ESI European Software Institute Center Eastern Europe

Partner of: Software Engineering Institute | Carnegie Mellon

SEMP SOFTWARE ENGINEERING MANAGEMENT PROGRAM

Q.A.
Осигуряване на качество на софтуера
(2020/2021, редовно/задочно)

based on:

Software Quality Management Models: Intro to Process Improvement (PI)

[SEMP Program course, in collaboration with Carnegie Mellon University]

Dr. George Sharkov FMI/PU & ESI Center Eastern Europe/Bulgaria www.esicenter.bg
gesha@esicenter.bg

Dr. Maya Stoeva FMI/PU
may_vast@yahoo.com

Информация, източници:

ESI Center Eastern Europe - Resources:
www.esicenter.bg >> general info and all in “Resources” (+ the model in pdf ver 1.3)

Links to CMMI models (from the source):
http://cmmiinstitute.com/cmmi-models
Download CMMI –DEV v 1.3 model (CMMI Institute, and SEI, Carnegie Mellon University)
(free, upon registration)

https://resources.sei.cmu.edu/asset_files/TechnicalReport/2010_005_001_15287.pdf

https://en.wikipedia.org/wiki/Capability_Maturity_Model_Integration

General

www.sei.cmu.edu
http://resources.sei.cmu.edu/library/
www.cmmiinstitute.com

FMI-PU dedicated site

[Maya Stoeva | Cources | Software Quality Management - CMMI (edesign-bg.com)]

Who are we?

ESI European Software Institute tecnalia

Since 1993

partner of:

Software Engineering Institute | Carnegie Mellon

ESI European Software Institute Center Eastern Europe

Since 2003

ESI Center Eastern Europe

PPP: SW Industry(BASSCOM), ESI & State ICT agency, supported by: USAID, UNDP

SEI Partner | Carnegie Mellon CERT RMM

it mark

CMMI

Helping companies and organizations compete by QUALITY and EXCELLENCE since 2003

Affordable "BIG" standards for "small" companies

ESICenters • ESI@net Partners

Corporate excellence perspectives

Corporate excellence is a balanced model

Kaplan and Norton structured it in four perspectives:

- Financial perspective
- Customers perspective
- Processes perspective
- Learning perspective

The Balanced Scorecard is a framework for translating a vision into a strategy by focusing on shareholder, customer, internal and learning requirements which collectively describe the strategy of an organisation and how that strategy can be achieved.

Kaplan & Norton
Harvard Business Review ,1992
"The Balanced Scorecard - Measures that Drive Performance"

Financial Perspective

Results-oriented perspective that covers goals and performance measures related to the financial performance of the company.

Typical indicators: Return on Investment (ROI), Shareholder Value, Increase of Revenue, Increase of Turnover, Cash Flow, etc.

Customer Perspective

Related to the market and customer segments and it directly supports the implementation of financial objective.

Typical indicators are: market segments, customer satisfaction, percentage of new customers, life cycle, quality, service, price - quality, delivery times, reputation, commitment to delivery times

Process Perspective

Defines and measures the processes, in which the company should invest and improve so that it can attain the goals in the customer and finance related perspectives.

Typical indicators: Processing time, % millstones met , process frequency, process costs, process quality, time to market, innovation cycle etc.

Learning and Growth Perspective

Structuring goals and performance measures related to the knowledge necessary for maintenance and further development of all perspectives.

Typical indicators: market innovation, intellectual competences, staff satisfaction, fluctuation, staff productivity, number of improvement proposals, quality of improvement proposals, training days, etc.

The sad truth

25% of all software projects are killed.

Companies are releasing products to their customers with 15% of the defects remaining in the product.

Many companies are spending 30-44% of their time and money on reworking software they have already written.

Companies meet their schedules only 50% of the time.

Sources: Capers Jones and Bill Curtis

Cost of Quality (CoQ)

Crosby describes Cost of Nonconformance as the extra cost incurred because a product or service wasn't done right the first time.

Cost Categories

Cost of Nonconformance
+ Cost of Conformance
= Cost of Quality

- Internal Failures + External Failures
- Prevention + Appraisal

CoQ Cost Categories (exercise)

Prevention	Appraisal	Internal Failure	External Failure
Costs associated with preventing defects	***Costs associated with "looking" for defects***	***Costs associated with defects found prior to implementation / release***	***Costs associated with defects found after the product is implemented / released***
Planning	Reviews	Rework	Warranties
Documentation	• System	• Requirements	Complaint adjustments
Training	• Requirements	• Design	Lost projects
Tools	• Design	• Code	Tech support
Policies and procedures	• Test Plan	• Documentation	Subsequent releases, patches, "Service Packs" (MS terminology)
Quality improvement projects	• Test Script	Defect re-testing	
Data gathering and analysis	Walkthroughs and code inspections	Process losses (testing downtime, changing deliverables, schedule slips, cost overruns, etc.)	
Fault and root cause analysis	• Testing (First-time)		
Quality reporting	Audits		
	CMM Assessments		
	• Class A,, B, C		

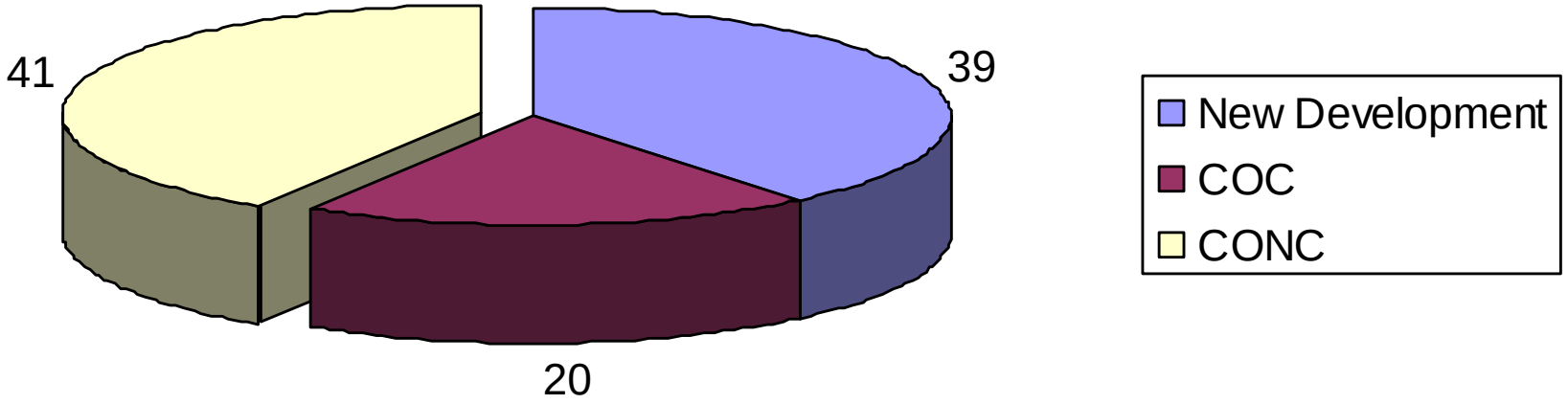
An Early CoSQ Experience

Raytheon

Where are software engineers spending their time?

