highly complex	Mod. complex	Moderately simple	Highly simple	Extremely simple
Element Reuse	None (0%)	Minimal (15%)	Some (30%)	Moderate (50%)	Considerate (70%)	Extensive (90%)
Low-Priority Deferrals	Never	Rarely	Sometimes	Often	Usually	Anytime
Models vs Documents	None (0%)	Minimal (15%)	Some (30%)	Moderate (50%)	Considerate (70%)	Extensive (90%)
Key Technology Maturity	>0 TRL 1,2 or >1 TRL 3	1 TRL 3 or > 1 TRL 4	1 TRL 4 or > 2 TRL 5	1-2 TRL 5 or >2 TRL 6	1-2 TRL 6	All > TRL 7
Process Factors	**1.09**	**1.05**	**1.0**	**0.96**	**0.92**	**0.87**
Concurrent Operational Concept, Requirements, Architecture, V&V	Highly sequential	Mostly sequential	2 artifacts mostly concurrent	3 artifacts mostly concurrent	All artifacts mostly concurrent	Fully concurrent
Process Streamlining	Heavily bureaucratic	Largely bureaucratic	Conservative bureaucratic	Moderate streamline	Mostly streamlined	Fully streamlined
General SE tool support CIM (Coverage, Integration, Maturity)	Simple tools, weak integration	Minimal CIM	Some CIM	Moderate CIM	Considerable CIM	Extensive CIM
Project Factors	**1.08**	**1.04**	**1.0**	**0.96**	**0.93**	**0.9**
Project size (peak # of personnel)	Over 300	Over 100	Over 30	Over 10	Over 3	≤ 3
Collaboration support	Globally distributed weak comm. , data sharing	Nationally distributed, some sharing	Regionally distributed, moderate sharing	Metro-area distributed, good sharing	Simple campus, strong sharing	Largely collocated, Very strong sharing
Single-domain MMPTs (Models, Methods, Processes, Tools)	Simple MMPTs, weak integration	Minimal CIM	Some CIM	Moderate CIM	Considerable CIM	Extensive CIM
Multi-domain MMPTs	Simple; weak integration	Minimal CIM	Some CIM or not needed	Moderate CIM	Considerable CIM	Extensive CIM
People Factors	**1.13**	**1.06**	**1.0**	**0.94**	**0.89**	**0.84**
General SE KSAs (Knowledge, Skills, Agility)	Weak KSAs	Some KSAs	Moderate KSAs	Good KSAs	Strong KSAs	Very strong KSAs
Single-Domain KSAs	Weak	Some	Moderate	Good	Strong	Very strong
Multi-Domain KSAs	Weak	Some	Moderate or not needed	Good	Strong	Very strong
Team Compatibility	Very difficult interactions	Some difficult interactions	Basically cooperative interactions	Largely cooperative	Highly cooperative	Seamless interactions
Risk Acceptance Factor	**1.13**	**1.06**	**1.0**	**0.94**	**0.89**	**0.84**
	Highly risk-averse	Partly risk-averse	Balanced risk aversion, acceptance	Moderately risk-accepting	Considerably risk-accepting	Strongly risk-accepting

The effort and schedule multipliers for these factors were determined such that a well-jelled, domain-experienced RAD project would be estimated as 9 people on a 27-PM project for 3 months, but that a misguided RAD project would take more like 40 PM and 9 months. Unfortunately, in the pre-2000 time frame, we did not have a critical mass of data to calibrate such a model. Recently, though, we have been participating in some research on expediting systems and software engineering via lean and agile methods, that led to a expanded set of product, process, project, people, and risk factors that account for relative schedule acceleration and deceleration [15]. These looked like a better basis for developing a revised CORADMO set of schedule drivers and rating scales, and are discussed next.

