# *BUILDING A MANAGEMENT DASHBOARD*

## PRACTICE OBJECTIVE

This practice is designed to build a management dashboard of key indicators to monitor the processes that drive the mission. The dashboard needs analysis of the key indicators indicating expected results and potential actions if the dashboard readings vary from expected results. The data building the dashboard should be valid (see "Ensuring the Quantitative Data Needed to Manage by Fact is Valid" MGT. 4.1).

## PRACTICE WORKBENCH

Figure 1 shows the workbench for this practice. It uses as input the quantitative data available for decision-making purposes and the objectives/ constraints of the process(es) being monitored. The practice prepares the dashboard and associates most likely actions to take if the dashboard indicators vary from expected readings.

## INPUT PRODUCTS

The inputs relate to the process(es) being monitored which are used to accomplish desired work. The inputs are the available valid data produced by the process, plus the process objectives which are related to the I/S organization's vision and mission. Also, the process business risks identified in MGT. 4.1 are needed.

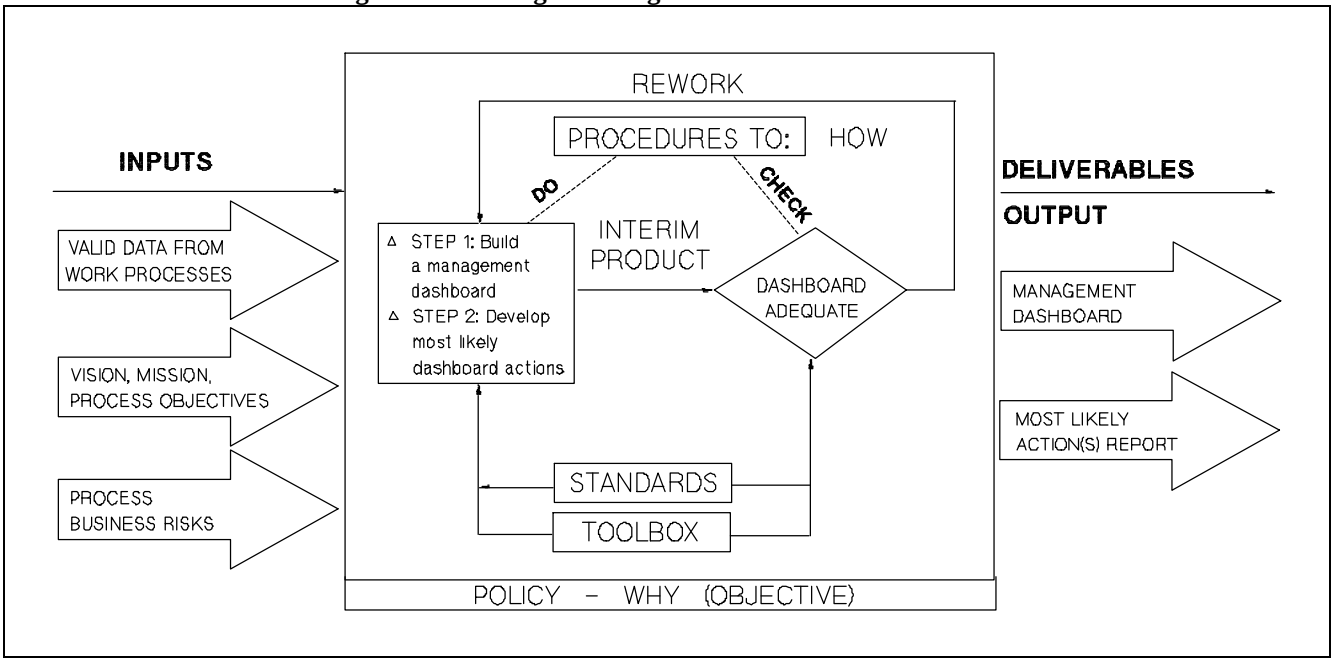
## IMPLEMENTATION PROCEDURES

The implementation of this practice requires two steps which are individually discussed in the following subsections.

### Step 1: Build A Management Dashboard (at the tactical level)

The objective of this step is to create a management dashboard that will report the quantitative and qualitative data in a manner that facilitates decision-making. The reporting process must be able to identify common from special causes of problems. Common-cause problems are those that can be corrected within the process; while special-cause problems normally require correction in another process.