OR

Where are we spending our software engineering budget?



Source: Raytheon Electronic Systems Experience in Software Process Improvement, CMU/SEI-95-TR-017, November 1995

Successful software process improvement programs can

reduce the number of defects delivered to customers by 95%

reduce software development schedules by 71%

increase productivity (measured in lines-of-code or function points per day) by 222%

realized an average ROI of 5:1

Sources: Capers Jones and Software Engineering Institute

Why Focus on Process?

Process provides a constructive, high-leverage focus...

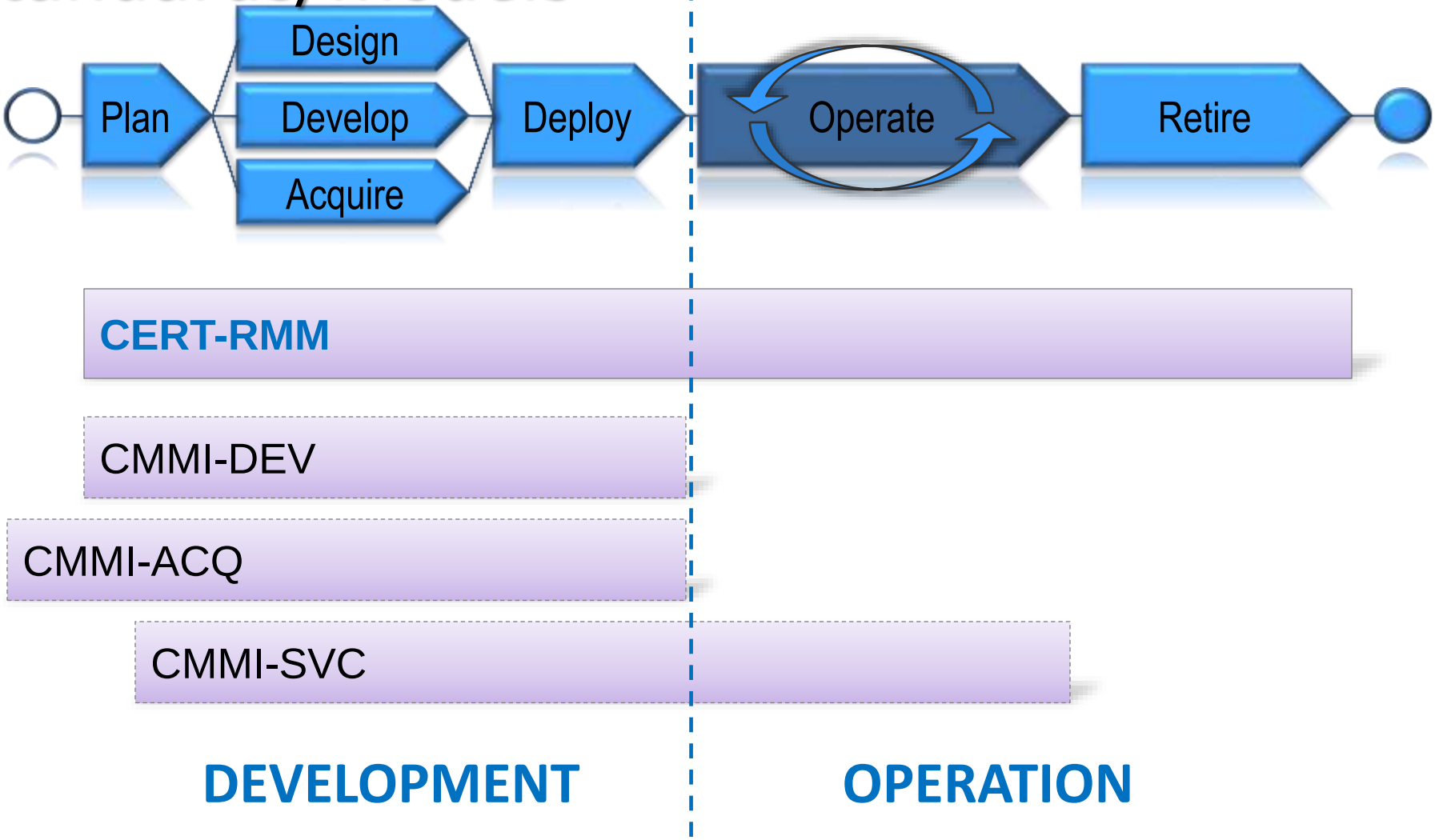
... as opposed to a focus on people

- Your work force, on the average, is as "good" as it is *trained* to be.
- Working harder is not the answer.
- Working smarter, through process, is the answer.