2. METHOD

The Systems Engineering Research Center (SERC) Research Task RT-34, "Expedited Systems Engineering for Rapid Capability and Urgent Needs," studied ways that systems engineering might be expedited, particularly within the aerospace/defense community. Through industry and government contacts, the study identified candidate firms and agencies that had a history of successfully compressing the development time of projects. In a series of onsite visits and in-depth follow-up interviews, the study identified a set of key factors [12] that, in combination with factors derived in the earlier CORADMO research [6], could be used to model RAD projects' schedule acceleration (Table 1). These factors fall in the categories of product, process, project, people, and risk.

Product factors describe the nature of the system to be developed across five sub-factors: simplicity, ability to reuse existing elements, ability to defer lower-priority requirements, degree that models (prototypes, simulations, etc.) can be substituted for written documentation, and maturity of the component technologies.

Process factors characterize the development methodology using three sub-factors: concurrency of artifact development (operational concept, requirements, code, etc.); degree of process streamlining; and the coverage, integration, and maturity (CIM) of tools used to support the development process[3].

Further detail on the process factors is provided for the process community. Use of concurrent vs. sequential processes has been consistently observed to accelerate schedule in the use of such methods as the spiral model [5], the Rational Unified Process [14], and agile methods [4], although with the need of mechanisms to synchronize and stabilize the concurrently-developed elements via buffered phases in the Microsoft approach [10] and evidence-based milestones in the Incremental Commitment Spiral Model [8].

The process streamlining subfactor drew on the Development Process Reengineering and Streamlining factor in the original CORADMO model [6], which primarily addressed removal of bureaucratic and procedural delays. The SERC study also identified other key contributors such as Kaizen performer-identified streamlining [11], [18] lean approaches such as Kanban[1]. The effect of tool support was found in the COCOMO database analysis [5] to be due about 50% for tool coverage and 25% each for toolset integration and maturity.

Project factors span four sub-factors describing execution of the development effort: project staff size; degree and nature of team collaboration; CIM of the single-domain models, methods, processes, and tools (MMPTs) employed; and CIM of the multi-domain MMPTs used, where required.

People factors describe the project staff using four sub-factors: general knowledge, skills, and agility (or, ability to thrive with the more concurrent nature of the agile/lean process) [9]; KSAs specific to the primary problem domain; KSAs spanning multiple problem domains, where needed; and team compatibility [7].

In addition, we used references [1], [2], [13], [16], [17], [19], [20] on rapid development, rapid fielding, and schedule acceleration in developing the rating scales for the product, process, project, and people factors. Finally, the Risk factor characterizes the project stakeholders' willingness to accept rapid but imperfect solutions [9]. Stakeholders may range from highly risk-averse, to strongly risk-accepting.

As discussed in the Introduction, a good baseline estimate for schedule reduction through rapid development methods has been found to be proportional to the square root of effort. CORADMO estimates duration (D) as the product of multipliers associated with the rating factors in Table 1 (Fi) and the nominal agile duration of the square root of baseline effort in person-months (PM),

$$D = \prod F_i \sqrt{PM}. \qquad (1)$$

It also has multipliers that adjust the original effort estimate to reflect the effect of RAD on effort.

As seen in Table 1, each of the proposed factors is rated along a 6-value Likert scale ranging from Very Low to Extra High, where factors rating lower in the scale tend to extend the schedule, and those rating higher to reduce it. Initial values of the schedule acceleration multipliers were chosen to span a relatively small range of duration expansion and reduction, pending model calibration. Our evaluation of rapid development projects in this research, however, suggests that people factors [12] and risk tolerance [9]—which tracks willingness to accept some product imperfections to improve schedule—have greater effects than the other factors, which is reflected in the greater span of their associated schedule multipliers.