**Figure 1. Building A Management Dashboard Workbench**



This step involves identifying the data needed for decision-making and then collecting and presenting that data in a format which facilitates decision-making. Decision-making can also be described as risk management. Thus, once the risks have been identified, decision-making will focus on minimizing the impact of those risks throughout the development process.

Let's examine one risk and the decision-making process associated with it. Let's look at the example of building an application system and the risk is that it won't be completed on time. The reporting system must then produce reliable data which properly describes the status of the software in relationship to its target implementation date. As reports indicate whether or not the project is on schedule, decisions will be made that will focus on completing the project by the scheduled implementation date.

The process requires a dashboard be established. The dashboard normally represents the risks associated with the project, expressed in positive terms such as whether or not the project is on schedule. The reports provide the underlying data which enables good decisions to be made to ensure that the objective (i.e., tactical dashboard indicator) is successfully achieved.

The desired results (i.e., results dashboard indicators) are incorporated into the decision-making process as process focus. Thus, while tactical process decisions are made on meeting or not meeting the schedule, those decisions are constrained by the results indicators which require that decisions be made in a process that focuses on the desired outcomes from the process, for example, an easy-to-use application system.

Task 1: Identify data needed

Identify data needed to build a process monitoring/reporting system. The process development team should determine what data is available from product controls, process controls, do procedures, and the operating environment.

The individual responsible for complying to standards, as well as those groups that have oversight responsibility, should be given information that indicates whether or not standards are complied with.

Feedback data can be manual or by-products of automated systems.

Examples of manual feedback information:

- Status reports (e.g., job status)
- In and out logs
- Correspondence
- Memorandum
- Time reporting
- Incident reports
- Audit reports
- Return receipts
- Signing for receipt
- Confirmations

Examples of automated feedback information:

- Job accounting systems
- DBMS logs
- Communication logs
- Operator logs
- Security logs
- Error files
- History files

Task 2: Ensure mechanism is in place

Ensure that a mechanism is in place to detect and report noncompliance to product and process standards to the process manager.

QAI's experience is that most information systems groups do not know their level of compliance to standards. Thus, determining noncompliance may be a complex task. Five methods for detecting noncompliance follow:

- Feedback data - Predefined methods which will report both compliance and noncompliance.

- End user complaint system - End users complain of poor performance which can then be evaluated to determine if noncompliance to standards was the cause.

- Quality audits - Quality people or auditors can periodically undertake audits to evaluate whether standards are or are not being complied with.

- Measurement of processes plus investigation of variances - Mechanisms can be used, such as control charts, to determine whether or not processes are in or out of control. Those that are out of control may be due to noncompliance with standards of the out-of-control projects.

Obviously, investigation must be taken before that conclusion can be drawn.

° Ask - Quality people can ask the operational developers if they are or are not complying with standards. It's surprising how many people will give honest responses to these questions.

Task 3: Build reports

Build a series of reports that provide the information needed to determine when action is required. The process reporting system will be based upon the defects identified to be monitored in process management. In many instances, there will be one report per major defect. While this step and task is separate from the task that set up the monitoring activities, the reports should be specified during building the do and check procedures of the process. This task involves the actual building of the mechanism to produce the reports, the format of the reports, and the frequency of the reports.

Task 4: Review risks

Review risks which need to be continually monitored. From Worksheet #1, select the risks which are indicated addressable by milestones and become familiar with the potential impact of each risk and the associated defects on meeting the objectives of the work process. Post the potential defects to Worksheet #2.

Task 5: Design key indicator

Design a key indicator (dashboard indicator) for each major defect. Each major defect that needs to be constantly monitored should be turned into a key indicator. For example, in testing the test process, management may need to constantly monitor the number of functions successfully tested, or the number of major defects which have not been corrected within a five-day period. Appendix 1 provides an example of a dashboard.

Task 6: Set reading for each dashboard

Set the expected average reading/standard reading for each dashboard. A determination needs to be made as to what reading is expected at any point in time. Note that this may be a constant reading, or might be one that changes over time. For example, for number of testing defects not corrected within a five-day period the standard may be true; while the number of functions successfully tested may change every week or every day over the entire test period. For example, when 25 percent of the test resources have been expended the expected reading may be 15 percent of the requirements; while at the point where 75 percent of the test resources have been used, the standard for functions successfully tested may be 65 percent.

**Step 2: Develop Most Likely Dashboard Actions**

Two types of most likely actions need to be developed for potential use when a dashboard indicator shows an undesirable reading. The two action analyses are for common cause and special cause.

