



MOC BEST PRACTICES

Gateway Consulting Group, Inc.

a leader in Management of Change process redesign and Electronic Document Management systems implementation

January, 2011

Volume 5, Number 1

Contents

- My Perspective
- In the Next Issue
"Action Item Management"

Upcoming Seminars:

Chem NEP Readiness
April 13, 2011
Element - Westin Hotel
14555 Vintage Preserve Pkwy
Houston, TX 77070

MOC Best Practices
April 14-15, 2011
Element - Westin Hotel
14555 Vintage Preserve Pkwy
Houston, TX 77070

Contact:

Gateway Consulting Group, Inc.
8610 Transit Road, Suite 100,
East Amherst, NY 14051
Phone: (800) 668-2334
eMail: info@gatewaygroup.com
www.gatewaygroup.com

Copyright © 2011
Gateway Consulting Group, Inc.
ALL RIGHTS RESERVED

My Perspective...

"Where's the latest MOC Best Practices newsletter?" To be honest, I've been busy assisting clients getting their Chem NEP Readiness initiatives kicked off, and also working on several MOC deployments (where we take the applicable best practice concepts, and actually implement them in working systems). But that's not why there's been a gap in newsletter publishing—read on...

Last July, I started on the task of creating a complete computational model of the MOC business process. That took a month. Validating the model took six months. Our business at Gateway is split between energy/chemical and life sciences clients, and the latter have very thorough software validation requirements imposed on them by the FDA. It's through the FDA-related work that my colleagues and I have acquired the skills and experience needed to validate both software and business processes implemented in software. At this point I have a validated MOC business process model, where we can truly trust the results. I call this the "MOC Standard Model".

So what good is it?

The MOC Standard Model is a validated benchmark against which business process modifications can be compared. You might ask questions like:



1. "How long does/should it take to do small, medium or large MOCs?"
2. "If we add another approver, how much extra does it cost? How much longer will it take?"
3. "If we scope the MOCs more thoroughly, what are the savings in scrap reduction?"
4. And, of course, everyone's favorite question, "If we do electronic MOCs, how much will we save?" Answer: "far more than you think"

Answers to these questions will evolve over future issues. Now, at least, we have the tools that allow us to come up with a reasonable answer.

1. Introduction

What constitutes good MOC practice, and how would one optimize a given MOC process?

This paper begins with a review of an MOC using a lifecycle concept [1]. The lifecycle model is expanded into a detailed task-level model of an MOC. The task-level model is used as the basis for a computer simulation model using the Petri net method [2]. The Petri net model is used to determine realistic values of important MOC performance characteristics.

2. Models of the MOC Process

2.1 Lifecycle Model

A high-level model of MOC, referred to as a "lifecycle" model, is shown in Figure 1. The diagram depicts a permanent, non-emergency MOC process.

A lifecycle is composed of "states": Initiation, Scoping, Change Design, etc. While the MOC is active, it could be considered to be "in" a state: e.g. the MOC is in the Approvals state.

2.1 Lifecycle Model continued...



Figure 1. Lifecycle model of an MOC process.

The notion of a business process represented as a collection of states is not new:

- NASA, instituted a “phased review process” in the 1960’s,
- various project management methodologies use the concept of a “stage-gate” methodology [3],
- although much more formalized, mathematicians and electrical engineers have used finite automata and finite state machine concepts for decades.

The more common tasks performed during each lifecycle state are listed in Table 1.