We evaluated the CORADMO model against a 12-project

Table 2. Commercial Projects rating factors and analysis

Application Type	Technologies	Person Months	Duration (Months)	Duration / √PM	Product	Process	Project	People	Risk	Multi-plier	Error %
Insurance agency system	HTML/VB	34.94	3.82	0.65	VH	VH	XH	VH	N	0.68	5%
Scientific/engineering	C++	18.66	3.72	0.86	L	VH	VH	VH	N	0.80	-7%
Compliance - expert	HTML/VB	17.89	3.36	0.79	VH	VH	XH	VH	N	0.68	-15%
Barter exchange	SQL/VB/ HTML	112.58	9.54	0.90	VH	H	H	VH	N	0.75	-16%
Options exchange site	HTML/SQL	13.94	2.67	0.72	VH	VH	XH	VH	N	0.68	-5%
Commercial HMI	C++	205.27	13.81	0.96	L	N	N	VH	N	0.93	-3%
Options exchange site	HTML	42.41	4.48	0.69	VH	VH	XH	VH	N	0.68	-1%
Time and billing	C++/VB	26.87	4.80	0.93	L	VH	VH	VH	N	0.80	-14%
Hybrid Web/client-server	VB/HTML	70.93	8.62	1.02	L	N	VH	VH	N	0.87	-15%
ASP	HTML/VB/SQL	9.79	1.39	0.44	VH	VH	XH	VH	N	0.68	53%
On-line billing/tracking	VB/HTML	17.20	2.70	0.65	VH	VH	XH	VH	N	0.68	4%
Palm email client	C/HTML	4.53	1.45	0.68	N	VH	VH	VH	N	0.76	12%

dataset of diverse but single-company projects executed by a Midwest software development firm that used agile practices, and that supplemented those practices with systems engineering (SE) processes distinguishing their approach from typical BigDesignUpFront-avoiding agile projects. These SE practices included detailed business process analyses, Delphi estimates of software testing effort, risk-based situation audits, and componentized architectures, among others. Use of systematic SE processes by the firm was considered to make these projects more comparable to the SE practices applied in the more complex aerospace/defense projects from which the factors in Table 1 were derived.

Note that all of the subfactors in a category are assumed to have the same set of schedule acceleration multipliers. Also, the multipliers were assumed to follow a geometric progression, so that the effect of each subfactor was determined by a single parameter, the acceleration range AR: the ratio between the lowest and highest multipliers. For example, the AR for the Product and Process factors is 0.87/1.09= 0.8, while the AR for People factors is 0.84/1.13 = 0.74. This meant that we were fitting the 12 project data points to 5 parameters (actually 4, since the Risk factor was nominal or 1.0 in all cases). Fitting different ARs for the 17 CORADMO subelements to the 12 project data points would have led to an overdetermined and relatively meaningless solution.

The model was also applied to a case study derived from observations of aerospace and commercial firms that have been affiliated with the Center for System and Software Engineering (CSSE) at the University of Southern California (USC). While this case study is not directly traceable to any single firm, it is representative of the range of projects and capabilities that we have seen in real firms. This application of the model allowed us to characterize the types of schedule effects that one might expect to see by varying the factors, which we plan to validate against actual projects in future research.

3. RESULTS

3.1 Calibration to Commercial Rapid Development Projects

Table 2 presents a dataset of twelve commercial rapid development projects ranging in size from 10 KLOC (thousands of source lines of code) to 400 KLOC, of varying complexity and technology. We rated these projects against the Product, Process, Project, People and Risk factors discussed above to compute the product of the schedule acceleration factors, and to compare them against the D/√PM calculated from the reported project duration and effort.

Factor ratings were selected based upon the reported characteristics of each project, and of the firm as a whole. The projects that employed C++ technologies received Low (L) Product Simplicity ratings as compared with the other HTML-Visual Basic projects and the described product complexity; the "Hybrid Web/Client Server" Product was rated Low (L) due to its high degree of innovation and requirements churn. For the Process factor, most projects used a highly concurrent development process, resulting in a Very High (VH) rating; some projects reported using more complex mixes of technology that suggest less concurrency, and therefore received lower ratings. Reported variation in project staff sizes is the primary reason for the varying Project ratings. The staff was described as being very capable and senior-level, and so the People factor rated at Very High (VH) across the board. Similarly, the firm documented a consistent and rigorous development approach, balancing good engineering against development speed, and hence were all rated at Nominal (N) Risk acceptance.

