



MOC Best Practices Whitepaper:

MOC Lifecycles

Rainer Hoff, Ph.D., P.Eng.
President
Gateway Consulting Group Inc.

December 2007

Copyright, © 2007, Gateway Consulting Group, Inc.—all rights reserved.

Selecting the Correct Framework for Discussion

There are many forms of communication that can be represented and analyzed using a “stack” model. The left hand diagram of Figure 1 shows a simplified stack for speech. At the physical level, speech requires sound waves—vocal cords and hearing apparatus are necessary for communication at this Physical Level to occur. Vocabulary and the rules of grammar govern communication in the Language Layer. Finally, the Semantic Layer represents that meaning that is implied by the words. For example, the words, “I saw a tokamak yesterday” are acceptable words, used in proper sentence structure (language). But the meaning is “yesterday I saw a tokomak nuclear reactor, so I must have been at a nuclear power or research facility” (semantic).

The second example is the well-known Internet Protocol Stack¹. And finally, there is the MOC Business Process Stack.

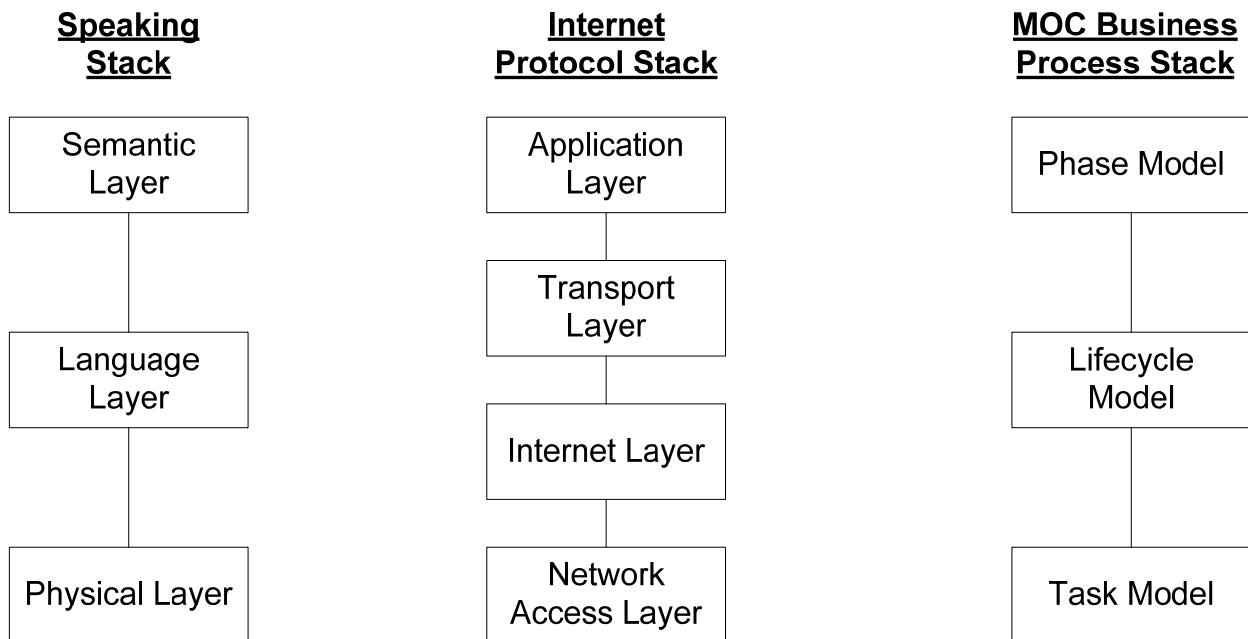


Figure 1. Protocol stacks.

Note how each layer “uses” the facilities provided to it by the lower layer. Note also, that the higher layers are more useful for conceptual discussions than the lower layers.

It’s very difficult to discuss MOC business processes, when the discussion begins at the task level. Unfortunately, many MOC improvement initiatives begin with a current flowchart of tasks, and then an attempt is made to generalize the flowchart into a set of guiding principles. Regrettably, the important goals and objectives of MOC are lost in the minutiae of individual tasks (“...and

¹ Some readers may be familiar with the Open Systems Interconnection (“OSI”) Basic Reference Model which describes a 7-layer stack. The diagram in Figure 1 shows the Internet Protocol stack which is not intended to be OSI compliant.

then the environmental rep. fills out the ENV-123 form...) and all the necessary exceptions (“...except if it’s a temporary MOC, and then we...”).