Task 1: Identify potential actions to address common causes within a process

The objective of this task is to predetermine the type of decisions/adjustments which could be made to optimize the process based upon the results desired from the process. Some have referred to this as "managing the process" and others have referred to it as "navigating through the process." The end objective is to meet the desired results in the most effective and efficient manner possible.

This task addresses the problems that can be satisfactorily handled by the existing process. The method selected for addressing this type of problem will not impact the results to be achieved from the process but, rather, impact the selection or application of methods used in performing the process. For example, the action might involve changing work hours to have an individual work from 4:00 p.m. to 12:00 p.m. rather than from 8:00 a.m. to 4:00 p.m. This change might be made because of availability of computer resources in the late afternoon and evening, while those same resources might not be available in the morning or early afternoon.

This task involves the following:

A) Establish triggers that will indicate when action may be required. For each identified defect that requires potential action, a decision must be made in advance as to what level of performance would trigger that action. It may be the defect itself, or it may be the magnitude of the defect that triggers action. For example, in our testing process a defect uncovered in testing requires action. On the other hand, an additional action may be required if the defect is not corrected in a five-day period. However, it is the elapsing of the five days that is a prerequisite to the second action. On the other hand,

it may be three or more major defects not acted upon for five days that would require an additional action.

B) Determine what action needs to be taken if the dashboard indicator is higher or lower than the expected reading/standard. Guidance should be provided the process manager on what action to take when:

- The indicator is higher than the standard
- The indicator is lower than the standard

The actions might change based on how much higher or lower the reading is. For example, defects uncovered during testing not addressed in a five-day period could result in one action if the indicator was slightly over, for example, check with the programmers on why the defects are not being corrected on a more timely basis; while if the indicator is much over standard, the action may be to stop testing until the defects are corrected.

C) Record the results of this task on Worksheet #2. Define guidance for taking action for defects requiring action (e.g., a defect trigger has been activated). The primary objective of this task is to define what actions should be taken should the selected defects occur.

*Review the process, policy, deliverable standards, and quality control method related to the defect*¾the process development team must understand the literal and intent standards for the deliverable, as well as the method used to control the risks associated with the process. Understanding the control method will help identify who is responsible for decision-making, while the intent and literal standards will help identify the action.

*Identify who has decision-making responsibility*-determine what area/manager is responsible for the action. This is generally the same person who is responsible for the accuracy and completeness of the deliverable.

*Identify the most common action to take if defect occurs*-the process development team must determine what action should be taken if the defect occurs. For example, if the user documentation is not easy to understand, then a common action to take might be to rewrite the documentation using shorter sentences and simpler words. The action that was posted to Worksheet #2 should be expanded and documented on Worksheet #3, showing the process being controlled, the deliverable, the quality control method, the defect name, the defect description, decision-making responsibility, and the action to take if the defect occurs. This information will come from all three actions in this task.

D) Devise strategies for taking both short-term and longer-term decision-making action. Some examples of short-term strategies and long-term strategies for decision-making follow.

*Short-term strategies* - Listed below are some short-term strategies that quality people can use in addressing compliance to standards:

- Short-term exemption - This strategy is most effective against projects which claim compliance to standards will force them to meet critical deadlines. Short-term exemptions, such as 60 or 90 days, can be granted to the project. The project is required then to come in compliance with standards within that exemption period.

- Third-party decision-making - This strategy is most effective when you are trying to introduce the concepts of standards; and yet want high compliance. Checkpoints are established which stop projects from proceeding unless they are in compliance with standards. For example, projects will not be accepted in operation unless they meet the minimum operation standards. Computer operations, being a third-party group, then enforces the standards.

- Twenty-four-hour alert status - This is another strategy effective against projects which do not willingly comply with standards. The concept says that without compliance to standards, only the developers can maintain the system and/or correct problems to the system. Thus, the project personnel must be on duty 24 hours a day in the event of problems. Some organizations using this concept even require those individuals to be in the computer room whenever the system is run.

- Freeze job task - This strategy is most effective with ambitious individuals. The concept says that they will be frozen in their current job until the project comes in compliance with standards. The incentive is that to be promoted or transferred, you must get your project in compliance with standards.

- Add automatic delete for nonstandard input - If the input is entered into computer media, a preprocessor can be created which will reject nonstandard input. For example, if a particular verb is declared a nonstandard verb, then a preprocessor to compilers can delete that verb in the event anyone chooses to use it.

