

**Information Structuring “Methodology”: Tools and
Techniques for Effective Program Management**
Prepared by Dr. Sami Alzahrani
Lecture 10

Lecture 10

PMG 665

Program and Portfolio Management

Introduction

The program/project management and industry specific methodologies describe in detail the process steps required to realize an organization's strategic goals. The realization of strategic goals is through portfolios, and this is one of the truest measures of an organization's intent, direction, and progress. If an organization decides on a new strategic direction or just a strategic alignment, it will need to balance its enterprise portfolio(s), which inevitably will result in issuing one or more program mandates.

Introduction

If in the absence of an enterprise portfolio, the executive committee will issue a program mandate as a way to meet the organizational strategic goals. Once the mandate is received, the program/project management methodologies will then further detail the steps to structure, define, plan, execute, monitor, and govern the program/project endeavors.

Information structuring methodology in context with the enterprise management framework



A program's success or failure is based on whether its vision and goals have been met, which in itself is a direct result of whether or not the baseline triple constraints of scope, time, and cost have been managed effectively. An analysis was carried out to determine the top common causes of program and project failures.

Information structuring methodology in context with the enterprise management framework



Many of the reasons given for program failure were the symptoms of a root cause. The root causes were then mapped into an Enterprise Management Framework, shown in Figure 18.1. Surprisingly, the results showed only one root cause per level of the Enterprise Management Framework outside of the Program and Project Management segment.

