Welcome to the Winter edition of the APMD Newsletter. The most significant Society event since our last newsletter was the Annual Leader meeting in October which was attended on our behalf by Director-elect Rory who provided the following synopsis of the several days of meetings.

Everyone who attended our Annual Leaders Meeting in San Diego, California in October will remember the warm weather, productive meetings and of course the great beach party. At the I&S and A&T meeting we gave an update to all other Divisions on the progress we have made in our inaugural year and our plans for the year ahead in 2020. As a new Division we were commended for quickly recruiting and regularly meeting with a respectable number of members who are engaged and actively partaking in our special project.

I was delighted to receive an award for communication on behalf of our AMPD Division. It was great to have our continued communication efforts recognized by the Society. We look forward to reporting back against our business plan throughout the year ahead and meeting you all again at the 2020 Annual Leaders Meeting in Puerto Rico.

Between now and then however we continue to work on the APMD Guideline document. To give you an idea of the type of information being included I have prepared a technical article summation of one of the Chapters of the document inside this issue.

Don’t forget the Division has its own LinkedIn group at https://www.linkedin.com/groups/8638297/, its own website at http://isaapmd.com/ and of course the ISA.org website: https://www.isa.org/apmd/

A listing of all the Division leaders can be found on the Division website and at the end of the newsletter.

Andre Michel
ISA, APMD Director
Amichel300@gmail.com
Director-elect Update

Rory Moloney

An understanding of typical automation projects across industries is a key requirement for this Division to prepare and provide a project management methodology specifically for the management of automation projects.

Along with the very familiar DCS upgrades and more traditional system integration projects, we now see that organisations in their droves are in a constant state of planning and executing cyber security projects. Cross-Industry organisations are now more than ever taking steps to protect their automation systems and environments from an ever-growing number of threats.

Deployment of solutions to prevent, detect and react to cyber threats are now representative automation projects that are commonplace across industries. The implementation of these protective solutions within a standard site landscape of leading edge and legacy systems, varying infrastructure and diversely skilled human resources presents a not so common set of very OT specific constraints.

In the Automation Project Management and Delivery Division, we address these OT specific constraints throughout the lifecycle of the project in the project management methodology that we are developing and look forward to sharing with the ISA community very soon.

Best regards, Rory

Editor’s Corner

Ian Verhappen

Hello and welcome to the Automation Project Management and Delivery Division newsletter.

We are planning to issue a minimum of two newsletters per year with each newsletter containing the following content and would like feedback on what you would like to see in this YOUR newsletter. Please get a hold of myself, Andre, or Rory with your suggestions.

Thank you to Andre, Rory and Simon for their contributions to this issue. Of course, if you the members wish to provide additional information that you think would be of interest for future issues please send it along.

Looking forward to hearing from many of you.—Ian

Is it time to join a committee?

Serving on a committee is a great way to help our Section and to sharpen your leadership skills. We have over a dozen committees, from education to exhibits to treasurer, and more. Surely there’s a good fit for you!

Just let any of our leaders know the committee in which you are interested. Volunteering on a committee is a great way to learn new skills while you network with other automation professionals. Our Section benefits, and you do, too!
What makes Automation Projects Challenging?
Automation and Process Control projects are challenging and can often invoke fear for the newly appointed Project Manager or Project Engineer in charge that may not have an automation background. In addition to all the usual challenges of any project, this particular type of projects usually come with unique challenges and level of complexity. This is the result of a variety of reasons including; integration of multiple niche engineering disciplines (software, computer systems, networks, cybersecurity and instrumentation), high dependency on User Requirements of disciplines outside of automation, integration of requirements from many areas (business, engineering, safety, environmental), delays in other areas of the project that propagate over into the automation area, new technology or technology constraints, and many more. In addition, Automation can often be viewed as a black box by the non-automation community, which tends to isolate these activities and people from the broader project structures and support. Lastly, Automation projects by the very nature of what you are trying to build are complex. It entails building an embedded intelligence into an integrated set of computer and people systems, in many cases to run a manufacturing process that very often has never been done.