- Posted lists of noncompliers - Individuals which do not comply to standards have their name listed in a prominent spot within the information systems function. The purpose of this is to clearly identify violators from normal practice. The technique is most effective when there is peer pressure for compliance to standards.

- Job evaluation criteria - Compliance to standards can be considered one of the criteria for evaluating performance. If people fail to comply with standards, that information can be put in their job folder, or fed to their supervisor. This concept is most effective when supervisors support compliance to standards.

- Reward compliance - This is one of the most effective strategies for compliance, but it requires the support and involvement of senior managers. Individuals who are in high compliance to standards are singled out for public recognition in the department. For example, the information systems director may identify these individuals at staff meetings, and then reward them in some manner. Rewards can include time off, special privileges such as attendance at a conference, a plaque or trophy to put on their desk, or perhaps a special parking privilege or other desirable job attribute.

- Internal audit support - The auditors can be strong proponents for standards. Notifying the auditors that there is noncompliance to standards, and requesting them to look at this issue in their next audit, may lead to some strong audit recommendations for compliance.

*Long-term Strategies* - The following long-term strategies should be put into place concurrently with short-term strategies:

- Address prerequisites to control - The best long-term strategy is to put into place those events which will change the environment to favor compliance.

- Training of management - Many managers don't understand the benefits of standards, and how compliance to standards can improve quality and productivity. Continual training sessions help do this.

- Outside expert - You can bring in some renowned expert in the field of quality or information systems to present the arguments for compliance to standards. Some managers will listen to outside experts, while they won't listen to their own staff.

- Standards policy - Management could issue a policy or directive which requires compliance to standards. Many staff members have never heard their senior managers say they want compliance, and thus don't believe it's important to their personal career or the function to comply to standards. However, establishing a policy is only the first step; the second step is management supporting the policy by putting in the programs and procedures which will make compliance a long-term strategy of the organization.

- Measurement programs - Through measurement, the difference between compliance versus noncompliance becomes obvious. When many organizations see a significant difference between the two approaches, they can be won over to the compliance approach.

- End user/senior management support for standards - In information systems organizations where there is not a strong internal support for standards, that external support may have to come from outside groups. Key end users or senior managers supporting standards may make a significant difference.

Task 2: Identify potential special actions

The objective of this task is to take action on special-cause problems which must be addressed by a process other than the one being performed. This task must identify which process can address the problem, and then use the common-cause decision-making process within that process.

This task is one of tracing the problem to the process in which the cause of the problem exists. Once the problem was identified as a special-cause type of problem, it cannot be addressed in the process in which the problem was uncovered. For example, if the process was one of building a software system and the problem was that there were inadequate resources to build the system, the build process cannot resolve that problem. The problem must be resolved either in the requirements definition process by changing the requirements, or the budgetary process to change the budget.

This task involves the following:

A) Trace the special-cause problem to the process(es) in which the cause of the problem can be addressed. The guideline relationship map created in the process mapping step is the preferred tool for showing processes that drive other processes.

B) Identify the decision-maker in the root-cause process and request a decision be made from that process to reduce/eliminate the identified problem. The common-cause decision-making process as previously described can be used by the decision-maker in the affected process to make a decision to address the special-cause problem. Note that the special-cause problem is a special-cause problem in the process in which it is identified; but it is a common-cause problem in the process where it can be addressed.

**CHECK PROCEDURES**

Validate that the decision made is a reasonable decision. This check procedure can be performed a variety of ways, including:

- Ask involved parties about the reasonableness of the decision.
- Attempt to relate the facts to the decision to determine that the facts support the decision.

**DELIVERABLES**

The two deliverables that will be produced from this practice are:

1) Dashboard indicating what risks must be monitored and the quantitative metric that will be used to monitor those risks. An example of a typical dashboard is offered as Appendix 1 at the end of this section.
2) The data and reports which underlie those risks to be used for taking action to address/minimize the risks included in the dashboard.

**USAGE TIPS**

The problem which is identified for action (the trigger that causes the action) must be categorized as a common cause, something that can be addressed within the process, or a special cause, something that needs to be addressed outside the process.