State	Major Activities
Initiation	<ul style="list-style-type: none"> ■ The change is formally started ■ Risk pre-screening is conducted
Scoping	<ul style="list-style-type: none"> ■ The change is scoped, which identifies, at a minimum: <ul style="list-style-type: none"> • What tasks need to be done • Who reviews the MOC • Who approves the MOC • Who is notified
Change Design	<ul style="list-style-type: none"> ■ Documents that describe the change are created or redlined
Impact Analysis	<ul style="list-style-type: none"> ■ The potential impacts of the change are analyzed. These include: <ul style="list-style-type: none"> • Process safety impacts (typically PHA’s) • Environmental impacts (e.g. permitting impacts) • Financial impacts (i.e. cost estimates) • Safety impacts (e.g. personal protective equipment, etc.) • Industrial hygiene impacts • Product and quality impacts
Approvals	<ul style="list-style-type: none"> ■ Previously-identified reviewers and approvers accept, reject or cancel the proposed change. ■ Rejection implies that further work is needed to detail the physical or financial aspects of the change proposal.
Implementation	<ul style="list-style-type: none"> ■ The physical change is made in the plant. ■ Inspections and mechanical integrity activities are conducted. ■ Procedures are updated. ■ Training is conducted.
PSSR	<ul style="list-style-type: none"> ■ In the case of plant-based MOCs, the Pre-Startup Safety Review validates that the change has been made safely. Discrepancies are noted and resolved. PSSR ends with an authorization to start up. ■ For non-plant-based MOCs (e.g. lab), or for changes that don’t end in a “start-up”, this is usually called the “Validation” state, rather than PSSR. The logic is the same inasmuch as a final check is made before the change is put into production.
Close-out	<ul style="list-style-type: none"> ■ The change is formally closed. <ul style="list-style-type: none"> • Metrics are gathered. • Redlines are incorporated into the base documents and released. • Quality assurance steps may be conducted.

Table 1: Typical tasks performed during each state of an MOC.

2.2 Task-Level Model

While an MOC is in a given state, a number of tasks must be completed in order to accomplish the work of the task. Once all the tasks for a given state are completed, the gating criteria are satisfied, and the MOC is “promoted” to the next state.

The task-level model shows the individual tasks that take place during a state. For instance, Figure 2 shows the first few tasks that take place during the Change Design state. The logic of this model fragment is as follows:

Assumptions:

- The MOC is small
- Small MOCs, by definition, require that 3 documents are redlined
- Small MOCs, also by definition, require that only 1 person do the redlines. That person is the MOC “owner”

The tasks are:

- CD-01 The Initiator decides whether s/he has the expertise to redline the drawings
- CD-02 If the Initiator does not have the expertise to redline the drawings, then someone, perhaps his/her manager, reassigns ownership of the MOC
- CD-03 The new owner of the MOC must spend some time understanding the requirements of the MOC; i.e. the new owner rises the learning curve for this MOC
- CD-04 Once the new owner understands what is needed, s/he redlines the 3 documents needed to define the proposed change. If ownership didn't change, as a result of a “No” decision in CD 01, then the Initiator redlines the 3 documents.

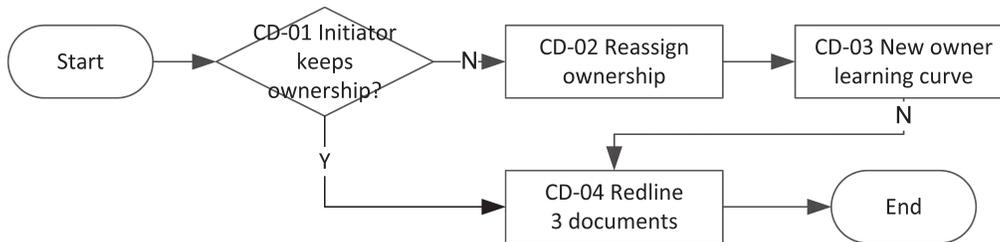


Figure 2. Task-level model fragment, for part of Change Design state

Only 3 tasks and 1 branch are shown in Figure 2. The entire MOC task-level model has over 300 tasks and over 180 branches.

