# MAXIMO MANAGEMENT OF CHANGE (MOC) IMPLEMENTATION

Best Practices and Recommendations

Presented by Stephen Hume, MaxTECH Chair

# Agenda

1. **Who owns the project and why it should be done**
2. **Where to start & how to nail down the business process**
3. **Identify the Roles**
4. **Define the Business Rules**
5. **Why you need a workflow**
6. **Recommended Configurations to the System**
7. **Implementation Plan**

**Stephen Hume**

Stephen has been working with IBM Maximo for over ten years in a variety of industries (Oil and Gas, Utilities). He has taught Maximo courses to end users for both Technical and Functional audiences and chairs the MaxTECH User Group.

# Background – MOC

Management of Change (MOC) is a business process and therefore needs to be owned by the business. It should not be dictated by the IT Department and forced upon the business. In most organizations, the owner of the MOC process is the Engineering Group. Senior leadership in Engineering must be the drivers and take the ultimate responsibility for implementation of the MOC business process.

**MOC already exists in every organization!**

MOC is not the tool or software, it is a business process and can be managed as a standalone process, on paper, in a spreadsheet, or using other software.

In many industries, MOC is a regulated requirement for health and safety HSE compliance. If you are not doing MOC in your organization, there would be total chaos! Your operations, processing plant, mining activities, refining capabilities, power generation capacity would all be at serious risk of failure. The health and safety of your employees, contract staff, and the general public would also be at risk. Therefore most governments or governing bodies have regulations requiring formal MOC processes to be in place.

# Where Should I Start?

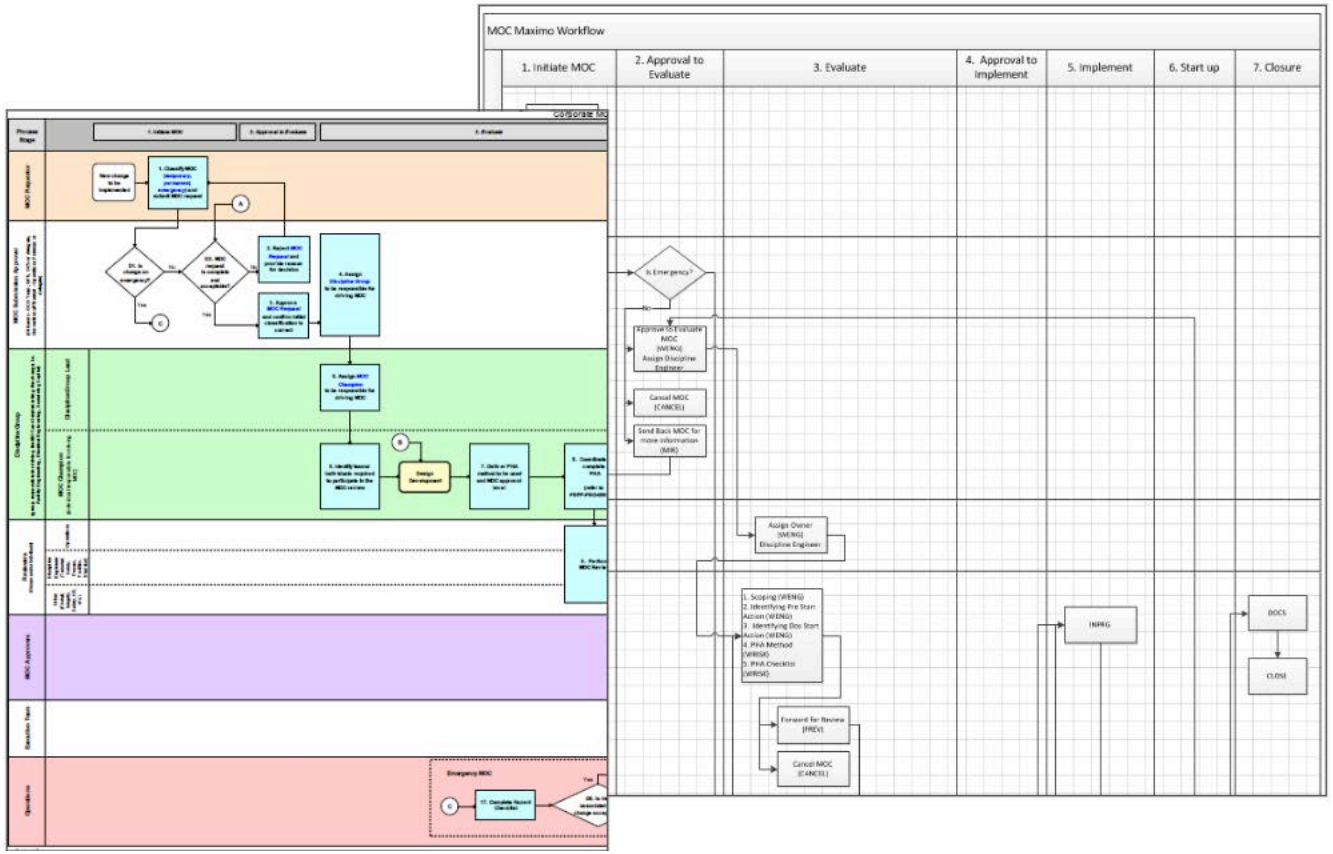
The best place to start is to understand the business process, documenting the flow of information from the beginning to the end of the process. The best way to visualize this is to create a swim-lane process diagram which identifies:

- The main roles in the MOC process
- The hand off of information from one stage of the MOC to another
- Approvals required along the way
- What happens when things are not approved

This process cannot be dictated to the business. Remember, it is owned by the business and they need to be involved in the documentation of the process. Without this, any project to automate/implement software to manage the MOC process cannot proceed. (Or if it does proceed, it will almost likely fail).

Some organizations call this stage in the project “Requirements Gathering”. This stage absolutely must be completed before any Functional and Technical Design can begin. The requirements will drive the functionality, the data requirements and the reporting requirements from the system that is being implemented.

# Where Should I Start?

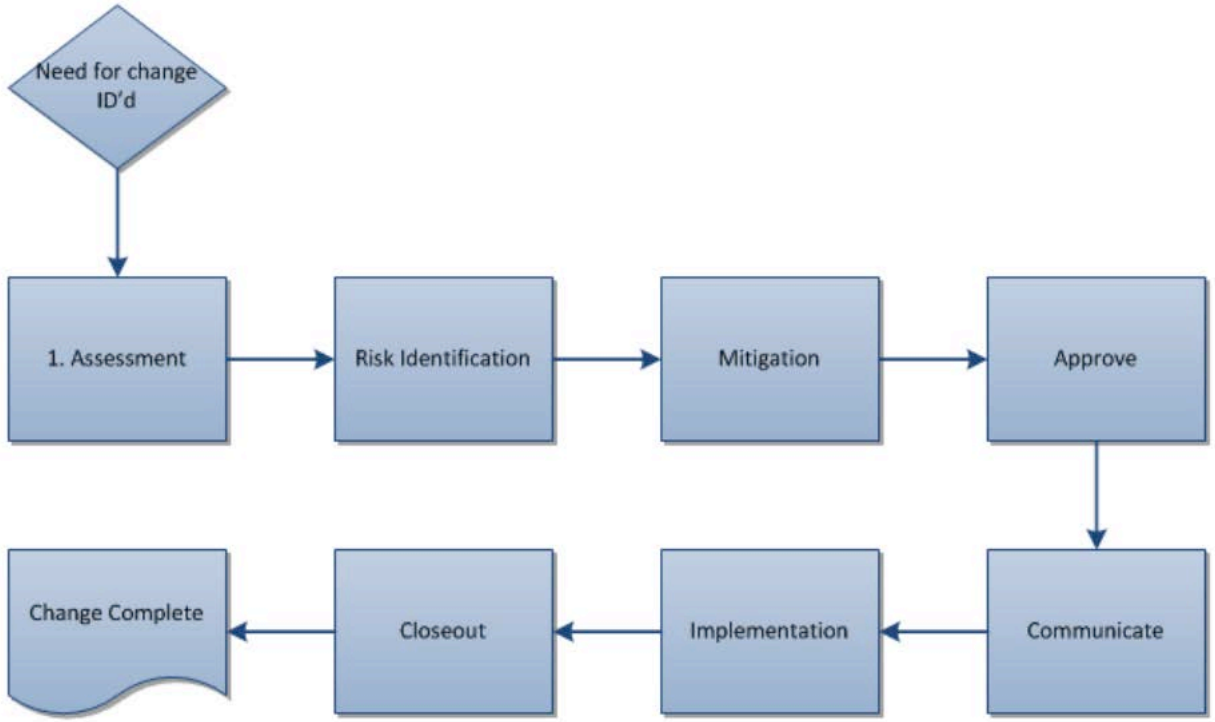



REMEMBER

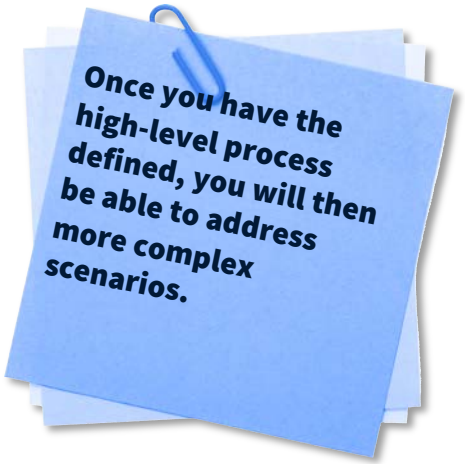
**MOC is the Process, not the Software**

# Step by Step: Define the Process

Start Simple – single thread high level process:



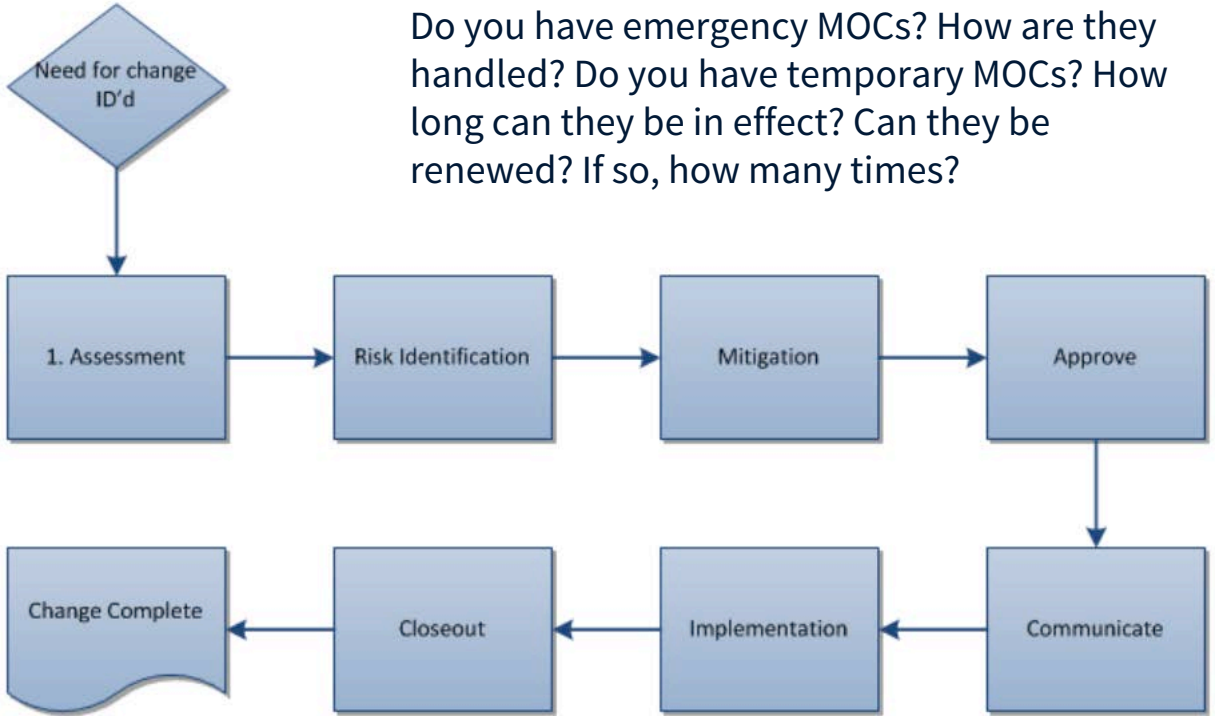

TIP **Identify the people or roles associated with each task in the process**



# Step by Step: Refine the Process

Then add more complexity to the process:

TIP **Approvers need to be able to delegate that authority**




Do you have emergency MOCs? How are they handled? Do you have temporary MOCs? How long can they be in effect? Can they be renewed? If so, how many times?