The author contends that MOC best practices are best discussed using a top-down approach, beginning with a high level “Phase” model, decomposing that into a “Lifecycle” model, and finally decomposing that into “Task” models (when necessary).

The MOC Phase Model

At the highest level, MOC’s can be represented by a Phase model, as shown in Figure 2.



Figure 2. MOC phase model.

“Problems” can be identified in a number of ways. Common examples are:

- Audit findings may suggest a change to equipment or process
- Incident investigations may reveal problems
- Other MOC’s may identify problems
- Inspections may discover problems
- An issue tracking or management system may house a list of “things to do”
- Engineering studies may identify how a chemical process would be improved by making certain changes.

Although Problem Identification is the necessary first phase, it is very diverse, and generally considered a precursor to the MOC process, not an integral part of the MOC process. Noting the similarity among the entries in the bulleted list, above, if one were to inclined to automate the problem identification phase, the logical tool would be an “issue management” (computer) application.

In the United States, the OSHA Process Safety Management² regulations are rather explicit, describing various elements which occur during Change Request: document the technical basis for change, update process safety information, assess the impact of the change, and acquire proper approvals. Also, the OSHA PSM regulations describe various elements which occur during Change Implementation: mechanical integrity inspections, pre-startup safety review, procedure updates, and training. Consequently, the Change Request phase and the Change Implementation phase are generally combined into a single business process, in the U.S., and that process is called

² 29CFR1910.119. The Process Safety Management regulations promulgated by the U.S. Occupational Health and Safety Administration.

the Management of Change, or “MOC”, process. Other regulations³ and recommended practices⁴ also identify elements of both the change request and the change implementation phases.

The combining of request and implementation into a single business process, is not universal. For instance, the author was recently involved in two MOC projects outside of the U.S. The first project had a comprehensive change request phase, similar to the US approach. However, change implementation was outside the scope of the management of change business process at this site, and was handled as a routine work order. The second project had a comprehensive change implementation phase, similar to the US approach. However, the change request was outside the scope of the management of change business process, and was handled like an extension to the issue management system.

Based on these observations, it appears that the combination of both the Change Request phase and the Change Implementation phase is the superset of all MOC business processes. So, henceforth, we shall consider Management of Change to be a business process which consists of both the Change Request phase and the Change Implementation phase.

MOC Lifecycle Models

The Change Request phase and the Change Implementation phase can each be decomposed into a number of states. The collection and sequence of states is termed the lifecycle. The essential difference between different kinds of MOC’s (e.g. full, short, permanent, temporary, normal, emergency, etc.) is that they have different lifecycles. A lifecycle for a full, permanent MOC is shown in Figure 3.

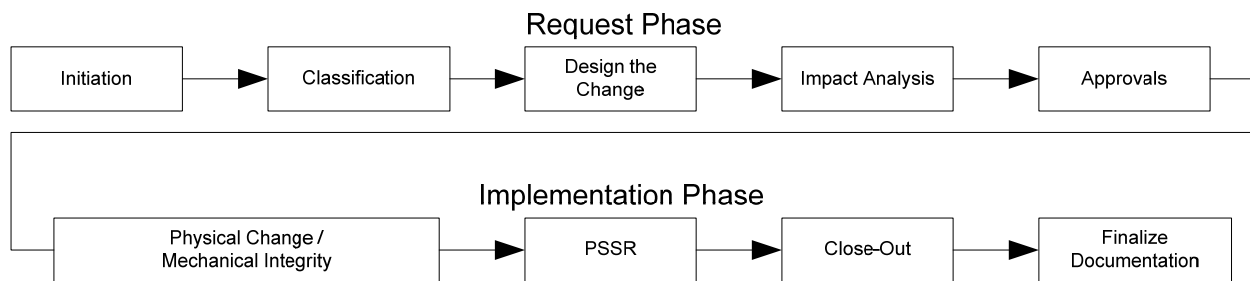


Figure 3. MOC lifecycle model—full, permanent MOC.