Note that the lifecycle model, Figure 1, only has ordering at the state level: i.e. all the tasks for one state have to be completed prior to the next state beginning. Within a state, the lifecycle model does not recognize task ordering. The task-level model, Figure 2, is highly ordered, although parallel tasks are certainly permitted. This poses a challenge when creating a model for the typical MOC since the exact order that tasks are performed during the actual execution of an MOC varies from one MOC to another. For the purposes of MOC performance assessment, any reasonable task-level model is likely to be adequate. Certainly the tasks and branches in Figure 2 look reasonable.

2.3 Computational Model

The task-level model describes the MOC process in sufficient detail that interested individuals can discuss the details of what happens in a given state. In order to quantitatively analyze the performance characteristics of MOC, it's necessary to develop a computational model. There are three common approaches to business process quantification:

1. Spreadsheets: while possible to create an MOC computational model using a spreadsheet, there are formidable difficulties dealing with complexity and the demands of stochastic processes,
2. Packaged business process simulation software: a number of vendors offer software to model business processes, and this is certainly a viable approach,

2.3 Computational Model, continued...

3.3. Petri nets: Petri nets were developed by C.A. Petri, in 1962 [2], and have been popularized for business process use by vanderAalst [4] and Reijers [5]. The strength of Petri nets is that they can represent business processes at a very granular level, and the size and complexity of the model is only limited by the analyst's time in assembling it. Timed Petri nets can easily handle stochastic processes.

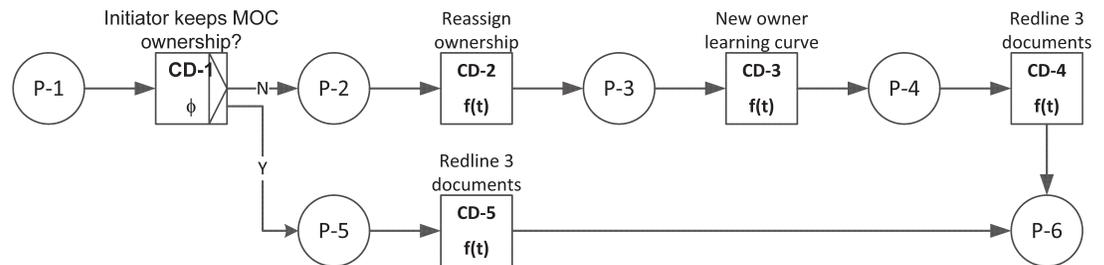


Figure 3. Computational model fragment for part of Change Design state.

While a comprehensive discussion of Petri nets is beyond the scope of this paper, the use of Petri nets is very analogous to the use of finite element methods for structural analysis problems. Specifically, a finite element model consists of nodes, which describe the model's topology, and elements which have prescribed behavior. In a Petri net, as shown in Figure 3, the topology of the business process is described by "places", indicated by the circular symbols, and the behavior of the model is governed by "transitions", indicated by the square symbols.

The places in Figure 3 are:

- P-1 is the start place.
- P-2 thru P-5 are places
- P-6 is the end place.

Place and transition labeling is arbitrary; these places could just as easily have been called Tom, Richard, Harriett, etc. The only requirement is that the names are unique.

The transitions in Figure 3 are:

- CD-1 The ϕ signifies that this is a "no operation" transition. Nothing happens, except that this output has an OR-split. Process flow goes from CD-1 to P-2 some percentage of the time, say, $x\%$. Process flow goes from CD-1 to P-5 $(1-x)\%$ of the time. Whether the process flow goes to P-2 or P-5, for a specific MOC, is unknowable in advance. However, over many iterations, the percentage of times that P-2 is activated will tend to $x\%$.
- CD-2 The $f(t)$ signifies that a resource performs a task during this transition, and that performing the task takes time. In this case, since the Initiator has not kept ownership of the MOC, the Initiator's manager reassigns ownership to someone else.
- CD-3 Like CD-2, this is a transition where work takes place. The new owner must rise the learning curve on the MOC just assigned to him/her.
- CD-4 The new owner redlines 3 documents, to document the proposed changes for a small MOC.
- CD-5 The Initiator has kept ownership of the MOC and redlines the 3 documents.