An adequate delivery methodology guide to manage automation projects and their complexities can make the process easier. This doesn’t always mean huge project teams and large budgets, in fact, many times, quite the opposite. By having a systemic approach to delivering automation projects, these challenges can be effectively managed; project teams can be more effective (and thus be smaller and less expensive) and can be routinely executed on schedule. However, if implemented poorly, Automation projects can cripple an overall project (and a facility for that matter) from being successful, or in some cases, even functional.

This document is the result of the combined automation project delivery experience of multiple automation professionals that were willing to share their experience and best practices. In addition to an overall delivery methodology, we will share many practical tools and templates that we use every day in our own ongoing project work. By sharing these methods and tools, we hope to drive a continuing dialogue within the small close-knit automation community about how to best deliver these projects, and to foster a community that shares best practices with one another.

We will try to interrelate two basically different disciplines and allow the reader to get the best of both: Project Management and Automation Engineering.

Project Management as a discipline started at the end of the 1800’s and was basically led by 2 Americans: Henry Gantt and Henri Fayol. Gantt is very well known for Gantt chart that we are still using every day when you look at your schedule. Fayol is recognized for the creation of the six-management functions which form the basis of the project management body of knowledge. Both were students of Frederik Taylor who initiated the WBS (Work Breakdown Structure) concept. In the 1950’s several scheduling techniques were develop include CPM (Dupont and Remington) and PERT (US Navy and Lockheed). Today most PM concepts are well distributed and are mainly popularize by two main organization: PMI and AACE.

On the other hand, Automation derives from the arrival of
mass production initiated in the early 20th century by Henry Ford, technically automated equipment existed in the 1800’s basically using cams to reproduce a specific sequence of events. Numerical control arrived in the 1940’s and exploded in the 1950’s. The 1970’s introduced PLC’s and DCS. MES arrived in the 1990’s along with the expert systems. Today automation is seen as the foundation of the new digital world in manufacturing (Industrie 4.0 and smart manufacturing)

A unified approach to automation project delivery

All roads lead to Rome. We do not believe there is only one way to efficiently manage your automation project. We are certain that the same method does not apply to 2 different companies or even in the same company in 2 different groups. Culture and economic climate are only some of the factors which would make the same method fail even in similar circumstances.

However, we do believe that the development of an optimum delivery strategy is a very similar exercise from one project to another and that it can be relatively easily described. The path to reach a better delivery method is what is important, not the method itself.

Key aspects of our approach include:

- Developed and implemented by end users
- Scalability and flexibility to manage different types of vendors/suppliers, including Engineering Companies and System Integrators
- Extensively used on many projects of different sizes and types
- When properly applied and executed, results are consistent and predictable schedule and budget outcomes

The Key aspects of this methodology include:

- Breaking projects into distinct phases,
- Breaking work down into distinct deliverables,
- Providing periodic reviews of progress, and
- Using metrics and measures to track and execute progress.

Automation Project Management Lifecycle

A project is a finite endeavor (having specific start and completion dates) undertaken to create a unique product or service which brings about beneficial change or added value. (PMBOK reference)

A project life cycle is defined as a series of phases that the project will go through during its life. Typically, project development includes several phases where the focus on the activities differs. Automation like other disciplines has its life cycle affected and impacts in turn other discipline’s life cycles. Automation has the particularity of being a lagging discipline and has many dependencies on earlier disciplines.

Chapter one of the document under development defines an automation life cycle that would fit most projects.

Automation Lifecycle

Below is a cross-functional chart (aka swim-lane) which displays the phases of a project in the x axis and each discipline and sub discipline in the Y axis. Within the chart, deliverables are identified. This is a powerful view to be able to see who (discipline) will produce what (deliverable) when (project phase).