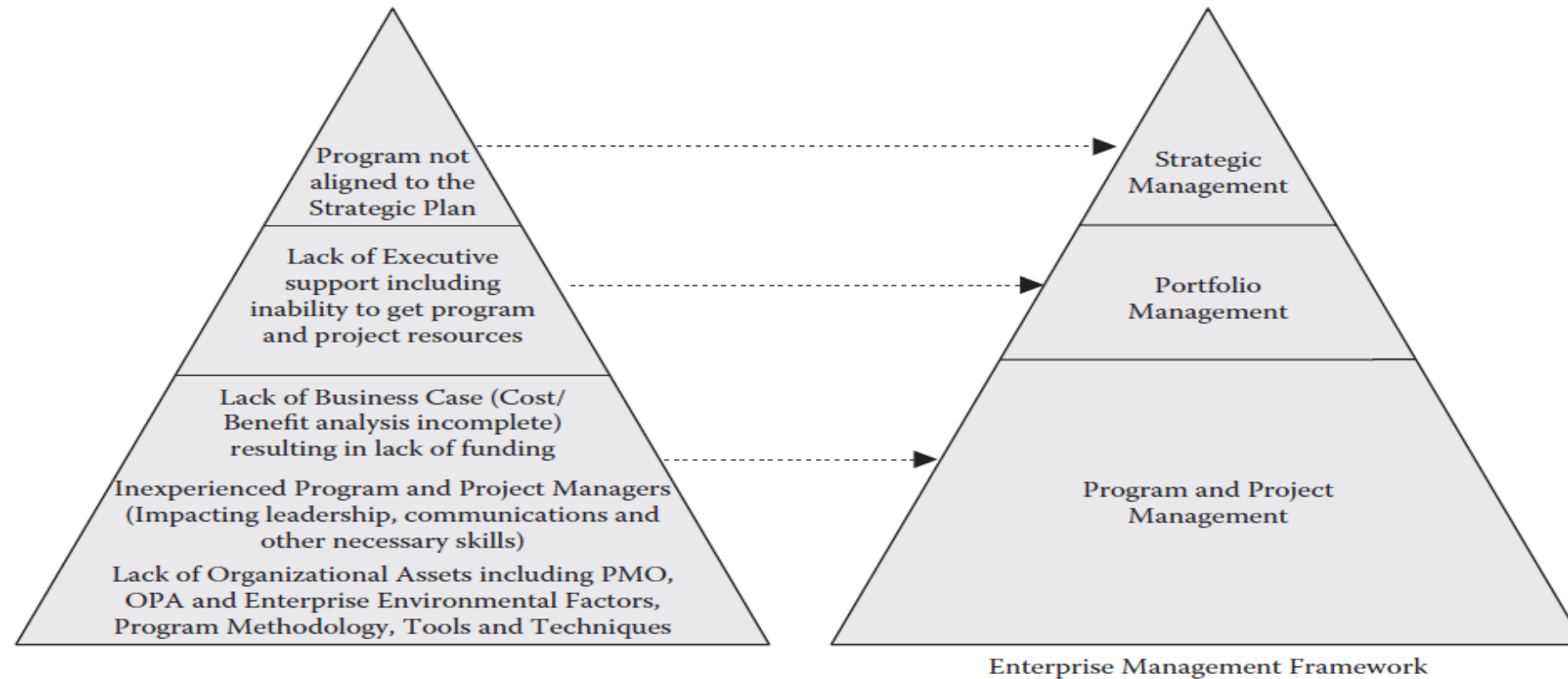


FIGURE 18.1

Program external factors

A program is impacted by external factors (influences) during its life cycle, and unless managed and optimized, the program will lose opportunities and perhaps suffer risk impacts that may not have risk mitigation or risk response plans.

Program external factors

If the program experiences constraints that could delay schedules and also experiences “integration challenges”, or if either are ignored, they will severely disrupt the program and distract program management attention that should be focused on program definition and planning. The complexity and relationship of program external factors must be understood before any optimization can take place.

Program external factors



a program can be impacted by five types of external factors:

1. Risks (Negative)
2. Risks (Positive) or Opportunities
3. Organizational Process Assets (OPAs)
4. Constraints
5. Integration Challenges

Risks (Negative)

Risks can be categorized in the enterprise as enterprise “external risks” and enterprise “internal risks.” Enterprise external risks are mainly the “unknown-unknowns,” consisting of political, economic, acts of God and man-made crises not known at the time of key decision making that then may impact the program. The risks associated with outsourcing are also included. Risks, such as changes to contract conditions, service set up, and service delivery, all will impact the program in some way.

Risks (Negative)

Global and regional outsourcing contracts are typically outside the influence of the program manager, and as a consequence, service definition and delivery issues coupled with service disputes will impact the program's ability to perform and deliver.

Risks (Positive) Opportunities

Opportunities can arise in obvious and in disguised forms. A disguised opportunity is one that could be perceived or interpreted as a threat. It takes skill to turn a perceived threat into an opportunity and then depending on the enterprise risk tolerance level to exploit this opportunity to achieve the maximum gain for the enterprise.

Organizational Process Assets

Enterprise assets in the form of OPAs, i.e., the plans, procedures, policies, and guidelines coupled with the EEFs (internal), should all serve as assets to the program if they are well thought out and designed. However, in many organizations an attempt is made to globalize standards, processes, and procedures to increase efficiency and effectiveness.

Constraints

From the outset, every program must minimize the impact of internal as well as external constraints to allow as much flexibility while increasing the number of potential options. Therefore, when decisions need to be made, the timing of the decision and the option selected will provide the greatest benefit at the optimal time. Internal constraints include fixed deadlines, resource limitations, closeout periods, funding limitations, and limited infrastructure. External constraints include government, industry, and local authority regulations and vendor processes.

Integration Challenges

With the advent of outsourcing, new challenges arise for the program manager. Forward or backward integration for the development/support processes will be required and to a limited extent a similar implication of the “make versus buy” decisions. Outsourcing may have been introduced enterprise wide, and set roles outsourced e.g., programming, testing, integration, project management, and hardware infrastructure. This all implies that processes, tools and techniques will need to be integrated into the outsourcing partner.