The transitions CD-2, 3, 4 and 5 model two different aspects of time:

- Start Time: the time before the person starts to work on the task. Just because someone has been assigned a task, doesn't mean that they are available to perform the task—they may be busy with other work. So each transition is given a start time of, say, 8 hr, whereupon the resource begins the work.
- Duration: the time it actually takes the resource to perform the task, say, 1 hr. Multiplying the duration of a task by the hourly rate of the resource, say \$50/hr, gives the labor cost for the task.

2.3 Computational Model continued...

The MOC performance model is intended to be as realistic as possible. Assigning a start time of exactly 8 hr and a duration of exactly 1 hr is not realistic, since these times are distributed according to some probability density function. Possible probability density functions include:

- Fixed time: $t = 8$, means that each event occurs at exactly the 8 hr point.
- Uniform distribution: events are evenly distributed in the interval $0 \leq t \leq 16$.
- Gaussian distribution: the “bell curve” distribution cannot be used for this work because the left tail of the distribution extends to $-\infty$, which would result in negative time.
- Weibull distribution: events conform to the probability distribution function shown below. In this example, the shape parameter, $k = 2.0425$, and the scale parameter, $\lambda = 9.5724$, produces a distribution with the median time of 8 hr.

$$p(t) = \left(\frac{k}{\lambda}\right) \left(\frac{t}{\lambda}\right)^{(k-1)} e^{-\left(\frac{t}{\lambda}\right)^k} \quad \text{for } t, k, \lambda > 0$$

This is what is meant by the term “stochastic process”, i.e. that many of the parameters are not given fixed values, but are specified as a statistical distribution, with the appropriate parameters.

2.4 The Complete Model

A Petri net model was created of the entire MOC process. It consists of 1,410 places and 1,273 transitions. In addition, 152 parameters were given realistic values: for example, average wait time before someone begins his/her task is 8 hr; percent of time that the Initiator retains ownership of the MOC and creates the redlines is 50%; the hourly rate for a professional person is \$75/hr, and so on for the remaining parameters.

The Petri net model determines two kinds of costs:

1. Labor costs: this is the effort involved in each task times the hourly rate for the resource,
2. Scrap costs: material costs are not included in the analysis, because they are determined by the physics of the change, and not impacted by the MOC business process. Scrap costs, however, are related to the business process: a poor MOC process, with inadequate documentation, will lead to higher scrap costs. Due to the variability of the true cost of scrap, an estimate of \$1,000 per scrap event is used.

2.5 Small, Medium and Large MOCs

A review of MOC data from Gateway clients reveals that there is no one “typical” MOC, at least not at the level of detail of this study. There is a large variation in size and complexity of MOCs. One way to deal with this is to define certain MOC configurations that are representative of the universe of MOCs. The author’s team acquired MOC data for thousands of completed changes. Only permanent and non-cancelled MOCs were analyzed. The MOCs were assigned 2 values: (1) the number of documents redlined during the Change Design state, and (2) the number of approvers that affixed their signatures during the Approvals state. Each MOC, thus scored, was then sorted according to the scores. The 10th-percentile, 50th percentile and 90th-percentile points were then identified, and labeled “small”, “medium” and “large”.

MOC Label	Small	Medium	Large
Size	10 th percentile	50 th percentile	90 th percentile
Number of documents redlined	3	5	12
Number of approvers	3	5	8
Process safety impact analysis	Checklist	What-if	HAZOP
Number of crafts involved in implementation	2	5	8
Number of inspections	2	5	8
Shutdowns during Implementation	No shutdown	Isolated equipment shutdown, in 20% of cases	Unit shutdown, in 20% of cases

Table 2. Parameters describing small, medium and large MOCs.