The diagram indicates that the lifecycle begins with the *Initiation* state. The MOC is “promoted” from one state to the next. An MOC can often be “demoted” from a state to the previous state. Also, there is a possibility of looping back to a much earlier state, as would be the case if the physical change is in the process of being made and an inspection discovers an unexpected

³ 40CFR68.75. The Risk Management Plan regulations promulgated by the U.S. Environmental Protection Agency.

⁴ API RP-75, *Recommended Practice for Development of a Safety and Environmental Management Program for Offshore Operations and Facilities*, American Petroleum Institute, Washington, May 2004.

configuration which warrants additional design work—this would cause the MOC to loop back to the *Design the Change* state.

States

A state is “a set of properties that describe the condition of something”. In the context of MOC, a state:

- has a set of rules associated with it.
- governs access control.
- generally does not require its subordinate tasks to be done in a certain order.

Rules. The most important rules, associated with a state, are those governing promotion. In order for the MOC to be promoted from one state to the next, certain things have to be completed. These “things” can be broadly categorized as:

- Work item completion: typically, certain documents must be created, or updated
- Approvals: sometimes, certain persons must approve the work items or the MOC itself
- Notifications: often, various stakeholders must be notified

Access control is usually different for each state. Access controls govern which person can perform which actions on which documents, forms and other work items. The person can be an individual (e.g. J. Brown), have a certain role (e.g. MOC_Coordinator), or be a member of a group (e.g. Area_Managers). There are many possible actions in an electronic MOC system, but the most common are query, search, view, modify and delete. The work items may include documents, forms, database entries, and, of course, the MOC form itself.

It’s clear that different access rights are needed at different states. For example, during *Initiation*, it’s reasonable that an Initiator can delete an MOC if s/he decides an MOC isn’t warranted by the circumstances. But, at any other state, especially *Close Out*, there’s no ability to delete an MOC.

Subordinate tasks, that need to be completed in order for the MOC to be promoted, do not generally need to be done in any particular order—they just need to be completed prior to state promotion. This is an area where trying to optimize MOC’s using a task-based model (rather than a lifecycle model, as discussed herein) usually fails; task-based models tend to order things unnecessarily. Consider these examples:

- Boy Scout Rank Advancement: To be promoted from “2nd Class Scout” to “1st Class Scout”, in the US, a person needs to participate in a certain number of campouts, a certain number of troop activities, and demonstrate proficiency in various elements of outdoor craft (e.g. reading a compass). These can be done in any order.
- Gold Level Frequent Flyer Plan Membership: Suppose you are a gold level member in a frequent flyer plan. You may be able to use the airport lounge for free, apply for free domestic upgrades, and so on (i.e. enhanced access rights). In order to get promoted to the platinum level, you may have to fly an additional, say, 25,000 miles. Of course, there’s absolutely no prescribed order to the flights you take and when you take them.

- *Design Change* State of MOC's: During the *Design Change* state, it may be necessary to update the P&ID, update the PFD, obtain an equipment specification and enter it into the appropriate filing system. Again, the success of the MOC process isn't dependent on the order that these work items are accomplished in.

Trying to force an order on these tasks simply slows things down. Besides, the characteristics of the change and availability of resources are the factors that actually drive what order the work will be completed in.

Stage-Gate Processes

This discussion on lifecycles and states, may be familiar to some readers, since it's the model that's used in what's commonly called the "Stage-Gate" methodology. Most organizations have a methodology of conducting capital projects, that's based on a Stage-Gate approach. The MOC lifecycles, discussed herein, are essentially Stage-Gate processes.

The Full, Permanent MOC Lifecycle

The full, permanent MOC lifecycle is shown in Figure 3. A brief overview of what the expectations are for each state, appears in the whitepaper, *MOC Best Practices*, also found at www.gatewaygroup.com.

Other Lifecycles

There are several other common MOC lifecycles which are listed in Table 1. How they compare to each other is shown in the Venn diagram of Figure 4. Lifecycle diagrams for each of these appear in the figures that follow.