Program “internal” constantly changing factors

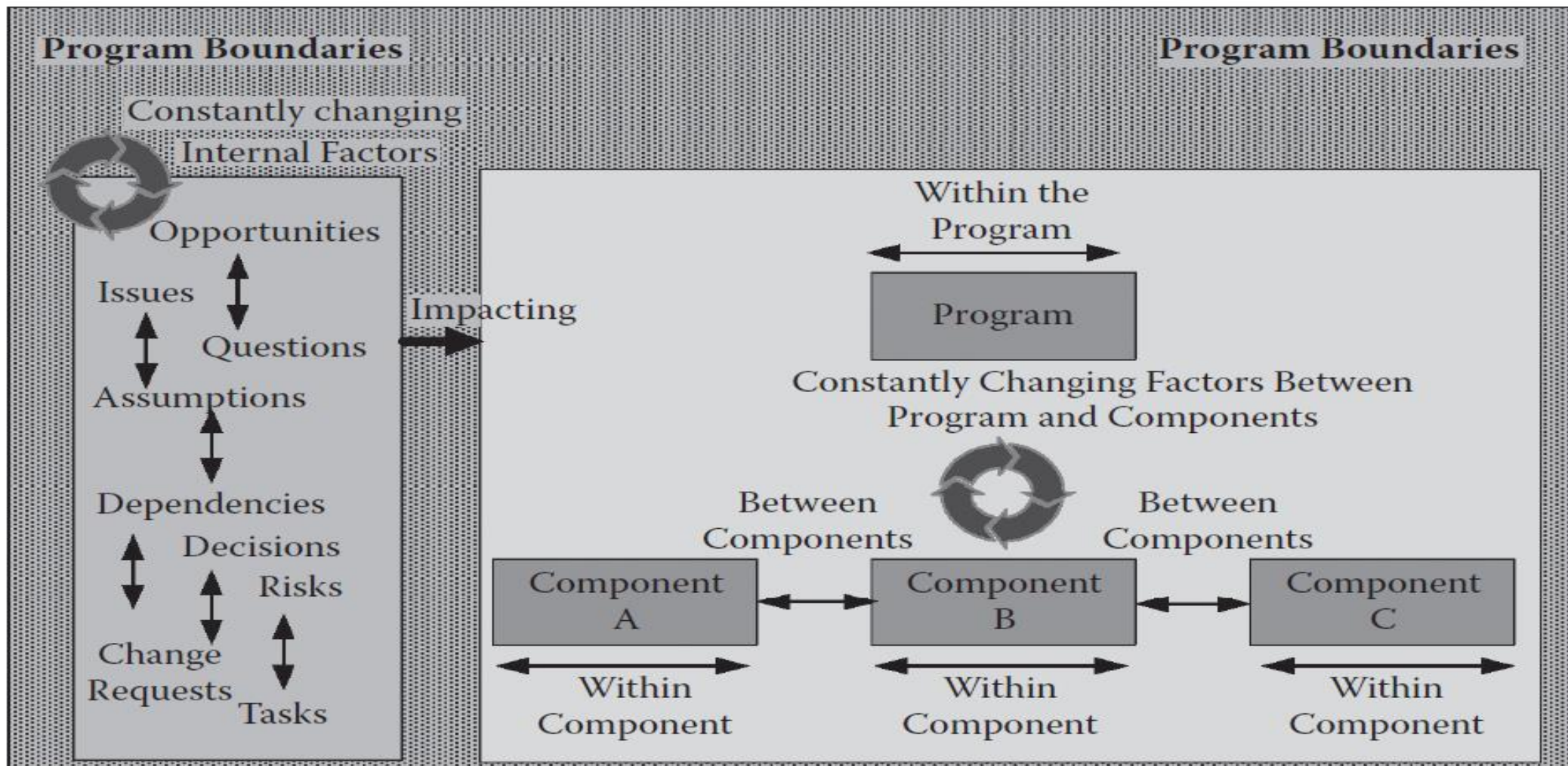
Program “internal” constantly changing factors is a phrase used to express the dynamic nature of a program. These factors need to be identified, categorized and assigned to the area(s) of impact and the person(s) responsible to address them in the most effective way. As a program is established, if a plan to structure this information and manage it based on a structured methodical approach is not developed, the program manager and his or her team(s) will be continually playing a “catch-up” game, which is not the best way to start a program.

Program “internal” constantly changing factors



Figure below shows the types of internal constantly changing factors that impact the program and its components. However, what the diagram cannot show is the complexity of program regarding the changing factors nor the number/complexity of each type of changing factor. Imagine the program covering several countries all with their own local related projects (components) and then all linking them back to the core program.

Program “internal” constantly changing factors



Program “internal” and “external” constantly changing factors



The program manager must manage program “internal” and “external” constantly changing factors to ensure control is maintained. The complexity of the program environment and its constant interactions with changing factors (influences) is hard to manage, and this may be a reason why between 30 to 70% of programs fail (Gartner, 2011).

Program “internal” and “external” constantly changing factors



For the program manager to successfully manage complexity, the challenge is to find a way to plan and set up the program environment so it is designed to be able to handle shocks and negative scenarios as a result of a sequence of related or non-related constantly changing factors.

Program management information system tools



The PMIS is a generic term to mean a series of interrelated tools supporting to collect and manage information needed to monitor and control the program. There is one other system tool that is not part of the PMIS tool set, and it is described as the Procurement Management Information System (PMI[®], 2008).

The PMIS and the Subset of Tools that Benefit the Most from the Information Structuring Methodology

The information structuring methodology can be applied across the whole program including the typical areas covered by PMIS such as configuration management, financial management systems, and earned value management, but these are not impacted the most in terms of information structuring. The PMIS tools/functions, which benefit the most from the information structuring methodology especially when the methodology is consistently applied across them are shown and described below in Figure 18.8.