3. Analysis

A Petri net model was constructed that models the behavior of small, medium and large MOCs. The collection of Petri net model plus the associated set of parameters has been labeled the "MOC Standard Model". This is the first attempt at defining such a model, and should therefore be considered preliminary.

3.1 Validation

A numerical model, such as a finite element or computational fluid dynamics model is only useful to the extent that the results can be trusted. Analysts using either of these methods typically validate their models by comparing results for steady-state cases for simple geometries, with known theoretical solutions.

For the MOC Petri net model, a similarly careful model validation exercise was undertaken. The validation approach was as follows:

- The model was constructed one state at a time. So the first model part was created for Initiation. The second part was created for Scoping, and so on.
- A parallel exercise was undertaken to create a spreadsheet model of each MOC state. The spreadsheet model could only represent the MOC process using the Fixed Time distribution for the time-dependent tasks, and with very restricted branching. More complex task time distributions or more complex branching made the spreadsheet calculations intractable.
- Results were calculated using the Petri net model, which was set to use the Fixed Time distribution. These results were compared to the spreadsheet results. Discrepancies were addressed until these 2 independent approaches yielded identical results.
- This process was repeated for each lifecycle state.
- As the complete model was being assembled, these validation tests were rerun. First the process was rerun with Initiation + Scoping, then Initiation thru Change Design, and so on until the entire process Initiation thru Close-out was validated.
- Once the entire model was thoroughly validated, all analyses were rerun with time-dependent parameters conforming to the Uniform distribution, and then again with the Weibull distribution, by setting the appropriate flag.

4. Results

The results in sections 4.1 through 4.5 are taken from the MOC Standard Model. Section 4.5 describes the results of using the Standard Model to assess other conditions, specifically the impact of errors.

4.1 Basic Results

The basic results for the entire MOC are shown in Table 3a (unless otherwise indicated, the results henceforth assume that time-dependent parameters follow a Weibull distribution). While the specific values in Table 3(a) are applicable to a specific set of parameters used in the Standard Model, the ratios between large, medium and small, from Table 3(b), are more generally instructive.

The duration calculation is based on the notion of an 8 hr workday, with no activity presumed for off-shift hours. Also, the Standard Model presumes the existence of an electronic MOC system, and electronic access to supporting documentation. It is highly unlikely that the durations shown in Table 3(a) could be achieved with a paper-based MOC system.

The effort and cost are closely related. Greater effort implies a greater cost. While cost and effort are linearly related for individual tasks, the entire MOC requires a mixture of resources at different hourly rates. That explains why the Effort Ratio and the Cost Ratio for medium and large MOCs are similar, but not identical.

MOC Size	Effort [person-hr]	Duration [d]	Cost [\$]
Small	37.51	22.95	2,603
Medium	112.58	34.18	7,201
Large	438.14	76.60	28,582

(a)

MOC Size	Effort Ratio	Duration Ratio	Cost Ratio
Small	1	1	1
Medium	3	1.49	2.77
Large	11.68	3.34	10.98

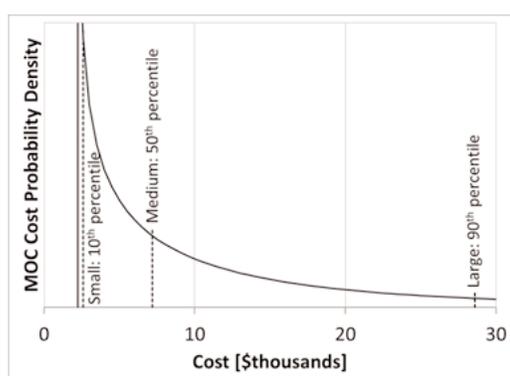
(b)

Table 3. Basic results for entire MOC process: permanent, non-emergency.