Some other disciplines, which might affect automation, and therefore, might need to be included are:

- Process
- Mechanical
- Civil
- Structural
- Electrical
- HVAC
- Piping
- Project PM
- IT/OT

The potential Automation sub-disciplines are:

- I&C
- Hardware Engineering and Software engineering

Project Phases

(Continued on page 5)
One of the organizing principles of the automation lifecycle in Figure 1 is the concept of phases.

We are using the following phases and will continue to do so for the rest of this document.

Phase 1 – Feasibility, justification, profiling
Phase 2 - Conceptual Engineering, planning ...
Phase 3 – Preliminary engineering
Phase 4 - Basic Engineering, design
Phase 5 - Detailed Engineering, system design, development
Phase 6 – Construction, execution, deployment
Phase 7 – Commissioning, testing
Phase 8 – Start-up, Qualification, go live and Startup
Phase 9 – Closeout, Production, Turn over, Operation, support

The benefits of breaking projects into phases are:
- The project has intermediate milestones on which to focus
- The resource requirements for the various phases are different. By having different phases, project staffing can be more easily understood and applied (prevents overstaffing in the beginning and /or understaffing in the end.)
- Allows for an orderly progression of the engineering process
- Defines Relationships and the evolution of the deliverables

Table 1 summarizes the above information with an estimate of the projected overall project accuracy.

The phases are not perfectly sequential and might be slightly different from one discipline to another. As a lagging discipline, Automation (the User Requirements are derived from other disciplines design), it is not unusual to start a new phase for certain deliverables when the previous phase is not finished.

A useful industry tool to classify project estimate accuracy is the AACE recommended Practice 18R-97 “COST ESTIMATE CLASSIFICATION SYSTEM – AS APPLIED IN ENGINEERING, PROCUREMENT, AND CONSTRUCTION FOR THE PROCESS INDUSTRIES” summarized in Table 2- AACE Estimate Classes

**Deliverables – How to keep focused**

Within the Swimlane diagram, each phase contains deliverables. Deliverables differ from activities in that they are time dependent and sequential. Early deliverables support the creation of the project concepts. In automation

(Continued on page 6)
we talk about Automation Philosophy and User requirements.

As the concepts materialize, deliverables will become clearer and will turn into specifications (Functional and Design Specifications).

Activities are associated with the action required to produce each of the deliverables. Some activities might lead to intermediate milestone which are required to produce the deliverables. Beware of activities that do not lead at some point to a deliverable. This lack of relationship usually means they are unnecessary. A typical example is when the project team says: "We have always done it this way". Maybe at some point there was a procedure requiring a deliverable which is not required anymore but the team keeps doing the activity.

It is much easier to measure deliverables than it is to measure activities. Therefore, we are using

(Continued on page 7)
Table 2 - AACE Estimate Classes

<table>
<thead>
<tr>
<th>Estimate Class</th>
<th>Primary Characteristic</th>
<th>Secondary Characteristic</th>
<th>Methodology</th>
<th>Expected Accuracy Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class 5</td>
<td>0% to 2%</td>
<td>Concept screening</td>
<td>Capacity factored, parametric models, judgement or</td>
<td>L: -20% to – 50%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>H: +30% to +100%</td>
</tr>
<tr>
<td>Class 4</td>
<td>1% to 15%</td>
<td>Study or feasibility</td>
<td>Equipment factored or</td>
<td>L: -15% to –30%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>H: +20% to +50%</td>
</tr>
<tr>
<td>Class 3</td>
<td>10% to 40%</td>
<td>Budget authorization or control</td>
<td>Semi-detailed unit costs with assembly level line</td>
<td>L: -10% to –20%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>H: +10% to +30%</td>
</tr>
<tr>
<td>Class 2</td>
<td>30% to 75%</td>
<td>Control or bid/ tender</td>
<td>Detailed unit cost with forced detailed take</td>
<td>L: -5% to –15%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>H: +5% to +20%</td>
</tr>
<tr>
<td>Class 1</td>
<td>55% to 100%</td>
<td>Check estimate or bid / tender</td>
<td>Detailed unit cost with detailed take-off</td>
<td>L: -3% to –10%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>H: +3% to +15%</td>
</tr>
</tbody>
</table>

(Continued from page 6)

deliverables as our key management tool during project execution.