**Answering these questions will start to define the Maximo configuration that will be required.**

# Step by Step: Identify the Roles involved

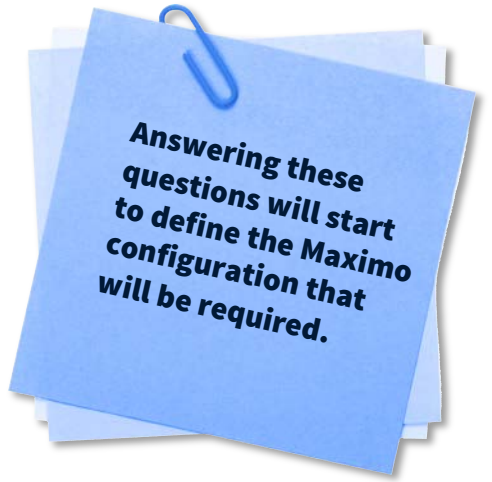
**Owner**–Who is the owner of an MOC, does ownership change during the life of the MOC, does being the owner of the MOC give any special privileges (edit abilities, routing workflow abilities)?

**Owner Group** –Are MOCs ever sent to a group to deal with, if so, who is in the group(s), do MOCs get handed off from one group to another?

**Reviewers**–When an MOC is first created, the out-of-the-box process is that it is sent to one or more reviewer(s)? Those people review the MOC, and if the information is complete, they can indicate that they have reviewed. Typically once the last person has reviewed the MOC, it can move to the next stage in the process.

**Approvers**–When does approval need to be completed for your MOC? Once all the engineering has been completed and the MOC has been updated? Or is this seen as an approval to proceed with the engineering. Approval means different things to different companies.

**Closure Reviewer** –When the MOC has been completed, who reviews the information and determines that the MOC can be closed?




TIP **Approvers and Reviewers and Owners need to be able to delegate that authority**

# Step by Step: Identify the Roles involved

Some organizations have identified additional MOC Roles.

Do they apply in your case and are there even others?

- Discipline Lead
- Champion (another name for owner)
- Approvers to Evaluate
- Implementation Approvers
- Start-up Approvers
- Temporary MOC Extension Approvers
- Process Hazard Assessment Reviewer(s)
- Pre-Start Action Completers

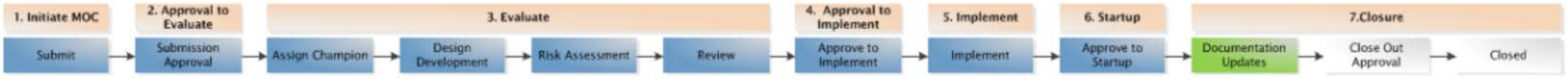
# Step by Step: Define the Business Rules

Maximo delivers the ability to have certain statuses for MOCs throughout their life cycle. There are two approaches that can be taken when implementing an MOC solution:

1. Add new statuses as required to identify the state of the MOC. This can get very complex and one of the main issues with this approach is that the MOC Status and the Work Order Status are a shared list -so when you add statuses for MOCs, they will appear in the Work Order application unless you conditionally hide them.

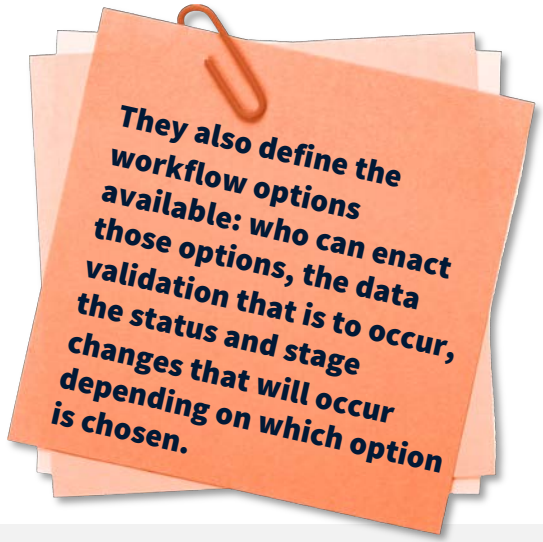
2. Keep the list of statuses for an MOC to a minimum and add a second attribute called “STAGE”. A single status of an MOC can have several stages. Changing stages is completed by workflow which will keep Status and Stage in synchronization. (Another reason why manual status changes should be disallowed or at the very least extremely limited!).