The product of the selected rating factors is shown in the Multiplier column of Table 2, and should be compared against the value in the Duration/√PM column, calculated from actual duration and effort. The close correspondence of these values in the Error column suggests that the acceleration-deceleration factors are appropriate, although additional work remains in that the calculated factors suggest greater schedule acceleration that was actually observed. The "ASP" project is an outlier that we cannot explain from the data reported. It had a team of 7 people produce a 16,875 SLOC product in just 1.39 months.

3.2 Agile SE Adoption Case Study

Table 3. As-Is Rating Factors

Accelerators/Ratings	VL	L	N	H	VH	XH
Product Factors	**1.09**	**1.05**	**1.0**	**0.96**	**0.92**	**0.87**
Simplicity			X			
Element Reuse	X					
Low-Priority Deferrals	X					
Models vs Documents		X				
Key Technology Maturity					X	
Process Factors	**1.09**	**1.05**	**1.0**	**0.96**	**0.92**	**0.87**
Concurrent Operational Concept, Requirements, Architecture, V&V	X					
Process Streamlining		X				
General SE tool support CIM (Coverage, Integration, Maturity)				X		
Project Factors	**1.08**	**1.04**	**1.0**	**0.96**	**0.93**	**0.9**
Project size (peak # of personnel)				X		
Collaboration support				X		
Single-domain MMPTs (Models, Methods, Processes, Tools)				X		
Multi-domain MMPTs		X				
People Factors	**1.13**	**1.06**	**1.0**	**0.94**	**0.89**	**0.84**
General SE KSAs (Knowledge, Skills, Agility)				X		
Single-Domain KSAs				X		
Multi-Domain KSAs		X				
Team Compatibility				X		
Risk Acceptance Factor	**1.13**	**1.06**	**1.0**	**0.94**	**0.89**	**0.84**
			X			

This case study illustrates the use of the revised CORADMO model in explaining the differences in schedule acceleration for various project approaches. The baseline situation for the case study is a hypothetical company division specializing in performing early-SE activities for defense applications in a diversified company, generally involving teams of roughly 20 systems engineers (SEs). The division has traditionally applied a sequential waterfall or "Vee" model to define a system's operational concept and requirements, and then developed a system architecture that satisfies those requirements. Defense

Table 4. Initial To-Be Rating Factors

(Initial Post-Adoption Rating Factors)

Accelerators/Ratings	VL	L	N	H	VH	XH
Product Factors	**1.09**	**1.05**	**1.0**	**0.96**	**0.92**	**0.87**
Simplicity			X			
Element Reuse	X					
Low-Priority Deferrals	X					
Models vs Documents		X				
Key Technology Maturity			X			
Process Factors	**1.09**	**1.05**	**1.0**	**0.96**	**0.92**	**0.87**
Concurrent Operational Concept, Requirements, Architecture, V&V				X		
Process Streamlining		X				
General SE tool support CIM (Coverage, Integration, Maturity)				X		
Project Factors	**1.08**	**1.04**	**1.0**	**0.96**	**0.93**	**0.9**
Project size (peak # of personnel)				X		
Collaboration support				X		
Single-domain MMPTs (Models, Methods, Processes, Tools)				X		
Multi-domain MMPTs		X				
People Factors	**1.13**	**1.06**	**1.0**	**0.94**	**0.89**	**0.84**
General SE KSAs (Knowledge, Skills, Agility)			X			
Single-Domain KSAs				X		
Multi-Domain KSAs		X				
Team Compatibility			X			
Risk Acceptance Factor	**1.13**	**1.06**	**1.0**	**0.94**	**0.89**	**0.84**
			X			