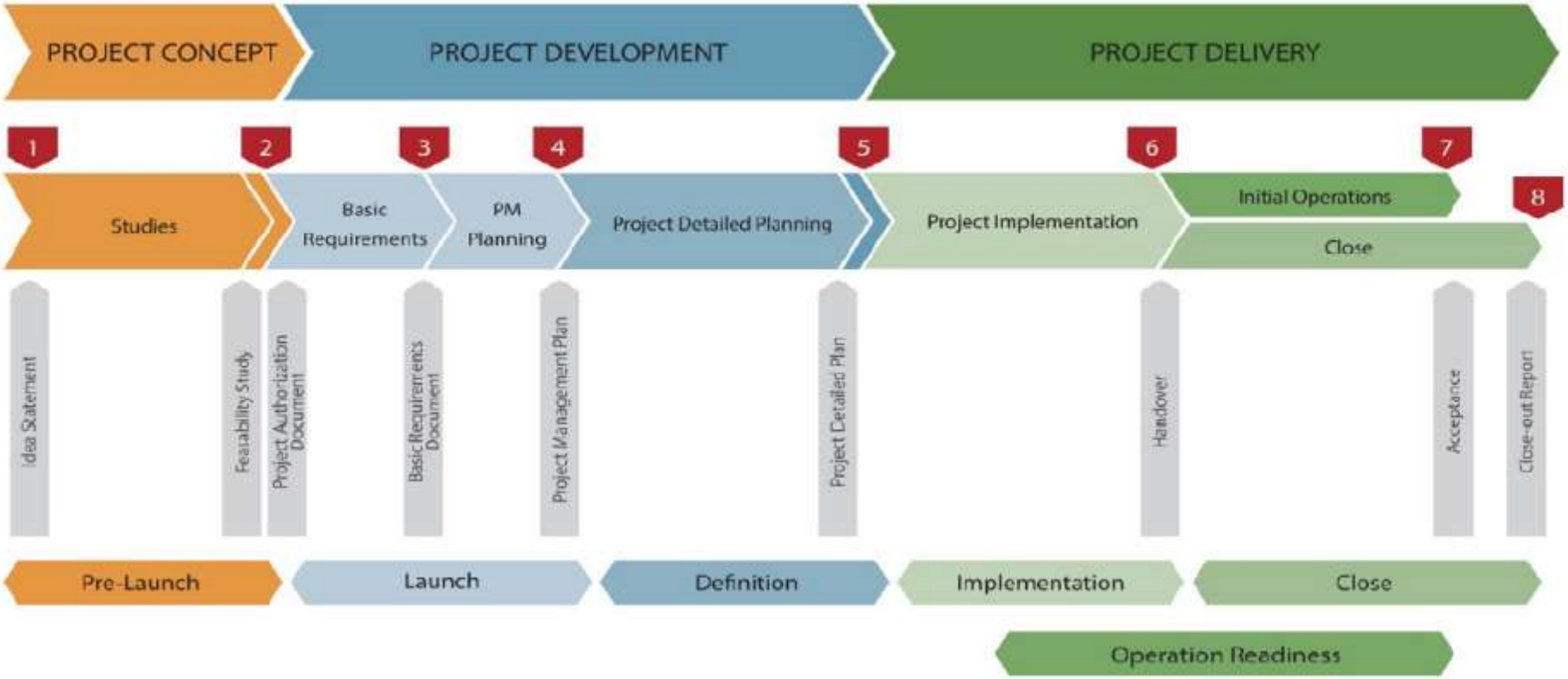
... as opposed to a focus on technology

- Technology applied without a suitable roadmap will not result in significant payoff.
- Technology provides the most benefit in the context of an appropriate process roadmap.

SW life cycle, software (quality) assurance standards/models



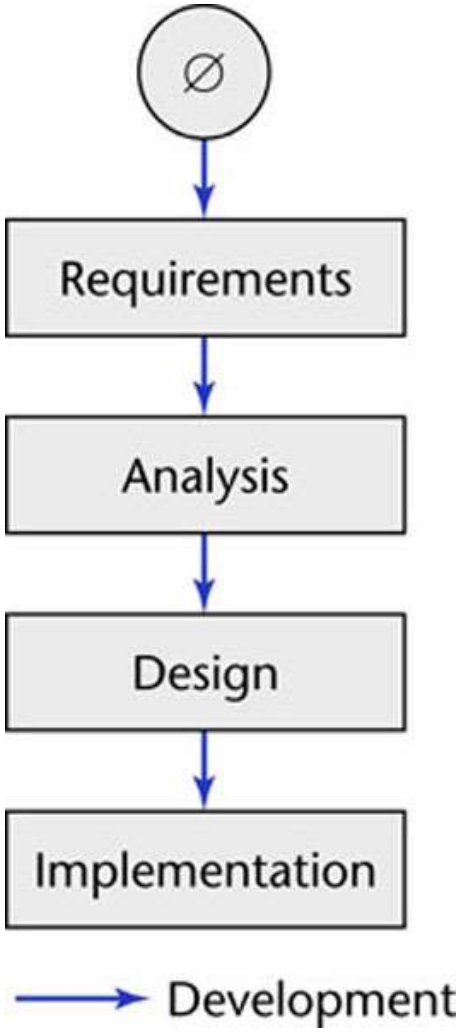
SW Project life cycle



Software Development in Theory

Ideally, software is developed:

- Linear
- Starting from scratch



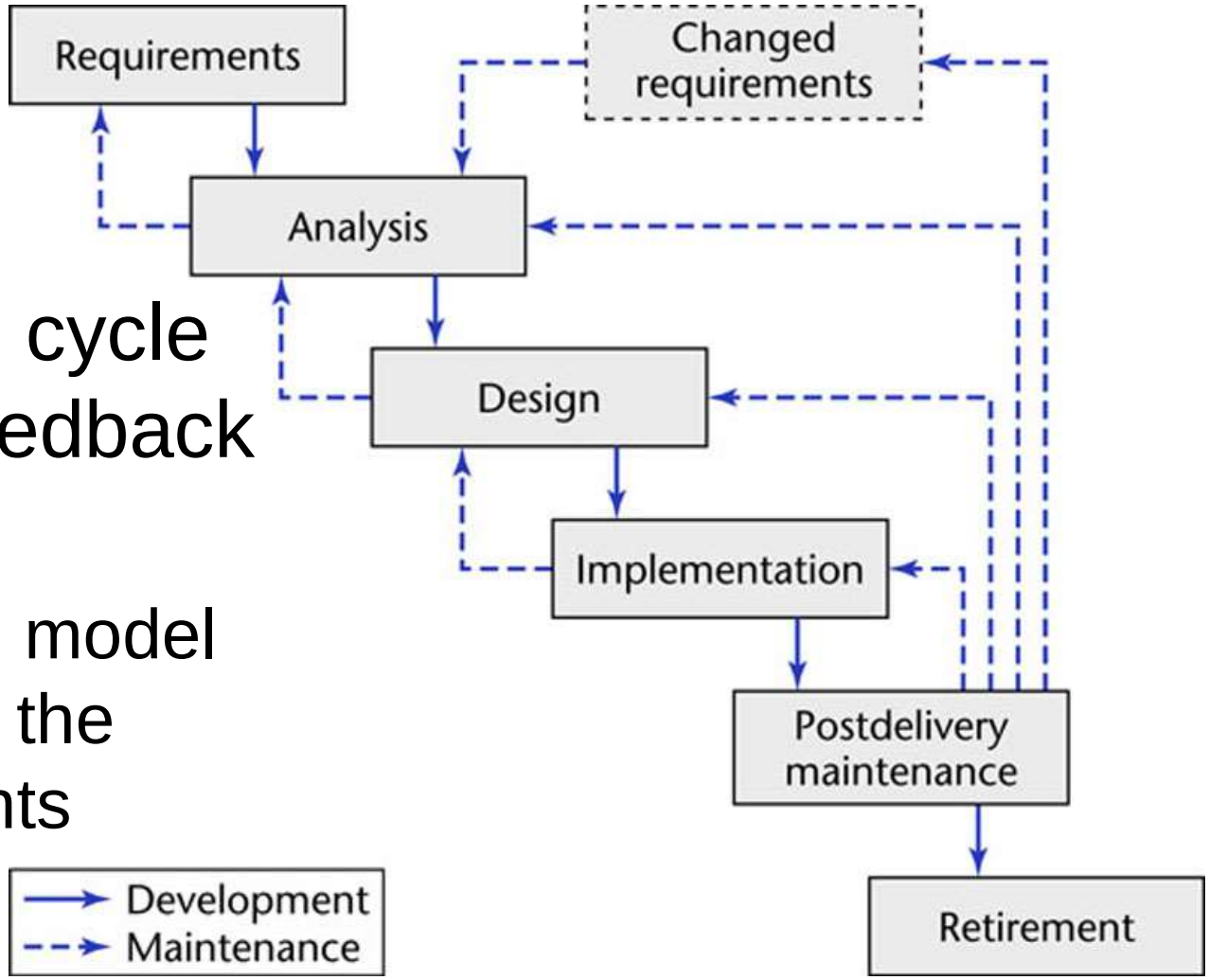
Software Development in Practice

In the real world, software development is totally different and is more chaotic

- Software professionals make mistakes
- The client's requirements change while the software product is being developed
- A software product is a model of the real world, and the real world is continually changing.

Waterfall Life-Cycle Model

- The linear life cycle model with feedback loops
  - The waterfall model cannot show the order of events



Waterfall Life-Cycle Model (Cont.)