This Progress Map is an example of statuses and the related stages of an MOC:



# Step by Step: Refine the Business Rules

Business Rules define the condition of the MOC application at every stage of the MOC:




Status : WAPPR - done
Stage: SUBMIT
User = REPORTEDBY
Category: Permanent/Temp/Emergency
Workflow Options:

1. Submit MOC for Approval to Evaluate (SUBMIT)
2. Cancel MOC (CAN)

Workflow Checks:

1. Must be at least 1 PLUSGMOCRVLST.REVIEWER
2. Must have an MOC Category Selected

Actions:

1. Status : SUBMIT, Stage: SUBMISSION APPROVAL
2. Status: CAN

Application Conditions:

1. MOC – Open to all
2. Review and Approve – Read-only except PLUSGMOCRVLST.REVIEWER on the Approval to Evaluate section need ability to add new row
3. **PHA Method Determination – Read-only to all**
4. **PHA Checklist Methodology – Read-only to all**
5. Action Tracker - Read-only to all
6. Pre-Start Actions - Read-only to all
7. Documentation/Closure - Read-only to all
8. Related Records - Read-only to all – except ability to add link to existing work orders
9. Change Log – Open to All
10. MOC Summary – Open to All

# Why do you need Workflow?

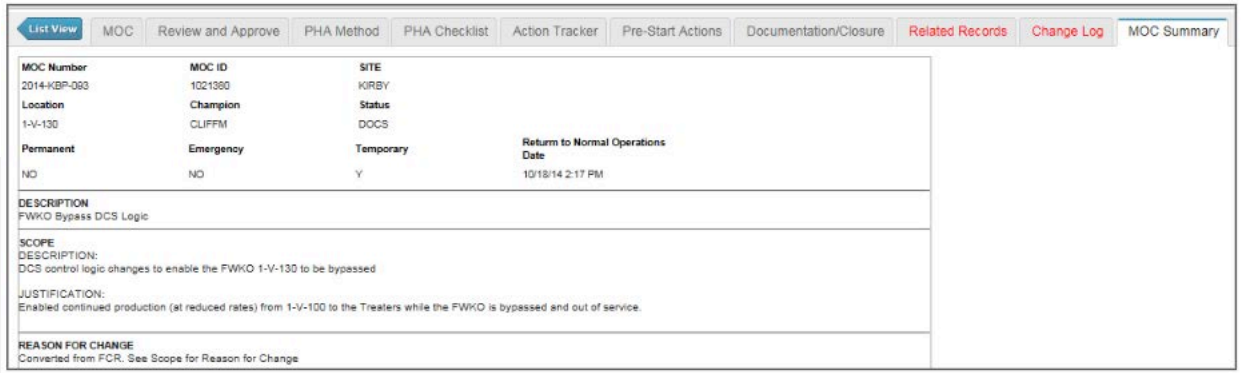
An automated workflow for the MOC application is vital for the following reasons:

1. It controls who can advance the MOC to the next stage.
2. It validates that all the data needed in the MOC has been completed before moving to the next stage.
3. It will keep the MOC Status and Stage in perfect synchronization.
4. It can send email notifications to people or groups of people when an MOC has been “sent” to them for action.
5. If can visually tell the user when they have missed something, or they should not be clicking the workflow icon.


6. A workflow can generate actions to do things like create corrective work orders, generate incident reports, spawn another MOC, all when and as required.
7. The workflow can set other MOC Values such as reviewed dates, approval dates, checkboxes, etc. Just by answering a question posed by the workflow the system can do those updates.
8. Workflow is the key function that will ensure data completeness and integrity in the MOC application.