Activity is how you achieve a deliverable. Deliverable is the desired outcome.

**Deliverable List**

To get a good plan for the work, a list of deliverables must be developed. It is not necessary to define every single deliverable at day one of the project; you can focus on the phase of work you are in (and the next one for planning purposes). Identify the deliverable and the level of completion each deliverable will need to achieve (i.e. draft, issue for review, reviewed, issue for approval, or approved) for the end of the phase.

This deliverables-based approach will also be useful when you go to model your project plan in your estimate, schedule, and project metrics.

Table 3 is a typical example of some automation deliverables broken into the various phases.

**Work Breakdown Structure**

A work breakdown structure (WBS) is a deliverable oriented hierarchical decomposition of the work. (PMBOK reference). A WBS also provides the necessary framework for detailed cost estimating and control along with providing guidance for schedule development and control.

The Work Breakdown Structure is a tree structure, which shows a subdivision of effort required to achieve an objective; for example, a program, project, and contract. In a project or contract, the WBS is developed by starting with:

- the end objective and

(Continued on page 9)
### Table 3 - Examples of Deliverables

<table>
<thead>
<tr>
<th>Phase</th>
<th>Input from other disciplines</th>
<th>Automation Deliverables</th>
</tr>
</thead>
<tbody>
<tr>
<td>Conceptual Engineering</td>
<td>Process Description</td>
<td>User Requirement Specifications</td>
</tr>
<tr>
<td></td>
<td>Process Flow Diagram</td>
<td>Automation Philosophy</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Automation Scope of Work</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Execution: EPC, MAC, SI</td>
</tr>
<tr>
<td>Preliminary Engineering</td>
<td>Piping and Instrumentation</td>
<td>Hardware and Software Functional Specifications.</td>
</tr>
<tr>
<td></td>
<td>Diagram (P&amp;ID)</td>
<td>System Architecture</td>
</tr>
<tr>
<td></td>
<td>Air Flow Diagrams (AFD)</td>
<td>Programming Standards</td>
</tr>
<tr>
<td></td>
<td>Process description</td>
<td>Automation Schedule</td>
</tr>
<tr>
<td></td>
<td>General Arrangements</td>
<td>Automation estimate</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Automation Project plan</td>
</tr>
<tr>
<td></td>
<td></td>
<td>System Integration Contract</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Preliminary Hardware Bill of Material</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Instrumentation Standards</td>
</tr>
<tr>
<td></td>
<td></td>
<td>I/O list</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Project Plans (Execution, Quality, Risk, Procurement, Communication, etc.)</td>
</tr>
<tr>
<td>Detailed Engineering</td>
<td>Commissioning plan</td>
<td>Hardware and Software Design Specifications’s Final BOM</td>
</tr>
<tr>
<td></td>
<td>Detailed plan layouts</td>
<td>Procurement packages (Panels, Instruments, Networking components)</td>
</tr>
<tr>
<td></td>
<td>Other disciplines</td>
<td>Construction Drawings (loop, networks, panels, others)</td>
</tr>
<tr>
<td></td>
<td>construction drawings</td>
<td>Control Narrative &amp; Software /HMI development</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Change Management Plan</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Training packages</td>
</tr>
<tr>
<td>Construction</td>
<td>Installed equipment</td>
<td>Automation Software and Simulation Software</td>
</tr>
<tr>
<td></td>
<td>Overall commissioning plan</td>
<td>Written Test protocols</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Executed Test protocols</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Factory Acceptance Test Protocols and Results (FAT’s) (panel and others) (aka CAT’s –</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Customer Acceptance Test)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Installed hardware</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Training Package</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Commissioning protocols</td>
</tr>
<tr>
<td>Commissioning and Start-Up</td>
<td>Commissioning Schedule</td>
<td>SAT</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Executed commissioning protocols</td>
</tr>
<tr>
<td></td>
<td></td>
<td>As built information and closeout files</td>
</tr>
</tbody>
</table>
successively subdividing it into manageable components
in terms of size, duration, and responsibility (e.g., systems, subsystems, components, tasks, subtasks, and work packages)
and which include all steps necessary to achieve the objective.
The WBS provides a common framework for the natural development of the overall planning and control of a contract and is the basis for dividing work into definable increments from which the statement of work can be developed and technical, schedule, cost, and labor hour reporting can be established.