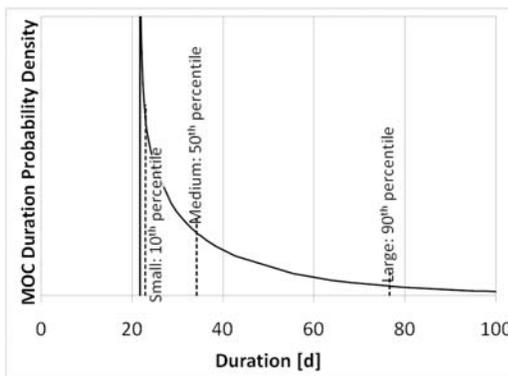
(a) Time and cost values. (b) Values normalized to small MOCs.

4.2 Cost and Duration Distributions

The 3 cost values, from Table 3(a), are fit to a curve to illustrate the probability distribution of MOCs with respect to cost, shown in Figure 4(a). It's clear that most of the MOCs are at the low cost end of the scale. A similar analysis is provided for MOC durations, in Figure 4(b).



(a)



(b)

Figure 4. (a) Distribution of MOC costs. (b) Distribution of MOC durations.

These diagrams point to an ongoing problem that faces MOC practitioners. When an MOC process is designed (i.e. the MOC procedure is written) at a site, the process must work for all MOCs. Therefore, the MOC process must accommodate the needs of the largest MOCs. But, at least 50% of the time, the MOC process is used for MOCs that are 4 to 11 times smaller than what the process was designed for. This partially explains why MOC users often find the process burdensome.

One solution to this problem is to make the MOC adaptive. One common technique, for paper MOCs, is for the MOC procedure to be comprehensive (i.e. covers even the largest MOCs), but many of the supporting documents or checklists are optional (and not needed for small MOCs). Implementing adaptive MOCs using a paper MOC process is challenging:

- if all the optional documents are provided as part of the MOC procedure, then the MOC package quickly becomes unwieldy (one site had 48 pages, including the optional ones!),
- if the optional documents are not provided as part of the MOC procedure, then the user must search for them in order to complete the MOC.

Electronic MOC systems are better suited to implementing adaptive MOCs.

4.3 State Breakdown

Figure 5 presents the cost and duration breakdown by state.

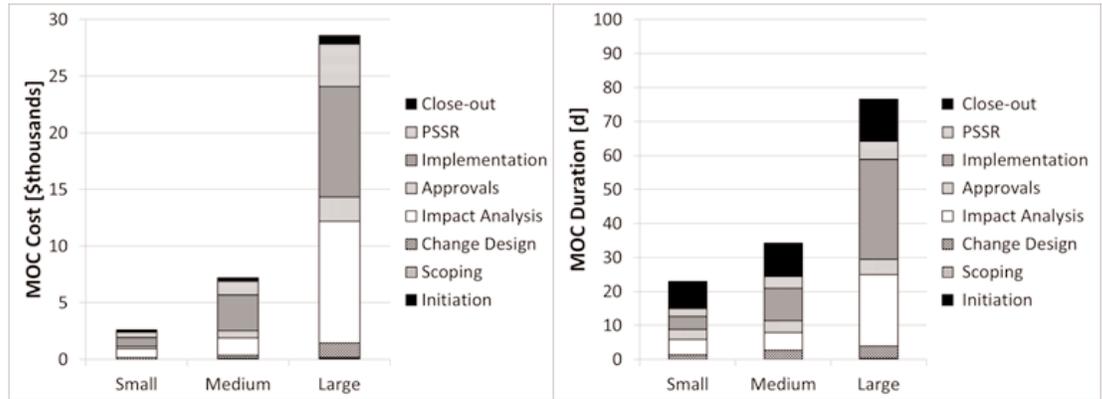


Figure 5. (a) Cost breakdown by state. (b) Duration breakdown by state.

The relative cost and duration of Initiation and Scoping are so small that they are not visible at the scale of the bar graphs.