## WORKSHEET #1 - WORKSHEET FOR DEFINING PROCESS DEFECTS

**PROCESS BEING CONTROLLED:** _____

| # | PROCESS RISK | CONTROL POINT | POTENTIAL DEFECTS | DEFECT TYPE | | PROCESS RESOLUTION | | PRIORITY/ IMPACT OF DEFECTS | | DEFECT SELECTED | | COMMENTS ON SELECTION | HOW TO ADDRESS DEFECT |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | Literal | Intent | Common Cause | Special Cause | Major | Minor | Yes | No | | |
| | | | | | | | | | | | | | |

**WORKSHEET #2 - PROCESS DASHBOARD**

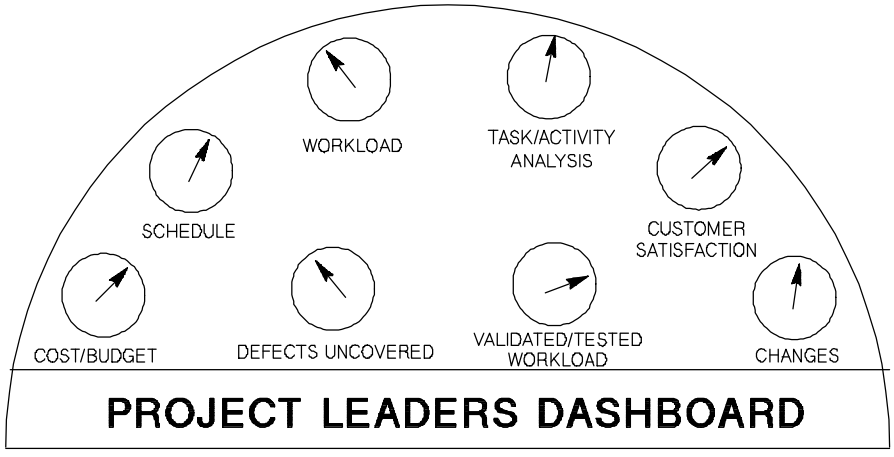
| POTENTIAL DEFECTS REQUIRING CONSTANT MONITORING | DASHBOARD INDICATOR | EXPECTED DASHBOARD READING STANDARD) | DASHBOARD ACTION FOR | |
|---|---|---|---|---|
| | | | HIGH INDICATOR | LOW INDICATOR |
| | | | | |

**WORKSHEET #3 - DEFECT ACTION WORKSHEET**

| |
|---|
| **PROCESS BEING CONTROLLED:** |
| **DELIVERABLE:** |
| **QUALITY CONTROL METHOD DESIGNED TO IDENTIFY DEFECT:** |
| **DEFECT NAME:** |
| **DEFECT DESCRIPTION:** |
| **ENFORCEMENT RESPONSIBILITY:** |
| **POTENTIAL ACTION TO TAKE IF DEFECT OCCURS:** |

# *APPENDIX 1: EXAMPLE OF A TYPICAL SOFTWARE DEVELOPMENT PROJECT LEADERS DASHBOARD*

This dashboard represents a typical dashboard that might be developed for managing the software development process. The key indicators/dials to be included in the project leaders dashboard are described below.




**Dial 1 - Cost/Budget**

This measure will indicate the relationship of meeting budget. It will indicate at any point in time whether the project is over or under budget. The measurement dial could either be in percent or time periods such as person-months, whichever appears more meaningful.

**Dial 2 - Schedule**

This dial will indicate whether the project is behind or ahead of schedule. It can be presented in percentage, or number of days/weeks behind or ahead of schedule.

**Dial 3 - Workload**

This unit of measure will indicate the amount of work required to be done during the project. For systems development, it will be expressed in either requirements or function points.

**Dial 4 - Task/Activity Analysis (Checkpoints/Milestones)**

This dial will indicate what checkpoints/tasks/activities/milestones have been met. If the milestones are used in a sequential manner, the dial can indicate which is the last milestone which has been achieved.

**Dial 5 - End User Satisfaction**

This measure will indicate the degree of satisfaction the end user has at any point in the developmental process. To do this will require periodically surveying the end user to determine the degree of satisfaction.

**Dial 6 - Defects Uncovered**

This measure will indicate the number of defects that have been uncovered to date in the project. To activate this dial would normally require the initiation of software inspections.

**Dial 7 - Validated/Tested Workload**

This measure would indicate the percent of requirements/function points that have been validated through testing. It would indicate the percent of workload that has been validated to work in an operational status.

**Dial 8 - Changes (Weighted Changes)**

This dial will indicate the absolute number of changes that have occurred since the project began. However, if the dial is to be meaningful in measuring growth and workload, a scheme of weighting changes needs to be worked out, so that the number produced would be a weighted number rather than number of changes.