A WBS permits summing of subordinate costs for tasks, materials, etc., into their successively higher level "parent" tasks, materials, etc. For each element of the WBS, a description of the task to be performed is generated. This technique (sometimes called a System Breakdown Structure) is used to define and organize the total scope of a project.

The WBS is organized around the primary products of the project (or planned outcomes) instead of the work needed to produce the products (planned actions). Since the planned outcomes are the desired deliverables of the project, they form a relatively stable set of categories in which the costs of the planned actions needed to achieve them can be collected. A well-designed WBS makes it easy to assign each project activity to one and only one terminal element of the WBS. In addition to its function in cost accounting, the WBS also helps map requirements from one level of system specification to another, for example, a requirements cross reference matrix mapping functional requirements to high level or low-level design documents.

WBS is covered in more detail in Chap 3 of the Guide.

(Continued from page 7)
Establishing the precedence between deliverables and activities

When establishing the delivery model, predecessor and successor relationships should be discovered and represented in the model. Some Typical Relationships for Deliverables are Start to Start, Finish to Start. Often there are multiple predecessors and successors.

The Delivery Model and Automation Lifecycle

The value of the swim-lane diagram (Figure 1 - Automation Cross functional chart aka swimlane) is that it is a concise visual representation of what (Figure 3 - Example of a WBS), when (chronology with predecessor/successor relationships), and who (who performs the work), all available for review and understanding of the entire project team (including those outside of automation). This swimlane view can really illuminate the many aspects of automation, how much is typically involved, as well as the complex timing arrangements, and start to peel away the veil of the black box for those outside of the automation discipline. the swimlane also provides a good high-level view for the automation engineers working on the project, so that they can see how their individual piece fits into a broader picture.

We now have a visual representation of the automation life cycle. For the same company in the same environment, this life cycle should be very similar and requires very little adjustment form one project to another.

Another valuable purpose of the automation lifecycle is that once the delivery model is well understood it will serve to create your first schedule. It represents the first draft of your PERT (network diagram in scheduling but that has nothing to do with ours). Many engineers try to jump directly to creating a schedule, without having clarity on deliverables, timing, and assignments. Unfortunately, years of experience have proven that this approach will generally fail as the creator spirals down into minute details on a certain task but will forget the big picture. Furthermore, to introduce the time factor too early can generate a never-ending discussion around timing when the focus must be on the sequence of events. Timing will come naturally if the sequence is well understood. The final execution date cannot be determined if the process is not well understood and clear to all project participants. We will expand on this subject elsewhere in the Guide.
BACKGROUND & CONTEXT

The HAZOP design verification process was developed by ICI UK LLC, a large petro-chemical company, in the mid-1960s in response to a series of serious process industry incidents. The use of HAZOP gained more traction in the industry after incidents such as Flixborough (1974) and Seveso (1976). It is a scenario/guide-word based hazard identification multi-disciplinary process applied to green-field, brown-field and revamp projects. Depending on size and complexity it is also applied to plant modifications. Part of the HAZOP process was to look at operability issues including the instrument and control aspects.