PMIS (Program Management Information System)

PMIS tool (subset) which require the information structuring methodology

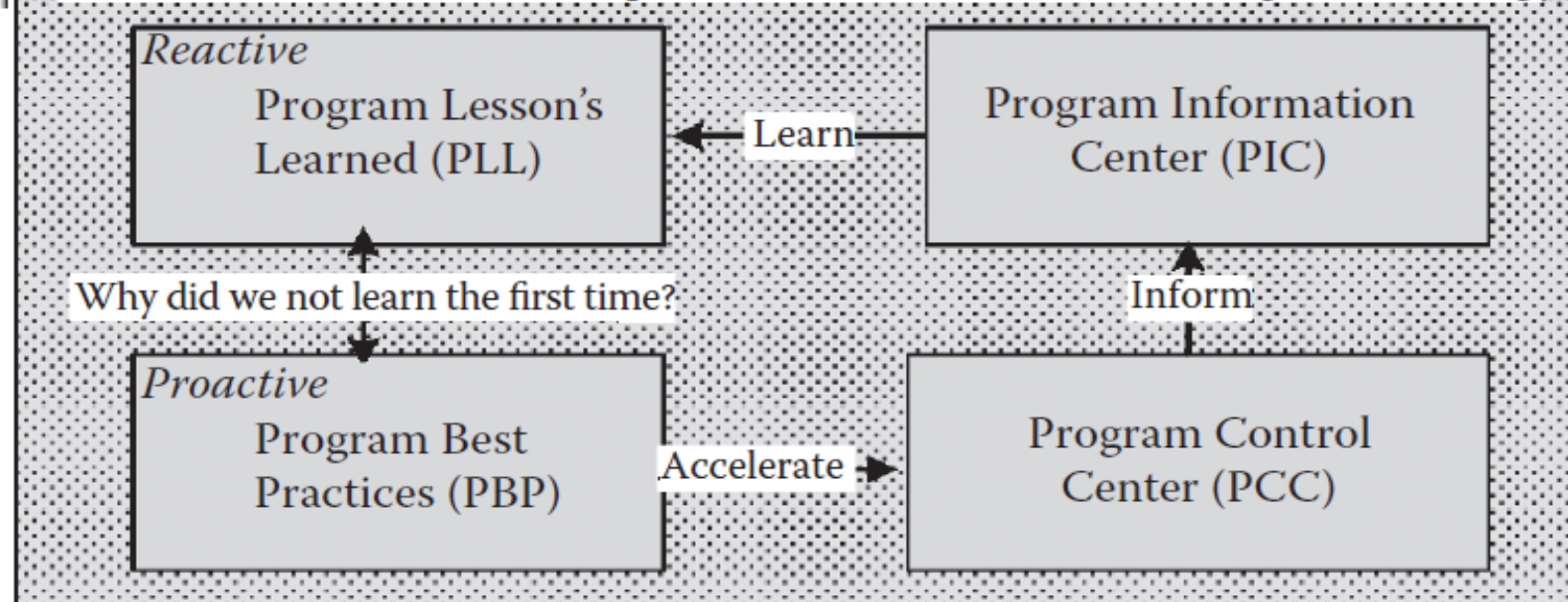


FIGURE 18.8

PMIS tools that benefit the most from the information structuring methodology.

Program Control Center

The Program Control Center (PCC) is the center or place to control the program. To be in control of the program requires not only the timely information of performance reports but also a complete understanding the potential impact of the constantly changing factors and how they can or are impacting on the program.

Program Control Center

To gain the full benefit of the PCC, the component or project leaders should use this tool within the embedded program information structure. Also every team member must be empowered to be able flag issues at any time, ask questions, raise new risks, update information on existing risks, and assign issues to person(s) while notifying others automatically to keep the relevant stakeholders informed.

Program Information Center

The Program Information Center (PIC) is the second tool within the PMIS and is used as an information store so information can be pushed or pulled to the extended stakeholder community. The level and amount of detail required by the stakeholders is determined when the team prepares the stakeholder register. Information contained in the PIC includes decisions made and resultant impacts on the program life cycle and/or project life cycle.

Program Information Center

It includes questions and answers, assumptions made, and the impacts on the program life cycle and/or project life cycle. All stakeholders have access to the PIC, and it will provide a dashboard of the program status as well as an information history categorized to the main program benefits and project deliverables.