Each one of these alternate lifecycles warrants its own lengthy discussion, and will be a topic for future newsletters. Furthermore, the rules associated with each state will be elaborated on in the future. Some topics will be debated: e.g. what does "finalize" mean in *Finalize Documentation* and why does this follow *Close-Out* ?

| MOC Lifecycle Name | Brief Description |
|--------------------|--------------------------------------------------------------------------------------------------------------------------------------------|
| Full, Permanent | The “normal” MOC lifecycle. |
| Short, permanent | Changes are routine to the point that documentation can be pre-defined |
| Full, temporary | Changes which are intended to be “undone” prior to a predefined deadline |
| Short, temporary | Changes are routine to the point that the documentation can be pre-defined, yet the change must be removed prior to a predefined deadline. |
| Emergency | Change initiated, usually off-hours, to address an immediate need. Emergency MOC’s must terminate in the creation of a Follow-Up MOC |
| Urgent | Changes needed to overcome imminent danger, where even the Emergency MOC is too cumbersome. |
| Follow-up | A formal documentation of Emergency and Urgent changes. |

Table 1. MOC lifecycles.

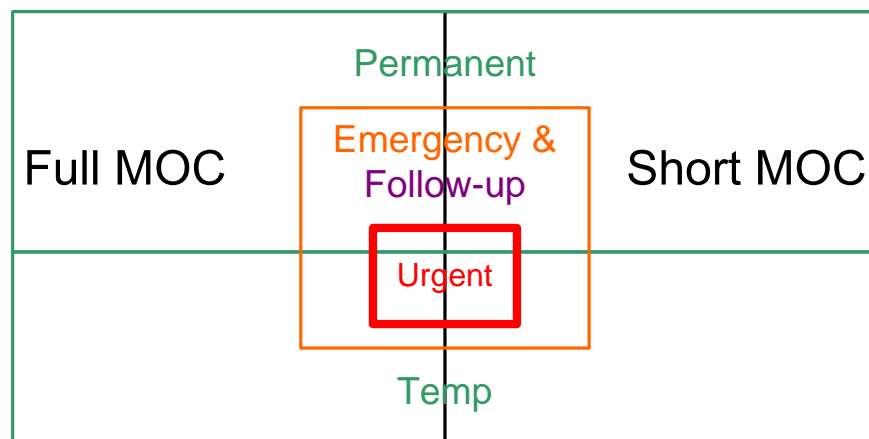


Figure 4. Venn diagram of all MOC lifecycles.

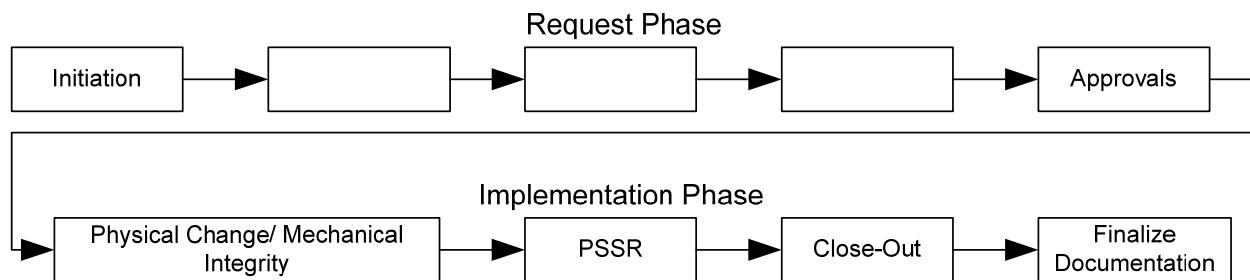


Figure 5. MOC lifecycle model—short, permanent MOC.