No phase is complete until the **documentation** for that phase has been completed and the products of that phase have been approved by the **software quality assurance** (SQA) group.

If the products of an earlier phase have to be changed as a consequence of following a **feedback loop**, that earlier phase is deemed to be complete only when the documentation for the phase has been modified and the modifications have been checked by the SQA group.

Waterfall Life-Cycle Model (Cont.)

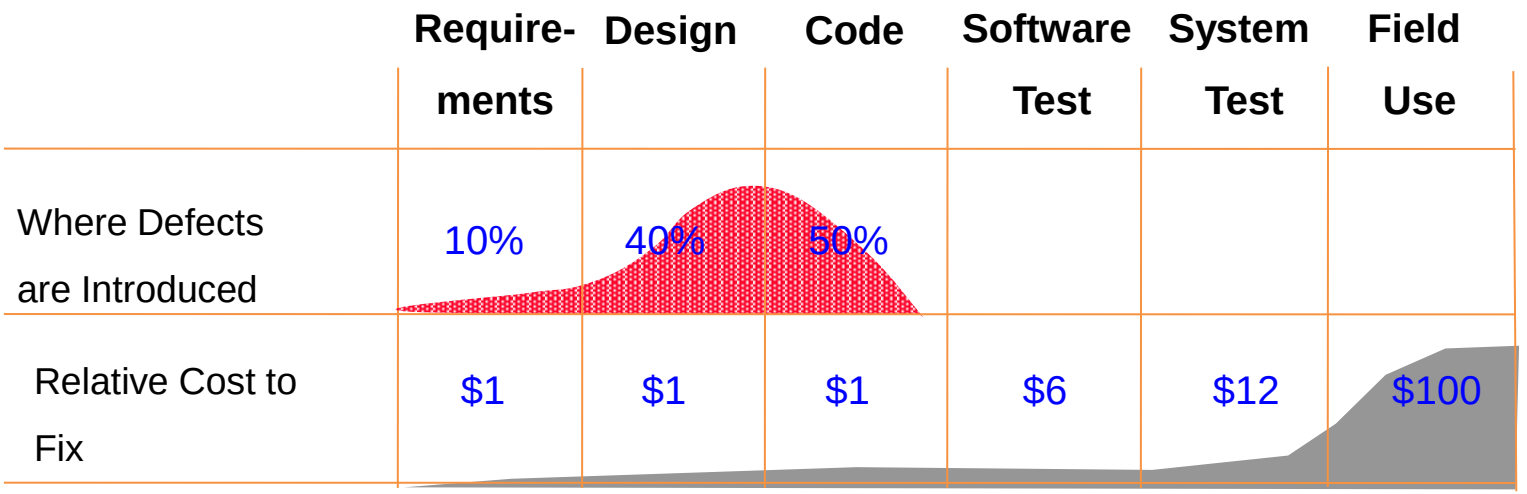
Advantages:

- Documentation is provided at each phase
- All the products of each phase (including the documentation) are meticulously checked by SQA. ➔ Maintenance is easier

Disadvantages:

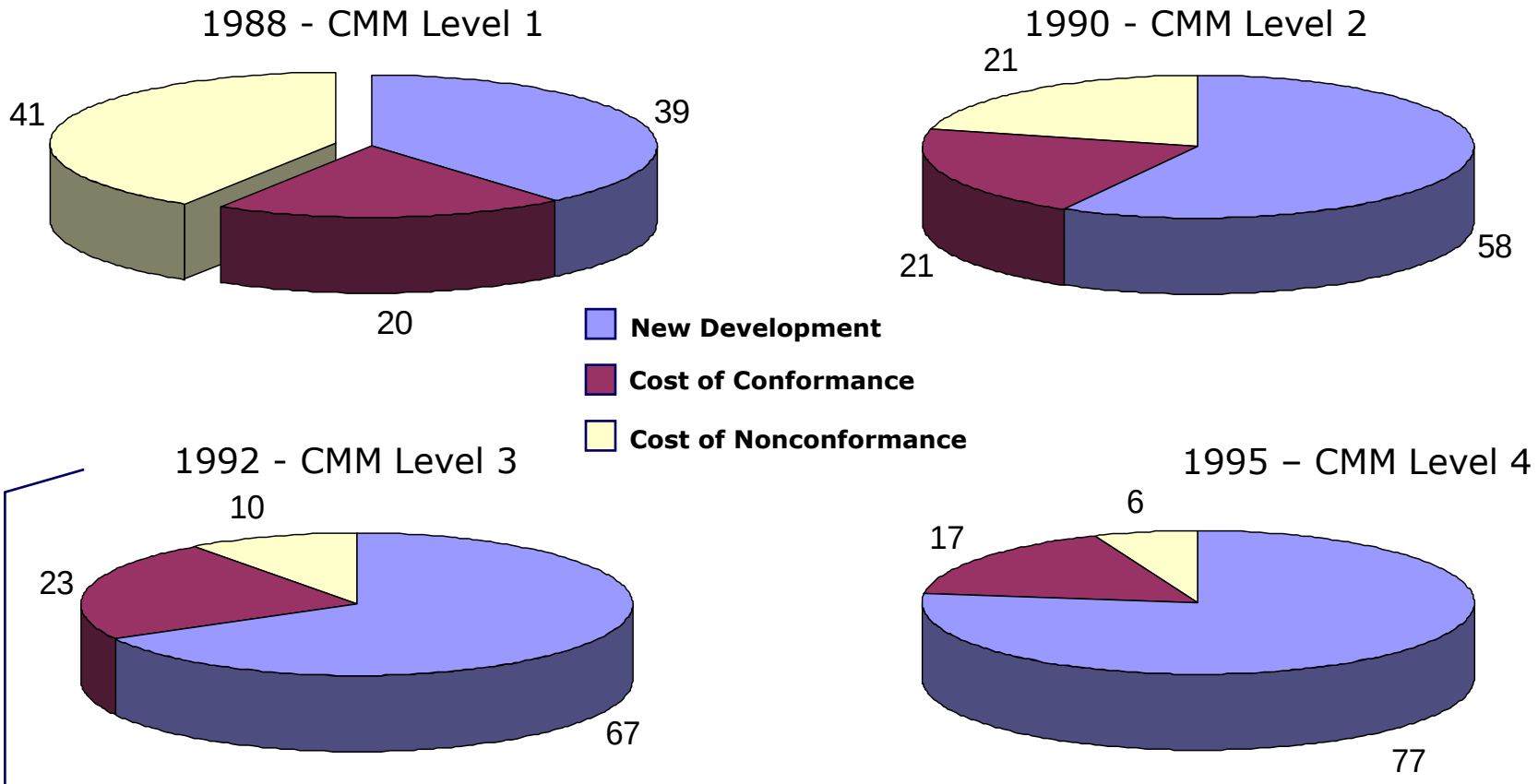
- Specification documents are long, detailed, and boring to read.