Table 5. Final To-Be Rating Factors

(Target Post-Adoption Rating Factors)

Accelerators/Ratings	VL	L	N	H	VH	XH
Product Factors	**1.09**	**1.05**	**1.0**	**0.96**	**0.92**	**0.87**
Simplicity			X			
Element Reuse	X					
Low-Priority Deferrals	X					
Models vs Documents		X				
Key Technology Maturity					X	
Process Factors	**1.09**	**1.05**	**1.0**	**0.96**	**0.92**	**0.87**
Concurrent Operational Concept, Requirements, Architecture, V&V					X	
Process Streamlining				X		
General SE tool support CIM (Coverage, Integration, Maturity)				X		
Project Factors	**1.08**	**1.04**	**1.0**	**0.96**	**0.93**	**0.9**
Project size (peak # of personnel)				X		
Collaboration support				X		
Single-domain MMPTs (Models, Methods, Processes, Tools)				X		
Multi-domain MMPTs		X				
People Factors	**1.13**	**1.06**	**1.0**	**0.94**	**0.89**	**0.84**
General SE KSAs (Knowledge, Skills, Agility)				X		
Single-Domain KSAs				X		
Multi-Domain KSAs		X				
Team Compatibility				X		
Risk Acceptance Factor	**1.13**	**1.06**	**1.0**	**0.94**	**0.89**	**0.84**
			X			

needs for rapid response projects, however, have led the division to desire a change to a more agile approach.

The baseline "as-is" factor ratings for the division are shown as boxes marked "X" in Table 3. The additional sub-factor detail available in this hypothetical division case study, only some of which was inferable in the commercial data, raises a question of how sub-factors that span a range of ratings should be handled. In COCOMO, a particular rating is chosen based on the preponderance of sub-factors that match the situation, possibly modified based on the expert judgment of the modeler. Here we have decided to average the multipliers, reasoning that higher ratings in some sub-factors offset lower factors in others.

The division's four product factor ratings are: moderately complex (N); sufficiently diverse to make reuse infeasible (VL); non-subsettable so that low-priority deferrals are infeasible (VL); able to use models vs. documents only 15% of the time (L); and involving only one or two slightly immature (TRL 6) technology elements (VH).

The three process factor ratings for the division are: highly sequential SE processes (VL); largely bureaucratic internal and external project and business processes (L); moderate SE tool coverage, integration, and maturity (H).

The division's four project factors are: project SE staff size between 10 and 30 people (H); good collaboration support across several metro-area facilities (H); moderate CIM for single-domain MMPTs (H); and minimal CIM for multi-domain MMPTs (L).

The *people* factor ratings are: good general knowledge, skills, and agility (KSAs) (H); good single-domain KSAs (H); good multiple-domain KSAs (H); but some difficult team interactions (L).

The divisions approach to *risk* is evenly balanced between risk-aversion and risk acceptance, leading to a nominal rating (N) and no effect on the schedule.

The selected ratings result in the following factor multiplier values, which calculates an overall acceleration factor of 1.01, suggesting the division's approach will result in a schedule duration close to the nominal case:

- Product: 1.0*1.09*1.09*1.05*0.92= **1.15**
- Process: 1.09*1.05*0,96= **1.10**
- Project: 0.96*0.96*0.96*1.04 = **0.92**
- People: 0.94*0.94*1.06*0.94 = **0.88**
- Risk: **1.0**

The division initially attempts a change to a more agile process approach by producing multiple artifacts (operational concept, requirements, and architecture) concurrently, instead of sequentially, as shown with the green arrow in Table 4. This was expected to reduce the schedule by 13%, from a 1.09 multiplier to 0.96.

However, when the project was performed, the organization was surprised that the actual schedule was about 15% longer rather than shorter. In performing a review of the cause of this, the division found that the project focused only on its agile and concurrency aspects, and neglected to examine the potential side effects of a too-hasty changeover.