Relationship between the PCC and PIC

The PCC and PIC conceptually are shown as two tools, but they can be the same tool depending on the security, functionality, and roles built into the tool. Figure 18.9 shows them as separate tools. The PCC is where the work is assigned, monitored, and completed. The PIC ensures that information transparency with the extended stakeholder community is available such as what has been done, based on what assumptions,

Relationship between the PCC and PIC



the decisions made including change requests, and published documents. It is important to determine which stakeholders have access to the PCC because this is where potentially sensitive information is stored, whereas the PIC is for non-sensitive information for the entire stakeholder community.

Relationship between the PCC and PIC

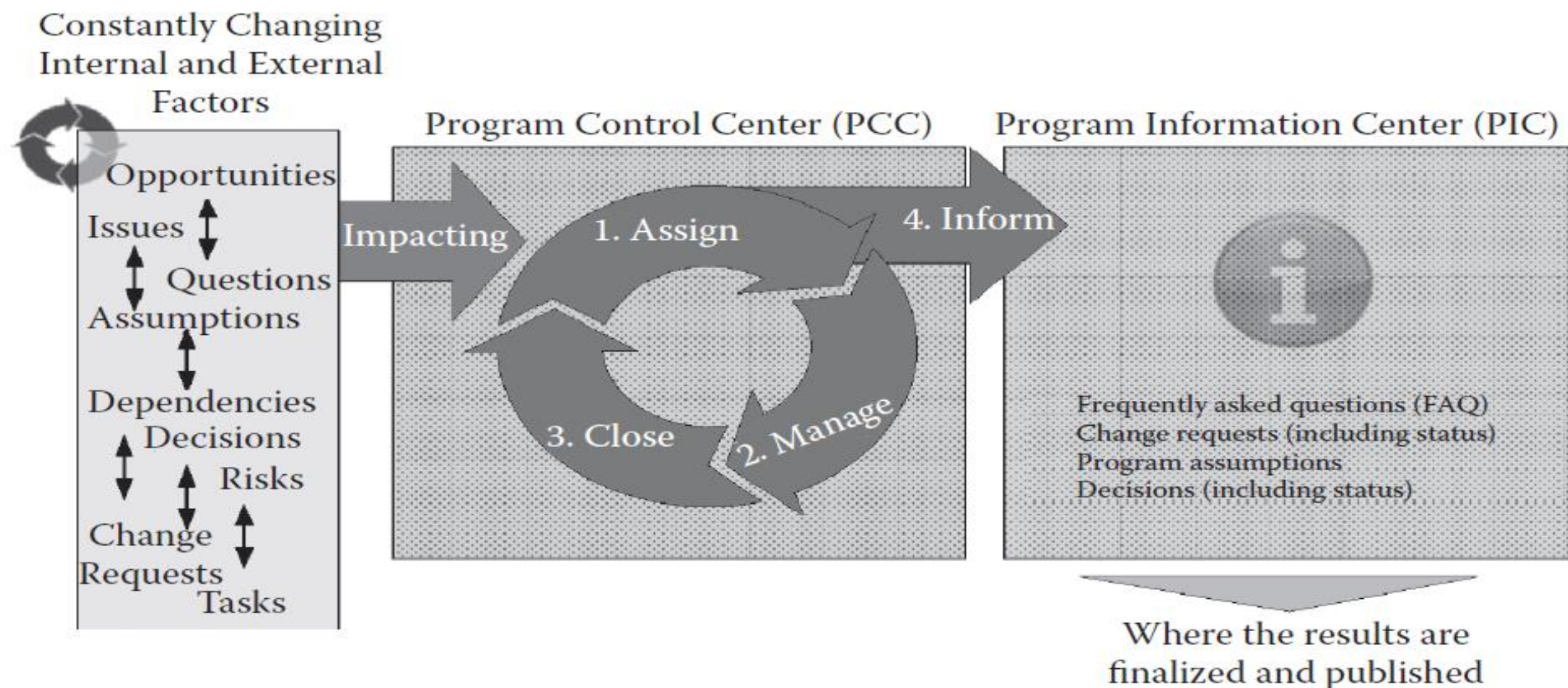


FIGURE 18.9

Relationship between the PCC and PIC.

Program best practices

Program Best Practices (PBP) contain all the templates and reference documents relating to best practices adopted within the program. It is a subset of the OPAs, and it is typically managed by the PMO.