Defects : Insertion Pattern & Cost of Removal



Source: SEPG Asia Pacific 2009
presented by Ravindra Nath, KUGLER MAAG CIE GmbH

The shift to increased profitability

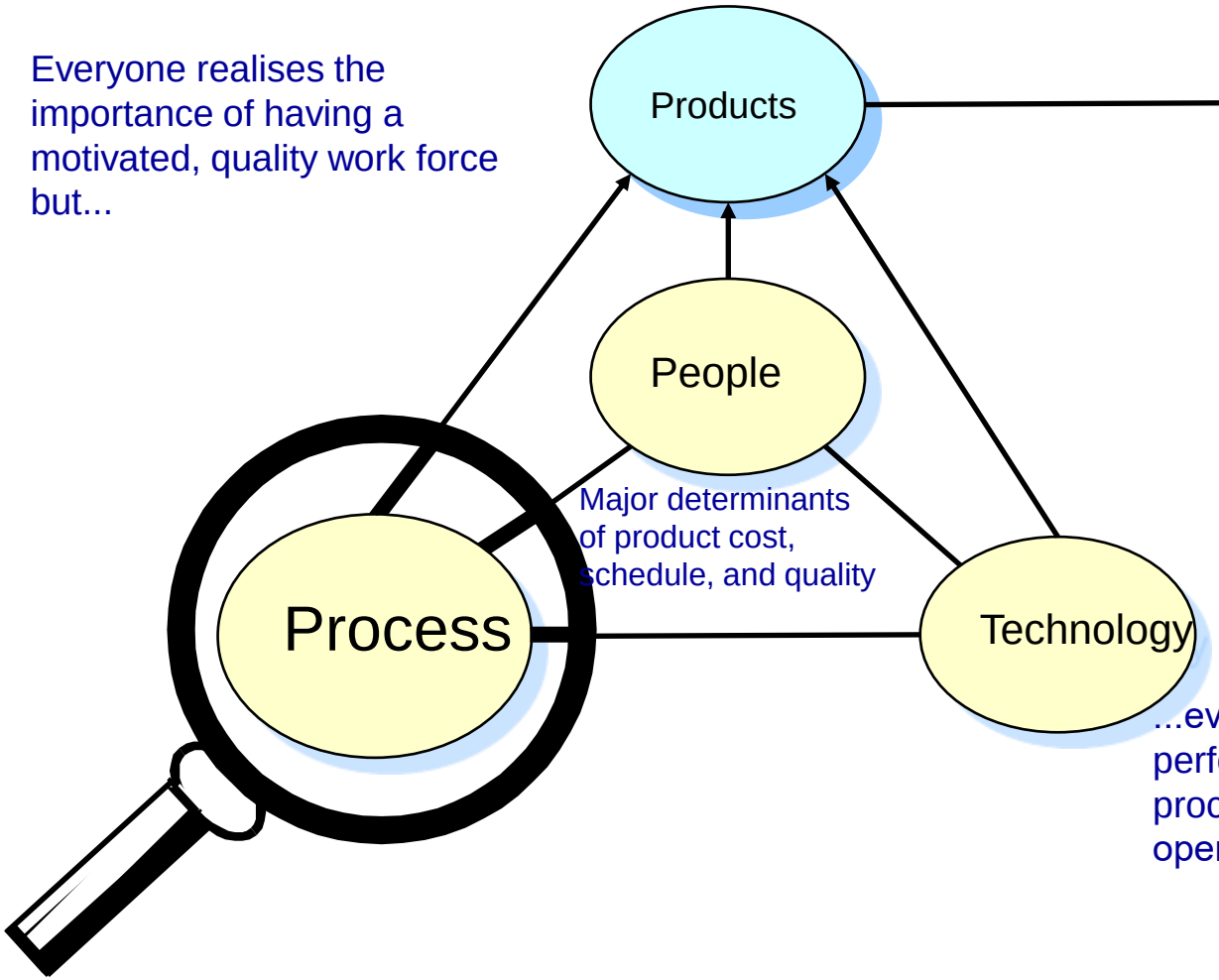


ROI 7.7:1, Productivity ⇧140%, $4.48M savings over 6 projects in 1 year

Source: Raytheon Electronic Systems Experience in Software Process Improvement, CMU/SEI-95-TR-017, November 1995

Quality Leverage Points

Everyone realises the importance of having a motivated, quality work force but...



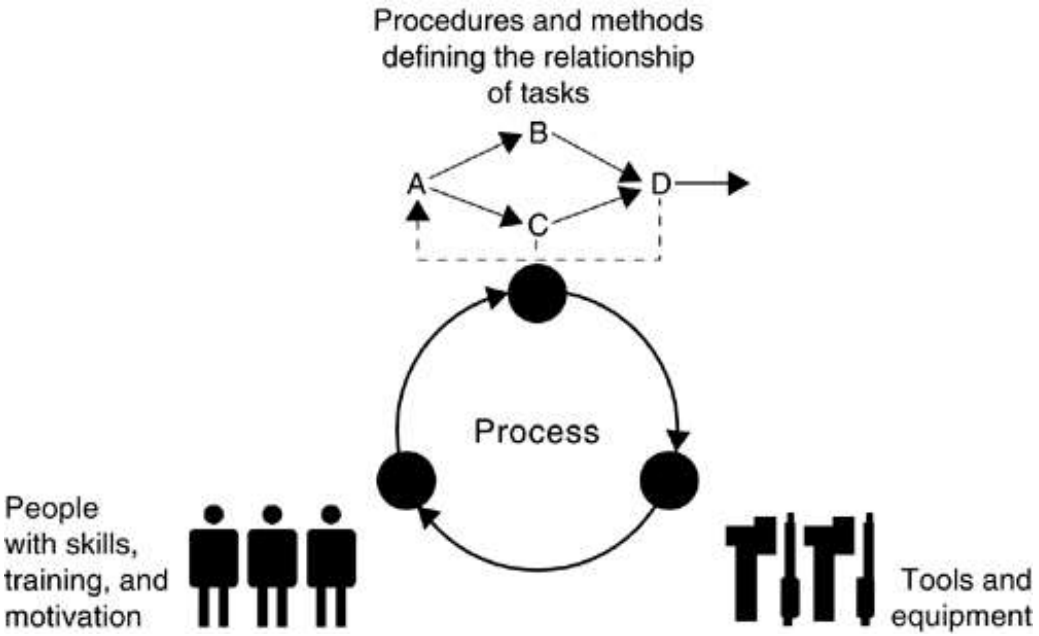
CUSTOMER SATISFACTION

...even our finest people can't perform at their best when the process is not understood or operating "at its best."

Why using models?

"All models are wrong,
but some are useful."

George Box



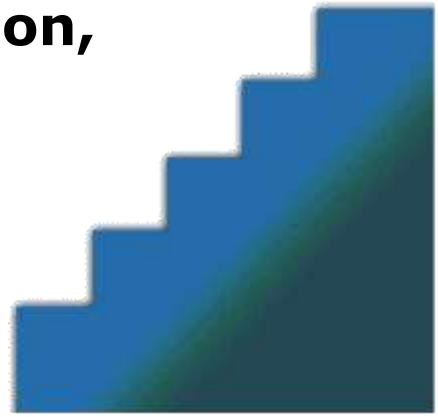
What is a Capability Maturity Model?