With respect to the other CORADMO factors, the project missed several other factors that affect the overall schedule. These include missed opportunities in addressing some of the improvable SE schedule influence factors, but not others, such as the largely bureaucratic internal and external project and business processes, and the Low-rated multi-domain MMPTs and KSAs. Other detrimental effects resulted from pitfalls in transitioning from sequential, heavyweight processes to agile processes, as illustrated by the red arrows in Table 4:

- *Key Technology Maturity.* In producing artifacts concurrently, the project overlooked some interactions between subsystems, and mischaracterized the maturity of technologies through insufficient analysis. This resulted in a change to a Nominal from a Very High rating, causing a slowdown factor of 1.0/0.92=1.09.
- *General SE tool support.* Using a mix of agile MMPTs tools and traditional MMPTs made their MMPTs less integrated, increasing the sub-factor rating to High from Very High, for a slowdown factor of 1.0/0.96=1.04.
- *General SE KSAs.* Rapid development approaches required a different mindset from team members, causing a slowdown factor of 1.0/0.94=1.06.
- *Team Compatibility.* A different style of collaboration is often necessary in agile development, requiring frequent face-to-face discussions rather than serialized document reviews. Team members or management may be uncomfortable with or hostile to such interactions, resulting in an increase of this sub-factor to Nominal from High, for a slowdown of 1.0/0.94=1.06.

Therefore, although one of the process sub-factors improves as a result of the division's improvement initiative, due to unintended effects several other sub-factors become worse. The resulting CORADMO estimate of the net effect is 0.88*1.09*1.04*1.06*1.06=1.13. Thus, the CORADMO factor analysis not only explained their slowdown factor of about 15%, but also it provided them with a roadmap of further agile improvements they could make to begin to experience agile speedups, along with estimates of the impact that these would have on their schedule.

As illustrated here, when an organization is considering an improvement initiative, it can use CORADMO as a tool to identify potential side effects and analyze their impacts. With this information, the organization can then take steps to ensure these side effects are countered with additional aspects of the improvement initiative, and thus improve the likelihood of achieving a goal of schedule reduction.

Reconsidering its improvement initiative given this analysis, Table 5 shows the company preparing for the future by restoring the sub-factors whose ratings had worsened to their baseline values (the yellow arrows in Table 5), through being aware of the potential problems and therefore taking positive steps to avoid them. This would eliminate an overall slowdown factor of 1.09*1.04*1.06*1.06=1.29. They also added further initiatives, illustrated with green arrows in Table 5, to:

- Perform concurrent V&V along with concurrent Operational Concept, Requirements, and Architecture activities, raising the rating to Very High from High, for a speedup factor of 0.92/0.96=0.96
- Improve bureaucratic internal and external project and business processes to be at least moderately streamlined, for a speedup factor of 0.96/1.04 = 0.92.

If these goals were achieved, the resulting CORADMO multiplier estimate would be 1.13*0.96*0.92/1.29=0.77, for a speedup of 23% over their original situation. Further, this schedule reduction would be achieved through improving only three process sub-factors, and simply remaining at the pre-initiative levels for all other factors, through being aware of potential detrimental effects and taking steps to ensure they would not occur.

On their next project, they were not able to realize the full 23% speedup, but were able to realize a 15% speedup factor instead of a 15% slowdown factor. Continuing use of the CORADMO-based schedule acceleration framework enabled them to not only achieve their initial 23% speedup target, but to identify additional improvements that accelerated their schedules even further.

4. DISCUSSION

The combination of the original CORADMO model and the additional insights on product, process, project, people, and risk factors provided by the SERC RT-34 analyses enabled the revised CORADMO model to explain the variations in schedule acceleration among the projects in Table 2. This is encouraging, but it is unknown to what extent the model will accurately describe projects outside this limited set. We are in the process of collecting additional data points over a wider variety of projects. These data should allow us to better calibrate the model and evaluate its wider applicability. At a minimum, though, the model can be used as a good checklist for assessing an organization's status and prospects with respect to schedule acceleration.