The relative ease of MOC creation has policy implications. In the interests of safety and encouraging employee participation, most sites permit almost anyone to initiate an MOC. But MOC initiation has huge leverage: for every dollar spent on initiation (i.e. the initiator's salary for the time involved), the company will spend another ~\$300 on pursuing the change (and this doesn't even include material cost). Therefore, it appears reasonable that the manager who is responsible for the asset should be required to accept the MOC after it has been scoped. In the absence of this prioritization function, the MOC system at the site will become clogged with dozens or hundreds of in-process MOCs.

Scoping also has huge leverage, like Initiation. Typically, scoping represents about 1/500th the cost of the overall MOC. Yet, the "work" of the MOC is determined during the scoping activity. This suggests that adequate attention must be paid to effectively scope the change.

4.4 Implementation: (Typically) The Largest Cost

The Petri net model attempts to realistically model the implementation process. Key features include:

- unit or equipment shutdown may be required (see Table 2)
- development of detailed change specifications, in addition to the redlines created during Change Design
- purchase requisition generation, ordering, material receipt, work order generation and issuing
- activities of representative crafts
- procedure updates
- training material development and training activities.

With this scope of work, it is not surprising that the Implementation state has the greatest cost.

4.5 Impact Analysis:(Typically) The Second Largest Cost

The MOC Standard Model considers three kinds of impacts:

- process safety impacts, usually addressed using a process hazards analysis
- environmental impacts
- financial impacts

It is assumed that the risks associated with small changes require less detailed analysis. So, the process safety impacts are analyzed using a checklist for small changes. Similarly rapid analysis techniques are used for environmental and financial analyses. Large MOCs are assumed to have risks that require more detailed analysis. So, the process safety risks are analyzed using the HAZOP methodology. Similarly detailed techniques are used for the environmental and financial analyses.

The detailed analysis techniques have two common features:

1. They require multiple people to assemble for a meeting to discuss the specific impacts of the MOC, thereby increasing the resource costs significantly.
2. The meetings must be scheduled in advance, thereby significantly increasing the duration of the impact analyses.

Clearly, when a detailed impact analysis is needed for risk mitigation purposes, its expense and time are unavoidable. However, if there is no reason for a detailed impact analysis, then there are significant cost and time benefits to performing only the necessary analysis.

How and when would one determine what kind of impact analysis is necessary? This can be done using pre-established rules, applicable to the site and its inherent hazards. The best time to identify the impact analysis methodologies is during scoping, since this allows the necessary resources to be arranged ahead of time.

4.6 The Effects of Errors

The previous results all used a zero-error assumption. Of course, that is not realistic—errors occur all the time, and error mitigation creates extra cost. The kinds of errors represented in the Standard Model are:

1. **Errors discovered during Impact Analysis:** If the change proposal, as documented in the redlines, does not provide sufficient information, the team/person conducting the impact analysis may stop and await additional information.
2. **Approvers rejecting the change:** Approvers may reject a change if it is improperly or inadequately described. The process stops awaiting additional information.
3. **Errors discovered during Implementation:** Prior to implementation, the change could be considered somewhat abstract—during implementation, the change is most definitely concrete. During implementation four kinds of problems are commonly encountered:
 - a) **Design deficiencies:** These are errors in the design of the proposed change. "Design" is to be interpreted as any error in documentation, which makes it difficult or impossible to continue work on the change.
 - b) **Physical discrepancies:** This is a common problem whereby it's discovered that the documentation does not correspond to what is actually installed in the plant already. When the documentation does not accurately reflect the current plant configuration, it brings into the question the adequacy of the change proposal as well as the adequacy of the various impact analyses.
 - c) **Execution deficiencies:** These are mistakes made by the resources implementing the change.
 - d) **Material deficiencies:** These are inadequate parts or materials, whose inadequacy may be due to the incorrect materials being installed or damaged during installation.