Capability Maturity Model:

A reference model of mature practices in a specified discipline, used to assess a group's capability to perform that discipline

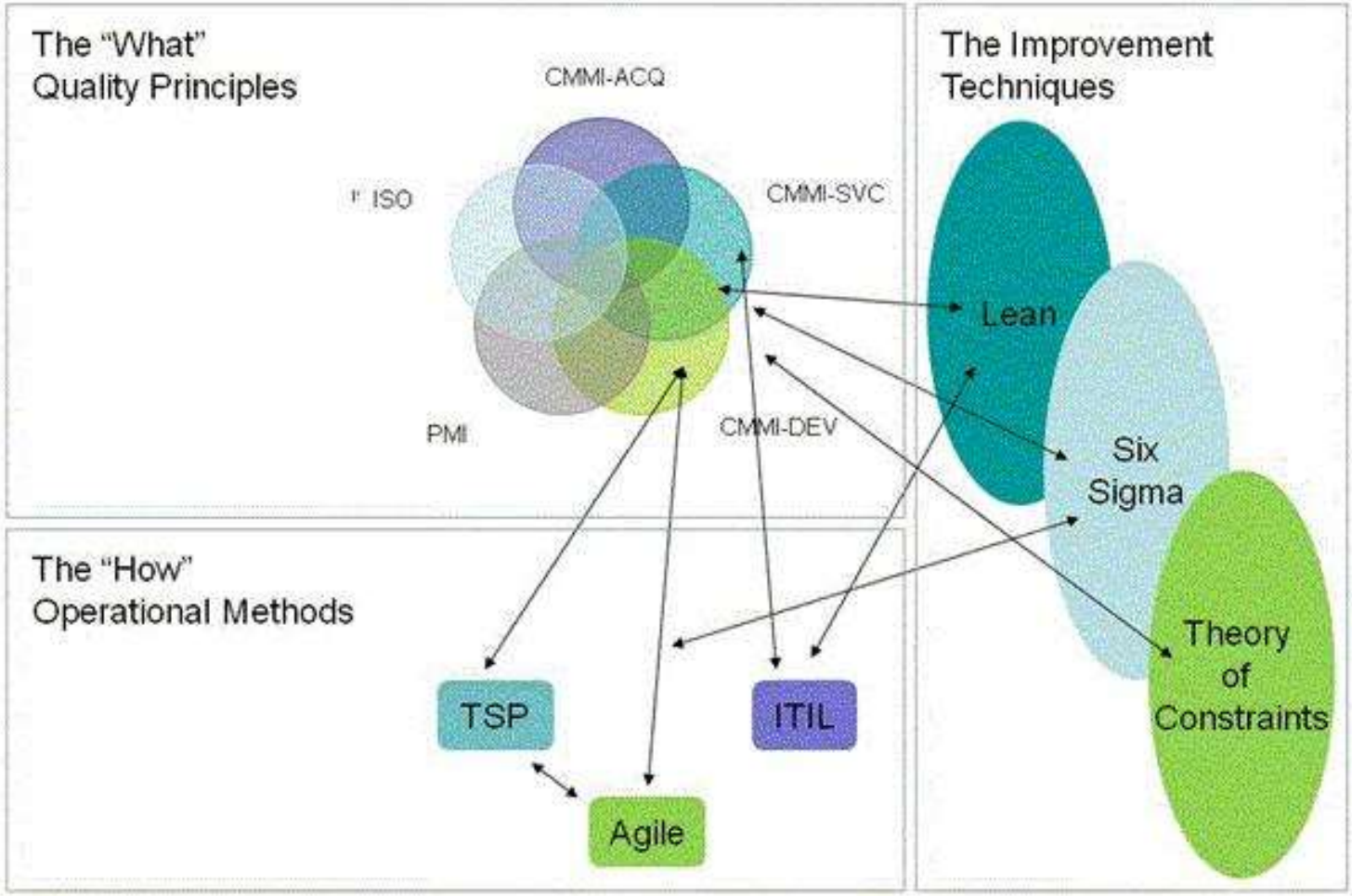
CMMs differ by

- **Discipline (software, systems, acquisition, etc.)**
- **Structure (staged versus continuous)**
- **How Maturity is Defined (process improvement path)**
- **How Capability is Defined (institutionalisation)**



"Capability Maturity Model®" and CMM® are used by the Software Engineering Institute (SEI) to denote a particular class of maturity models

CMMI and other models



CMMI (SEI/CMU) – reference model de facto industrial standard
CMMI DEV, CMMI ACQ, CMMI SVC

5 Focus on process improvement

Optimizing
Measurably increased process capabilities

4 Process measured and controlled

Quantitatively Managed
Use of statistical and other quantitative techniques in managing the processes and results

3 Process characterized for the **organization** and is proactive

Defined
Commonality among projects allows more uniform estimation of performance.

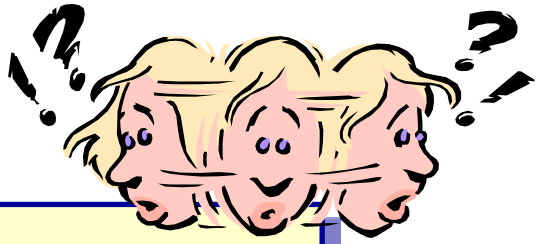
2 Process characterized for **projects** and is often reactive

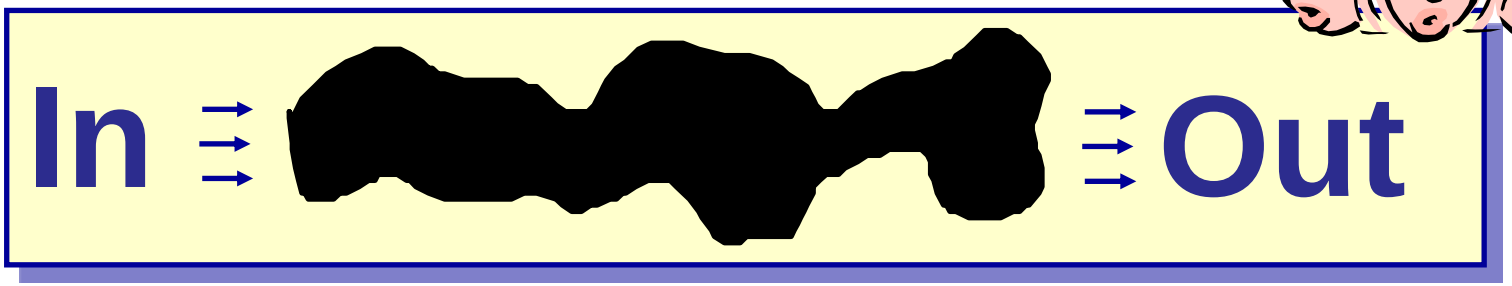
Managed *(ex "repeatable")*
- Requirements flow in.
- Plans are developed in accordance with policies.
- Activities are performed in accordance with plans.
- Measurements and reviews occur at defined points.
- The product flows out and (usually) works

1 Process unpredictable, poorly controlled and reactive

Performed
- Requirements flow in.
- A product is (sometimes) produced by some amorphous process.
- The product flows out and (we hope) works.