Overall, as was observed in [9], an organization's culture is one of the critical factors in its ability to achieve near-term gains from going to agile methods. A good deal of careful re-culturing is needed to take an organization of people who feel comfortable and empowered by having sets of standard policies, practices, and procedures that define success in the organization, to an organization where people feel comfortable and empowered when there is a minimum of such standard policies, practices, and procedures. Several of the CORADMO factors can help in gauging an organization's progress in making such transitions.

5. ACKNOWLEDGMENT

This material is based upon work supported, in whole or in part, by the U.S. Department of Defense through the Systems Engineering Research Center (SERC) under Contract H98230-08-D-0171. SERC is a federally funded University Affiliated Research Center managed by Stevens Institute of Technology.

6. REFERENCES

[1] Anderson, D., Kanban: Successful Evolutionary Change for Your Technology Business. Blue Hole Press, Sequim, WA, 2010.

[2] Arthur, L., Rapid evolutionary development, Wiley, 1992.

[3] Baik J. and Boehm, B., "Empirical analysis of CASE tool effects on software development effort," *ACIS*

International Journal of Computer & Information Science, vol. 1, no. 1, pp. 1–10, 2000.

[4] Beck. K., Extreme Programming Explained, Addison Wesley, 1999.

[5] Boehm, B., A Spiral Model of Software Development and Enhancement, Computer, May 1988, pp. 61-72.

[6] Boehm, B., Clark, B., Horowitz, E. Brown, A.W., Reifer, D. Chulani, S., Madachy, R., and Steece, B., *Software Cost Estimation with Cocomo II with Cdrom*, 1st ed. Upper Saddle River, NJ, USA: Prentice Hall PTR, 2000.

[7] Boehm B., Ingold, D., Dangle, K., Turner, R., and Componation, P., "Early Identification of SE-Related Program Risks," Mar. 2010.

[8] Boehm, B., and Lane, J., Using the Incremental Commitment Model to Integrate System acquisition, Systems Engineering, and Software Engineering, Cross Talk, October 2007, pp. 4-9.

[9] Boehm, B. and Turner, R., *Balancing Agility and Discipline: A Guide for the Perplexed*. Boston: Addison-Wesley, 2004.

[10] Cusumano, M., and R. Selby, Microsoft Secrets, Harper Collins, 1996.

[11] Deming, W.E., Out of the Crisis, MIT Press, 1982.

[12] Ford J., Colburn, R. ,and Morris, R. "Principles of Rapid Acquisition and Systems Engineering," Air Force Institute of Technology, Wright-Patterson Air Force Base, Ohio, 2012.

[13] Highsmith, J., Adaptive software development: A collaborative approach to managing complex systems, New York: Dorset House Publishing, 2000.

[14] Kruchten, P., The Rational Unified Process, Addison Wesley, 1999.

[15] Lepore D., Columbi, J., et al., "Expedited Systems Engineering for Rapid Capability and Urgent Needs," Final Report, SERC RT-34, December 2012. http://sercuarc.org

[16] McConnell, S., Rapid development. Redmond, WA: Microsoft Press, 1996.

[17] Murman, E., Allen, T., Bozdogan, K., Cutcher-Gershenfeld, J., McManus, H., Nightingale, D., Rebentisch, E., Shields, T., Stahl, F., Walton, M., Warmkessel, J., Weiss, S., and Widnall, S., Lean Enterprise Value: Insights from MIT's Lean Aerospace Initiative, New York, NY: Palgrave, 2002.

[18] Ohno, T., Toyota Production System, Productivity Press, 1988.

[19] U.S. Air Force, Quick reaction capability process. (AF Instruction No. 63-114). U.S. Air Force, 2011.

[20] Womack, J. P., & Jones, D. T., Lean thinking : Banish waste and create wealth in your corporation Simon & Schuster, New York, NY, 1996.