

# Model-Based Systems Engineering: Documentation and Analysis

*Week 4: Managing the Model*

Project

Name

Jane Doe

# Instructions

Before you begin, you should save your project on your local drive. We recommend the following format:

*Lastname\_Firstname\_Course3\_Week4*

**Please note:** You will not be able to re-download your file after submission; therefore, please keep this file in a central location for future reference.

The work in the project deliverable is **individual**.

After you submit your project, you will self-assess your work as well as the work of three peers. If you have any questions, feel free to contact a TA in the Discussion Forum.

Although work is strictly individual, sharing ideas and concepts with other students is encouraged.

**Note: edX has a 10MB file size limit for document submission.** If you have selected large image(s), you may need to resize before submitting, OR you may simply include a web URL for the image in the image location. Be sure to submit your assignment at least one hour before the deadline to provide time for troubleshooting.

**Once the deadline passes, you will not be able to upload the document and therefore will not be able to submit and complete the assignment.**

# Week 4 Project

## Overview

Your assignment for this project is to build a model management plan for an MBSE system. You are encouraged to use the same system that you selected in Weeks 1 and 2.

As you learned in Week 4, there are many important factors to consider when designing a model management plan.

Develop your model management plan and touch on eight key components.

### REQUIRED STEPS

**Step 1:** Model introduction

**Step 2:** Initial verification and validation of the model

**Step 3:** Governing the inputs to the model

**Step 4:** Communicating model results

**Step 5:** Model configuration management

**Step 6:** Defining the model owner

**Step 7:** Funding the model

**Step 8:** Process for model changes over the lifecycle

**Step 9:** Model end of life and renewal planning

**Step 10:** Submit your project and review others' submissions

# Model Introduction

Briefly describe your MBSE approach for which you are creating a model management plan.

The MBSE approach applies to software development for a tech organization developing enterprise software products. Such an organization has many different disciplines that are involved in product development such as software development, designers, testers, product management, customer support, etc. All these disciplines use different tools to get their job done for example software development uses Github as code repository, JIRA for program management, etc. Our MBSE strategy tries to connect all these different software tools so that if someone makes a change that affects the entire project, then those changes could be translated to other disciplines that are involved. i.e our MBSE strategy maintains a single up to date source of information for the entire project team.

## Step 1: Initial Model Verification and Validation

How do you validate the model in the first place? Under what bounds does the model need to be valid?  
Who will validate the initial model?

The MBSE model could be validated by doing a test run on a single project to check if the model is performing as expected. If there are glitches, then the team members could report it and the model could be fixed. Once the performance of the model is found to be satisfactory it could be rolled out to other projects within the organization.

## Step 2: Governing the Inputs to the Model

Will restrictions be applied to each of these inputs? How will bugs and flaws in the model be reported and addressed?

The MBSE strategy will connect to different software used by different teams, there will not be any manual inputs to the MBSE model. It would use webhooks to listen for any updates that team members have posted in the different software that are being used in the project. If it finds any updates, it will notify the users and will propagate the changes to all other software so that everything is in sync. The model can also prevent users from making unnecessary or unapproved changes. For example, before changing a feature specification, the model might ask the user to provide the necessary approval from their manager, which might be integrated as part of the software workflow.

If there are bugs or flaws in the system, then they will be reported to the model owner, who would assign these bugs to the right individuals for fixing. He can also take decisions on workflow updates that change the process of how information is handled through the model

## Step 3: Communicating Model Results

Will the assumptions and caveats be communicated along with the results? If so, how? Will the supporting test cases be communicated along with the results? If so, how?

The model results will be available to all members of the core team, however, only the program manager will communicate model results to the senior leadership. This will ensure consistency of presentation, but it will also introduce delays. The verification results will not regularly be communicated, but they will be held in backup.

## Step 4: Model Configuration Management

How can multiple projects use the same model? Will data restrictions be imposed for different project teams?

The MBSE model would have workspaces for each project. Each workspace will have all the different team members involved in the given project. The data and the notifications for the project will only be visible to the team members working on that project. Updating the information from one software to another will not fall under the workspace restrictions however this should be ok since most of these software have inbuilt authentication and authorization mechanisms for their users..

## Step 5: Defining the Model Owner

What responsibilities does the model owner have (technical, financial, etc.)? What decision rights do they have?

One of the main responsibilities of the model owner is to create linkages between software modules for a given project. For example, a certain task in a project management tool will be linked to the a specific code base for that project. Another responsibility would be to assign authorization access to different team members on a project based on their role and responsibilities for the project. Both the above mentioned responsibilities will be part of the workspaces for projects, to ensure correct information flow for the given project and prevents unauthorized data access. The model owner is also responsible for resolving any change requests or bug reports that may have been requested by team members. The model owner would take the final decision on any modifications or updates to the models resulting from the change requests.

## Step 6: Funding the Model

How will the model be funded?

The model will be funded by the business unit. It is just like any other software tool used by the business unit to efficiently develop their software products. Since the MBSE approach impacts the entire product development organization, it makes sense to fund it from the business unit's budget. This will also be in sync with how software tools used by various teams are funded.

## Step 7: Process for Model Changes Over the Lifecycle

How will changes to the model be re-validated? Who will re-validate them? Is there a need to show traceability to past results? How will they be handled?

The model owner will be responsible for the validation process. S/he will assign model validations task to specific individuals in different teams, potentially involving individuals that raised the change requests. Based on the feedback received, s/he will decide upon whether to pass or reevaluate the change to the model.

There isn't a need to show traceability of past results, since the MBSE model's basic purpose is to act as a bridge that helps information flow throughout the software tools used in product development. The model does not host information itself nor does it house business rules or technical product specifications and for that reason it does not need to allow traceability of past results.

## Step 8: Model End of Life and Renewal Planning

What is the life cycle duration or time horizon for the model? Who is scanning outside to see what new modeling capabilities may be available?

The model is generic, and only acts as a bridge to allow information flow between software tools. It is equally applicable to all the development projects current and future within the organization, therefore, it should have a relatively long lifecycle period, spanning at least 6-8 years. I mention 6-8 years because in the tech industry things change pretty fast--there might be a change in development methodology or the organization might start using new software tools. This might lead to updating or even replacement of the model.

The model owner would be responsible for keeping a lookout on prospective modeling capabilities that might help supplement or even replace the MBSE approach, based on the given needs of the product development process



# Submit Your Project and Review Others' Submissions

Submit your completed Week 4 Project file

- Note: The maximum file size that can be submitted is 10MB.

Assess the work of your peers

- A scoring rubric can be downloaded from the Week 4 Project Instructions page.