In all cases, the errors are further classified as follows:

- **Minor errors:** Minor errors are those that can be remedied quickly, and will not change any of the impact analyses in any way. The MOC remains in the same state, and the process continues after the error is remediated.
- **Major errors:** Major errors are those where documentation is significantly altered, and the changed information causes one or more of the impact analyses to be modified. When a major error is encountered, the MOC is usually demoted to the Change Design state. Once the documents are modified, the change must work its way through its lifecycle again, beginning at Change Design.

Reasonable values were selected for the probability of major or minor errors of types 1 thru 3(d), from the previous list. The minor errors have little effect on the already expensive MOC process. The charts in Figure 6 compare results with and without errors. The minority of MOCs that do have major errors (~10%), have their costs and durations almost double. But, when the doubling of costs and durations is averaged over all MOCs, the average results are still significant, but not that startling.

4.6 The Effects of Errors continued...

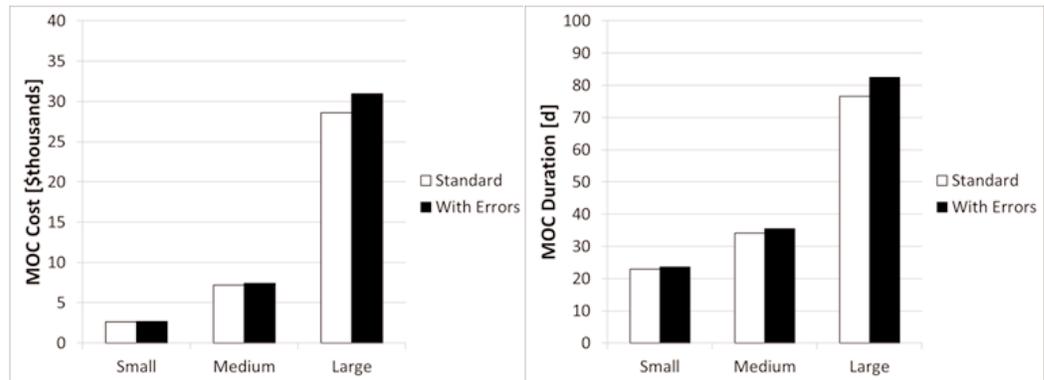


Figure 6. Impact of errors on (a) costs, and, (b) duration.

Anything that can be done to reduce the frequency of errors in MOCs is clearly a benefit: perhaps better scoping, perhaps greater care in redlining the proposed change?

5. Conclusions

The lifecycle or "stage-gate" model is useful for describing the high level MOC business process, and the lifecycle is further detailed into a computational model. The computational model uses the Petri net method which permits very granular representation of business processes including branching and stochastic behavior of resources. The model was extensively validated.

A "Standard MOC Model" is proposed which can be used as a benchmark in business process improvement studies. The standard model reveals several things about MOCs:

- large MOCs are ~11 times as costly as small ones
- large MOCs have a duration of about 3 times as long as small ones
- Implementation and Impact Analysis are the most costly states, and require the greatest time
- When major errors occur in the execution of an MOC, the cost, as well as the duration, of the MOC can easily double

As discussed in the text, the following ideas may make MOCs more efficient:

- Managerial review of MOCs just after initiation and scoping
- Better scoping tools, to ensure that the work of the MOC is properly scoped
- Adequate time and care to ensure that the proposed change is properly documented on the redlines.

6. References

- [1] Hoff, R., 2007, "Lifecycles," MOC Best Practices, 1(2).
- [2] Petri, C. A., 1962, "Kommunikation mit Automaten," Ph.D. Dissertation, University of Bonn, Bonn.
- [3] Belliveau, P., Griffin, A., and Somermeyer, S., 2002, The PDMS toolbox for new product development, John Wiley & Sons, inc., New York.
- [4] vanderAalst, W., and vanHee, K. M., 2002, Workflow management: models, methods, and systems, MIT Press, Cambridge, MA.
- [5] Reijers, H. A., 2003, Design and control of workflow processes: business process management for the service industry, Springer, Berlin, New York.