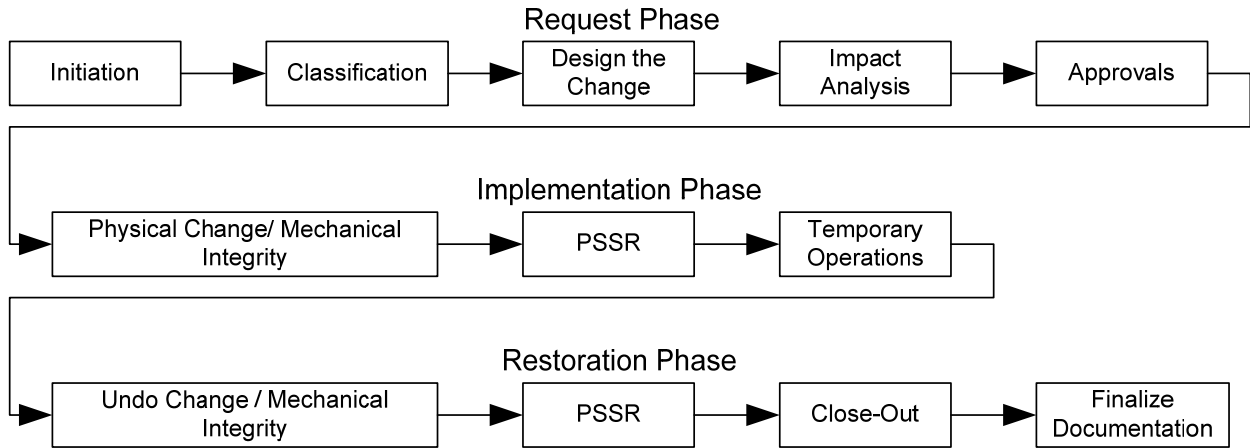


Figure 6. MOC lifecycle model—full, temporary MOC.

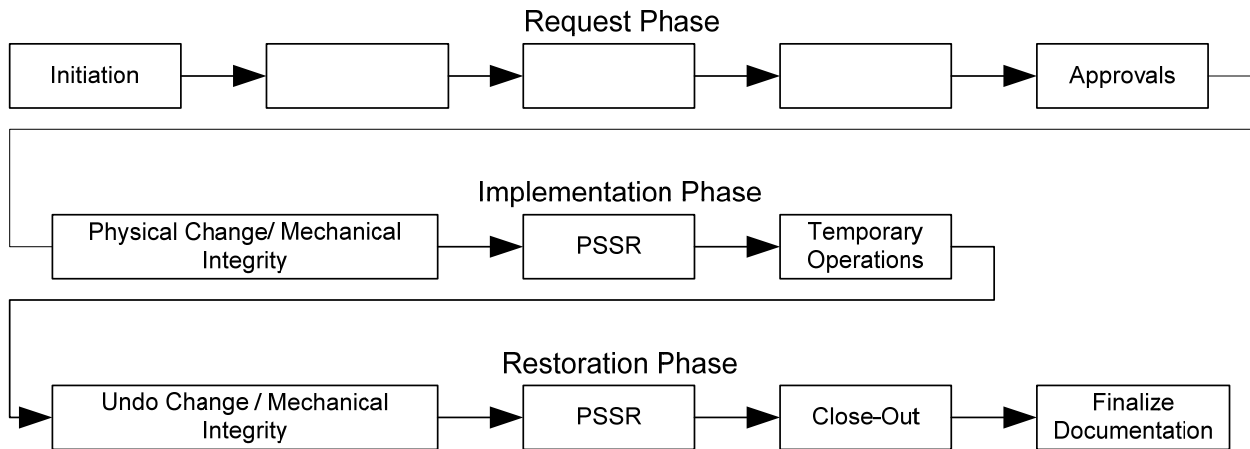


Figure 7. MOC lifecycle model—short, temporary MOC.

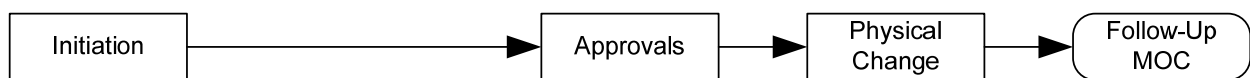


Figure 8. MOC lifecycle model—emergency MOC.

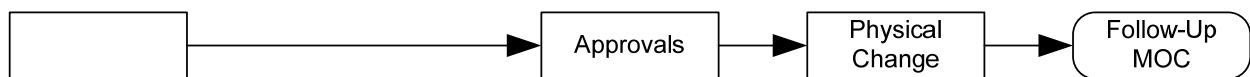


Figure 9. MOC lifecycle model—urgent MOC.