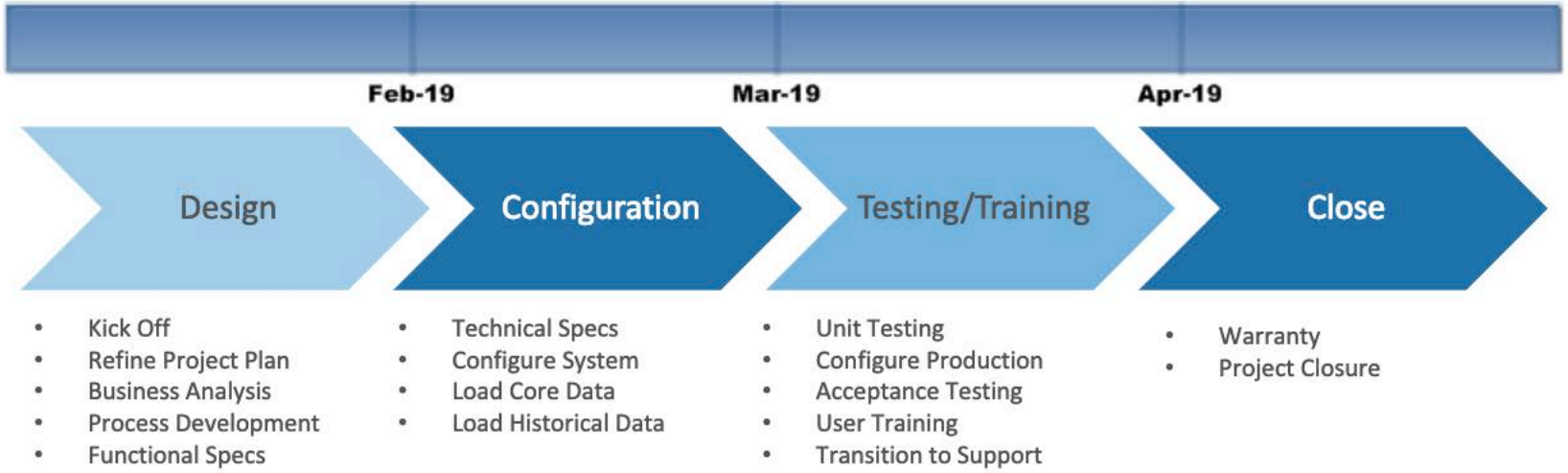
# Recommended Configurations

1. Do not implement MOC without an automated workflow.
2. Add an MOC Stage attribute to the Work Order table.
3. Set up the conditional expression to identify when a user is a delegate of an approver or they are the approver, then use that condition to set checkboxes to read only or editable.
4. Add a rejected attribute to review and approver lists, keep a record of rejections as well as approvals.
5. Go beyond the “review, approve” lists and add whichever lists you need for your MOC workflow.
6. For certain dates in the MOC (such as return to normal operations date), create an automation script to disallow future dates.
7. Add a Progress Map to the MOC Details tab.
8. Add an MOC Summary Tab, which uses an automation script to display all MOC data on a single tab (great for reviewing the MOC Data).
9. Colour the Change Log and Related Records tabs if there is data present:



# Implementation Plan

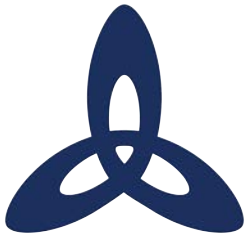
This a high-level implementation plan for the MOC Application. It includes process definition, functional specifications, technical specifications, data conversion and loading, system configuration, unit and acceptance testing.



# THANK YOU!

Success! I hope that this MOC solution proves useful in any future projects.

MAXTECH



**Ask us a technical question:**

maxtech@bpdzenith.com

www.bpdzenith.com/maxtech

MaxTECH is the first ever dedicated Maximo Technical User Group aimed at Maximo Administrators, Developers and Technical Support staff.

It is a great place for users to ask and answer technical questions, learn from each other, collaborate and help improve Maximo in your organization.

MaxTECH was founded in 2017 by BPD Zenith and is chaired by Maximo Consultant Stephen Hume. We host several events every year (Calgary, St. Louis, MaximoWorld, Houston, MUWG, Northern California MUG, Maximo UK & Ireland User Group) including digital events.

MaxTEACH is a free online user group designed to go in depth into a Maximo topic.