ML1: Performance Is Unpredictable





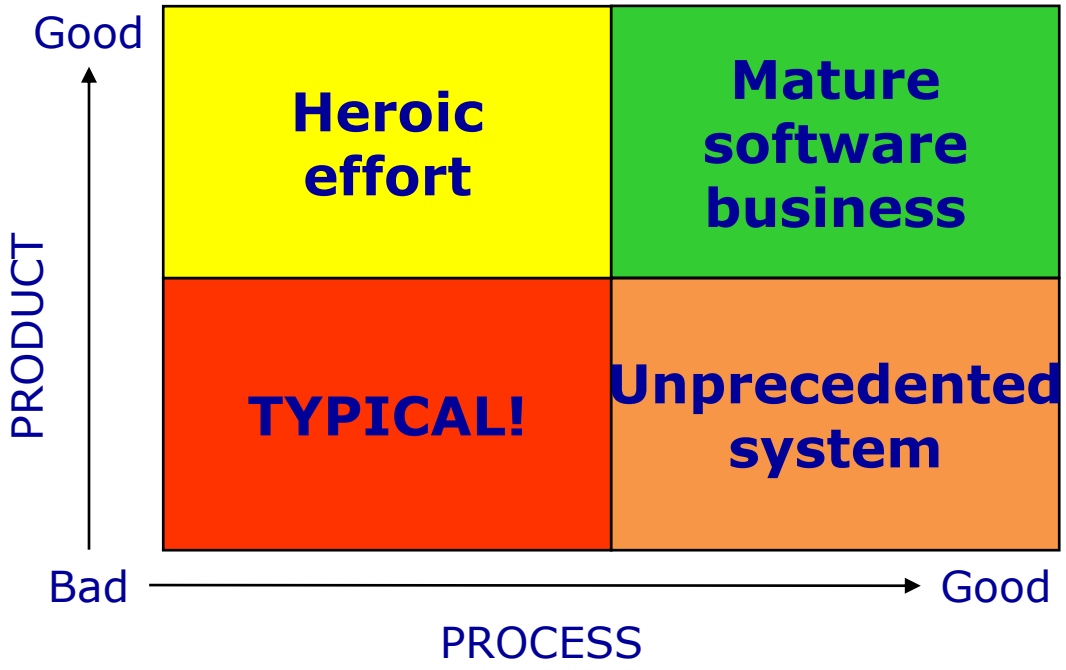
Requirements flow in.

A product is (sometimes) produced by some amorphous process.

The product flows out and (we hope) works.

REMEMBER? Corporate excellence – INTERNAL

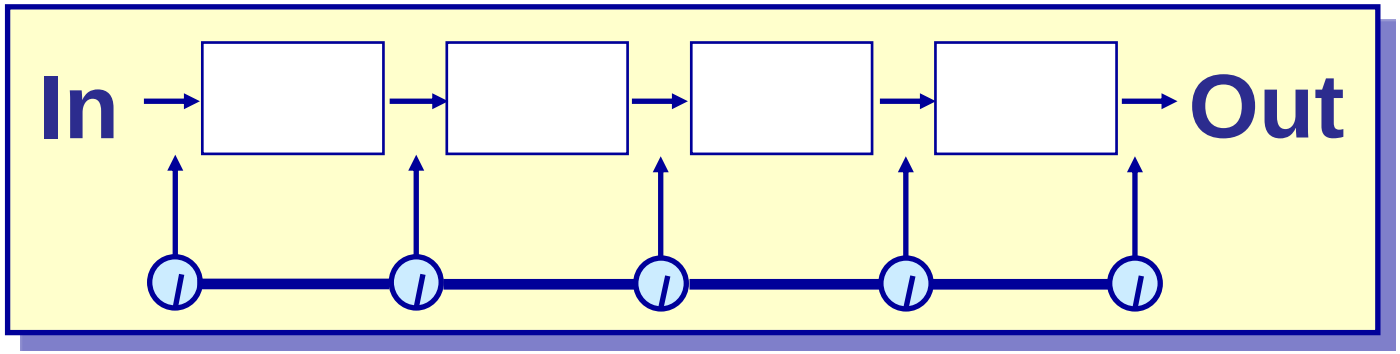
The corporate excellence is BASED on good internal processes



"The quality of a product is largely determined by the quality of the process that is used to develop and maintain it."

Based on TQM principles as taught by Shewhart, Juran, Deming and Humphrey.

ML2: Process Is "Managed"



Requirements flow in.

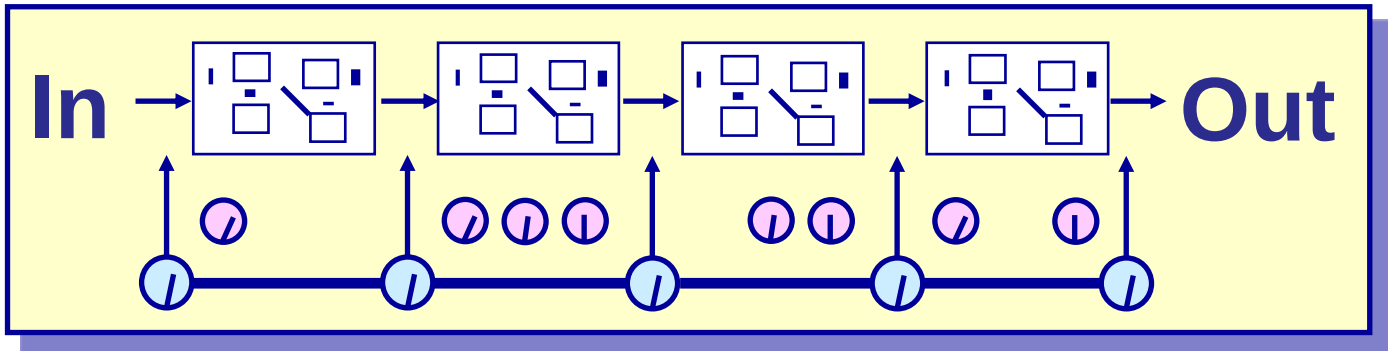
Plans are developed in accordance with policies.

Activities are performed in accordance with plans.

Measurements and reviews occur at defined points.

The product flows out and (usually) works.