Whilst HAZOP was a very useful tool, limitations became apparent by the 1980s as automation became more complex. Plants were being run closer to the safety boundaries in order to attain higher efficiencies and lower costs. This was made possible by the changeover from pneumatic to the new digital controls and instrumentation. Unfortunately, the rate of incidents at this stage didn’t improve. There were more incidents caused by complex interactions in the automation design. These failures were often caused by erroneous and/or inadequate design requirements rather than actual physical equipment failures. HAZOP wasn’t up to the task of assessing the modern and more complex automation designs. The typical guide words from HAZOP, as shown below, aren’t very extensive and can’t adequately cover more complex automation strategies.

**Instruments**
- Sufficient for control
- Too many
- Correct location

**Electrical**
- Area Classification
- Isolation

**Plant Items**
- Operable
- Maintainable

**Testing**
- Equipment (including trips and alarms)
- Product

BENEFITS OF CHAZOP

The Need for CHAZOP; more details

The use of programmable electronic instrument and control systems (sometimes called PES for Programmable Electronic Systems) has now become standard in the process industry. These systems include distributed control systems (DCS), Programmable Logic Controllers (PLCs), Supervisory Control and Data Acquisition (SCADA) systems and Safety Instrumented Systems (SIS) designed and implemented to conform with IEC61508 and IEC61511.

The trend to control systems possessing greater flexibility and power is accelerating with the introduction of standardized IEC 61558 fieldbus communications protocols such as Foundation Fieldbus, Profinet and ControlNet between field instruments and the PES, the growing use of highly programmable and powerful “smart” field instruments, wireless field instruments, intelligent field control stations and Windows-based operator stations.

Many hazards and mal-operations are unintended behavior of the control system in response to unforeseen scenarios. These software / hardware / process interactions can be very difficult to predict and even more difficult to diagnose when they occur. They can often be mistaken as "intermittent" hardware faults.

However, there is now the realization, often through bitter experience, that the control design intention is not only difficult to specify unambiguously but can be thwarted by the failure of the system in ways which were previously not possible with discrete panel-based controllers or hard-wired relay systems. Major disasters such as Longford (1998), Texas City (2005) and Buncefield (2005) were all at least partly caused by control system problems.

The CHAZOP Technique

Building on the successful HAZOP technique, CHAZOP provides a systematic, multi-disciplinary technique to improve the safety.
and operability of computer-based control systems. It is used as complementary FMEDA (failure mode & effects diagnostics analysis) tool after the completion of HAZOP to refine the automation design specification.

Depending on the application, there are four different types of CHAZOP which can be tailored for a given project.

The Preliminary CHAZOP—carried out prior to detailed design to ensure the specification will meet the User’s requirements.

The System CHAZOP—provides a review of the “Control System”, including the network, electrical supply, I/O, hardware, internal and external data and network connections, environmental factors and cyber-security.

The Loop CHAZOP—reviews loop functionality such as continuous and on/off control, open/close valve operations, and motor start and stop, confirming the design intent and investigating the impact of failures on safe operation.

The Sequence CHAZOP—is a technique for reviewing “sequential control”, such as start-ups, shutdowns, batch processes, time-based operation and similar logic.

Focus where it’s needed.

By selective application of the different CHAZOPs, the design team can drill down to those areas with greatest impact – going beyond the system architecture to analyze keys loops or sequences.

The Benefits of CHAZOP

It’s been shown by the UK Health and Safety Executive that most causes (and an even greater proportion of cost) of failures with control systems is due to errors in the specification of the control task.

The application of CHAZOP allows the design intent to be refined and communicated clearly to those responsible for execution. This delivers the following benefits:

Supports templating

- By its nature the CHAZOP process directs the design to using configuration templates. The team is better able to identify, select and classify more effective templates.

Faster and more effective testing

- CHAZOP usually results in better understood and more effective design specifications. This, in turn, leads to better acceptance testing. Very frequently, testing (e.g., FAT) fails due to incorrect and/or incomplete requirements rather than simple coding errors.