Program best practices

Program Lessons Learned

Program Lessons Learned (PLL) is structured in the same way as the PCC and PIC, therefore allowing comments and reports to be captured around the activity areas (components, deliverables, and work packages). It is a non-restrictive database accessible by all. As a program is progressing, on completion of formal deliverable reviews, a lessons-learned document is written and categorized against the area/item/deliverable. On completion of a component, key findings are reported, and the lessons learned are archived. The PLL also is managed by the PMO.

Selecting the Appropriate PMIS Support Tools

Often program managers try to manage programs with single-user tools. A classic example is to use a spreadsheet as a program task list or a time reporting system. To help program and project managers decide the best support tools to use, a simple set of critical success criteria shown below will help with an initial selection of program support tools.

Selecting the Appropriate PMIS Support Tools

One of the success criteria is often challenged—a free tool, i.e., not charged to the program or project. Often, program managers are not prepared to pay a contribution for a multiuser tool and therefore decide to use a single use tool that is widely available within the enterprise. A program manager should know when to invest in the program and not be “penny wise and pound foolish.”

Selecting the Appropriate PMIS Support Tools

Team members and consultants cost thousands of dollars a day so doing tasks as a result of confusion or misunderstanding arising from a lack of an integrated support tool is worth that small initial tool contribution. When a support tool is being proposed, it should be first checked against the list below to see if it meets all the success criteria.

Selecting the Appropriate PMIS Support Tools

The tool should not be implemented if it misses even one success criterion because the outcome will be an unused tool, and a program will have an unnecessary risk. Guidelines for program performance tools in terms of success criteria follow.

Selecting the Appropriate PMIS Support Tools

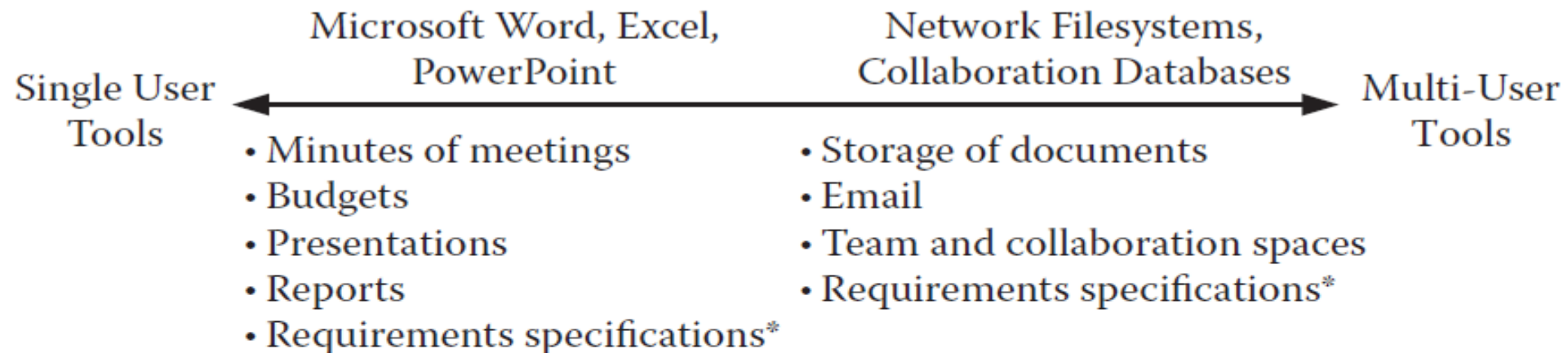
- Multi-user—several people can update task lists simultaneously
- Easy to use—people do not change from using existing tools if the new ones are harder to use
- Globally accessible—team members regardless of where they work are able to use the same tools
- Performance—the tool has optimal performance wherever the teams are located

Selecting the Appropriate PMIS Support Tools

- One integrated tool rather than several—one view of the program and one view from all the knowledge areas
- Free to use—no additional cost to the program and/or its component project's stakeholder groups

Enterprise tools that are available and free for the program may not meet the tool success criteria for some of the intended uses. Figure 18.10 below

Selecting the Appropriate PMIS Support Tools



*Depending on the complexity of the program and the component requirements their interdependencies/cross references

FIGURE 18.10

Program/project single and multi-user tools.



shows there are uses for single user tools, but the limitations should be understood, and the temptation to extend their uses to areas that are better served by multi-user tools should be avoided even if additional program costs will need to be incurred.



Reference

Program Management: A Life Cycle Approach
by Ginger Levin, PMP, PgMP

Chapter 18