Faster commissioning

- The commissioning example of a new DCS at a plastics factory with four similar reaction streams went so well that during the commissioning there were no issues for the night shift support engineers.

- Through CHAZOP, the benefit of better standardization and alignment of control templates becomes clear, leading to a much more effective design phase of the project. This also streamlines start-up acceptance testing and sign-off.

Greater acceptance by operators

- By being part of the CHAZOP team, the operating crew can help make the system more user friendly.

Fewer spurious trips and less downtime over the life of the plant

- The systematic, multidisciplinary review uncovers more unusual scenarios, to reduce the chance of an unexpected event not being considered in the system design.

A safer plant

- IEC61508/61511 requires an independent review of the Safety System and Safety Instrumented Functions. A CHAZOP review provides this requirement and at the same time helps ensure the Safety System doesn’t interfere with the beneficial operation of the plant.

Cyber Security

- Systematic consideration of a control system helps with the identification of potential vulnerabilities to facilitate and enhance a more detailed Cyber Security study to performed.
# APMD Leadership Team

<table>
<thead>
<tr>
<th>Position</th>
<th>Person</th>
<th>Email</th>
</tr>
</thead>
<tbody>
<tr>
<td>Director</td>
<td>Andre Michel</td>
<td><a href="mailto:Amichel300@gmail.com">Amichel300@gmail.com</a></td>
</tr>
<tr>
<td>Director-elect</td>
<td>Rory Moloney</td>
<td><a href="mailto:rmoloney@ITS.JNJ.com">rmoloney@ITS.JNJ.com</a></td>
</tr>
<tr>
<td>Past Director</td>
<td>Ardis Bartle</td>
<td><a href="mailto:ardisbartle@apexmeasurement.com">ardisbartle@apexmeasurement.com</a></td>
</tr>
<tr>
<td>Program Chair</td>
<td>Trung Nguyen</td>
<td><a href="mailto:trung2704@gmail.com">trung2704@gmail.com</a></td>
</tr>
<tr>
<td>Membership Chair</td>
<td>Michael Baldauff</td>
<td><a href="mailto:michael.baldauff@fujifilm.com">michael.baldauff@fujifilm.com</a></td>
</tr>
<tr>
<td>Newsletter Editor</td>
<td>Ian Verhappen</td>
<td><a href="mailto:Ian.Verhappen@cima.ca">Ian.Verhappen@cima.ca</a></td>
</tr>
<tr>
<td>Standards &amp; Practices</td>
<td>Maurice Wilkins</td>
<td><a href="mailto:wilkinsmj@gmail.com">wilkinsmj@gmail.com</a></td>
</tr>
<tr>
<td>Special Assignment</td>
<td>Hafiz Banire</td>
<td><a href="mailto:hbanire@gmail.com">hbanire@gmail.com</a></td>
</tr>
<tr>
<td></td>
<td>Gregory Bischoff</td>
<td><a href="mailto:gbischof@amgen.com">gbischof@amgen.com</a></td>
</tr>
<tr>
<td></td>
<td>Dave Adler</td>
<td><a href="mailto:brownsburgboiler@gmail.com">brownsburgboiler@gmail.com</a></td>
</tr>
<tr>
<td></td>
<td>Griselle Pratts</td>
<td><a href="mailto:grisellepr@gmail.com">grisellepr@gmail.com</a></td>
</tr>
<tr>
<td></td>
<td>Carlos McGowan</td>
<td><a href="mailto:los602@hotmail.com">los602@hotmail.com</a></td>
</tr>
<tr>
<td></td>
<td>Carlos Pereira</td>
<td><a href="mailto:carlospereirag@gmail.com">carlospereirag@gmail.com</a></td>
</tr>
<tr>
<td></td>
<td>Maxyme Lachance</td>
<td><a href="mailto:Maxym.Lachance@bba.ca">Maxym.Lachance@bba.ca</a></td>
</tr>
</tbody>
</table>