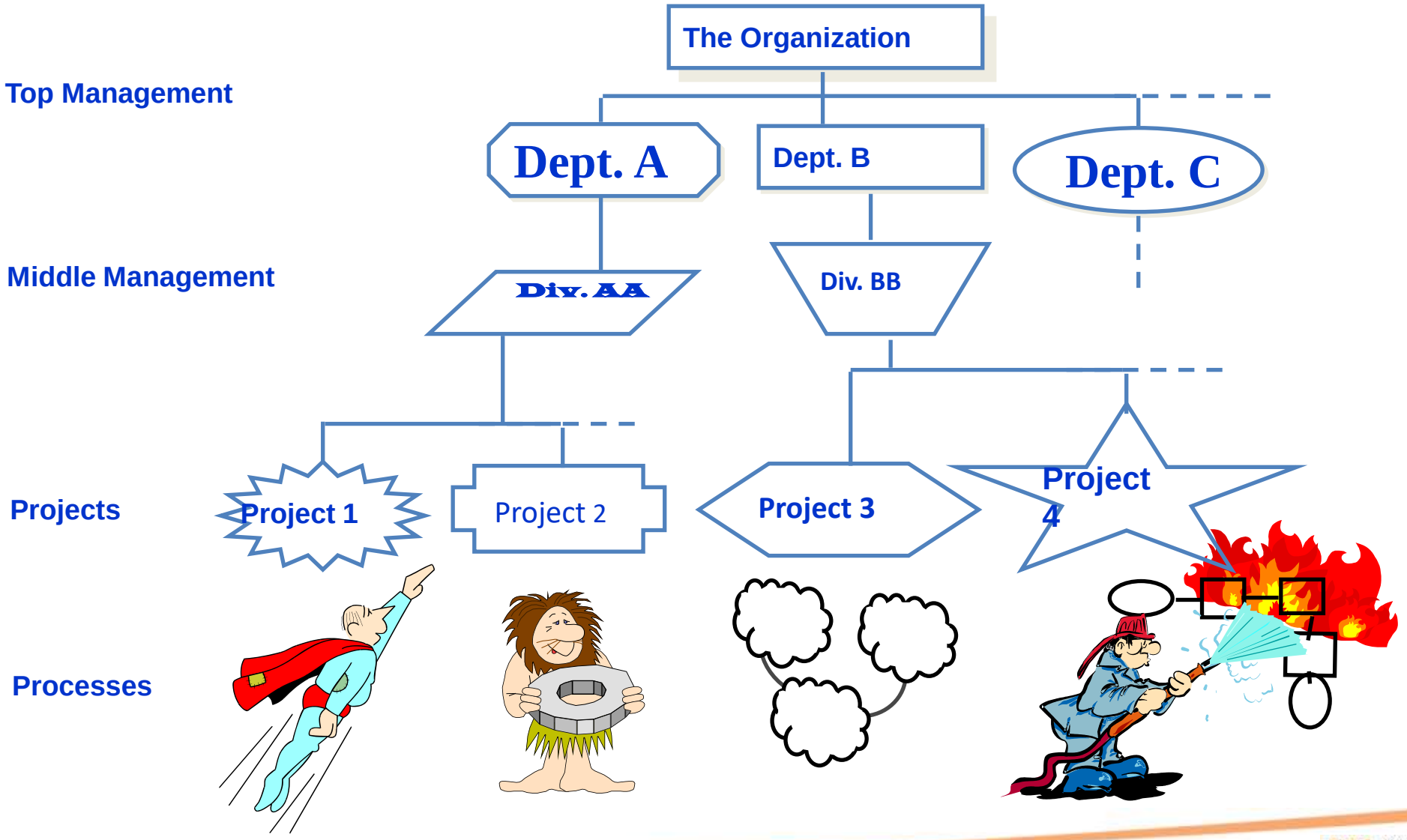
ML3: Managed According to a Defined Process



Commonality among projects allows more uniform estimation of performance.

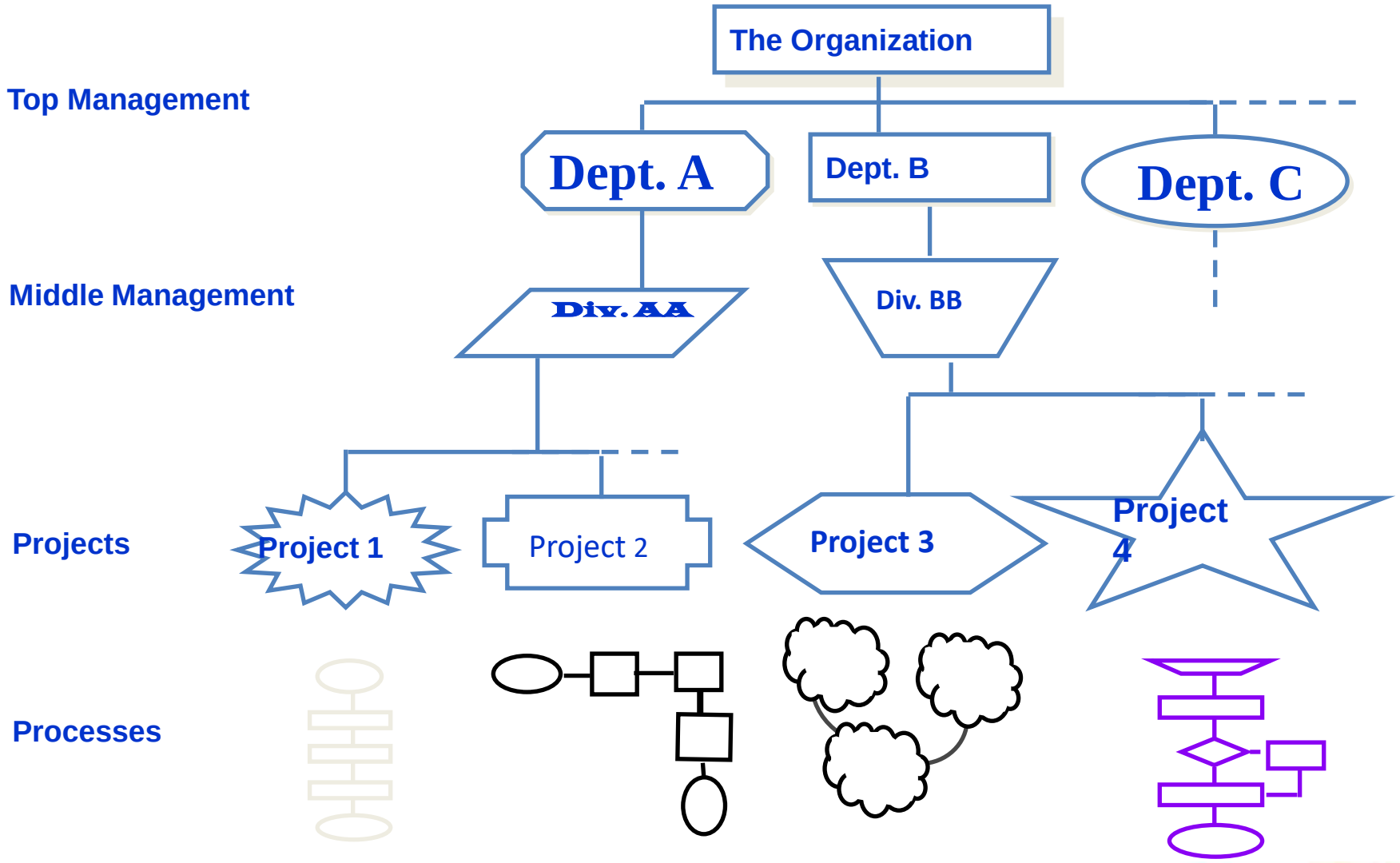
Sample Level 1 Organization

few processes in place



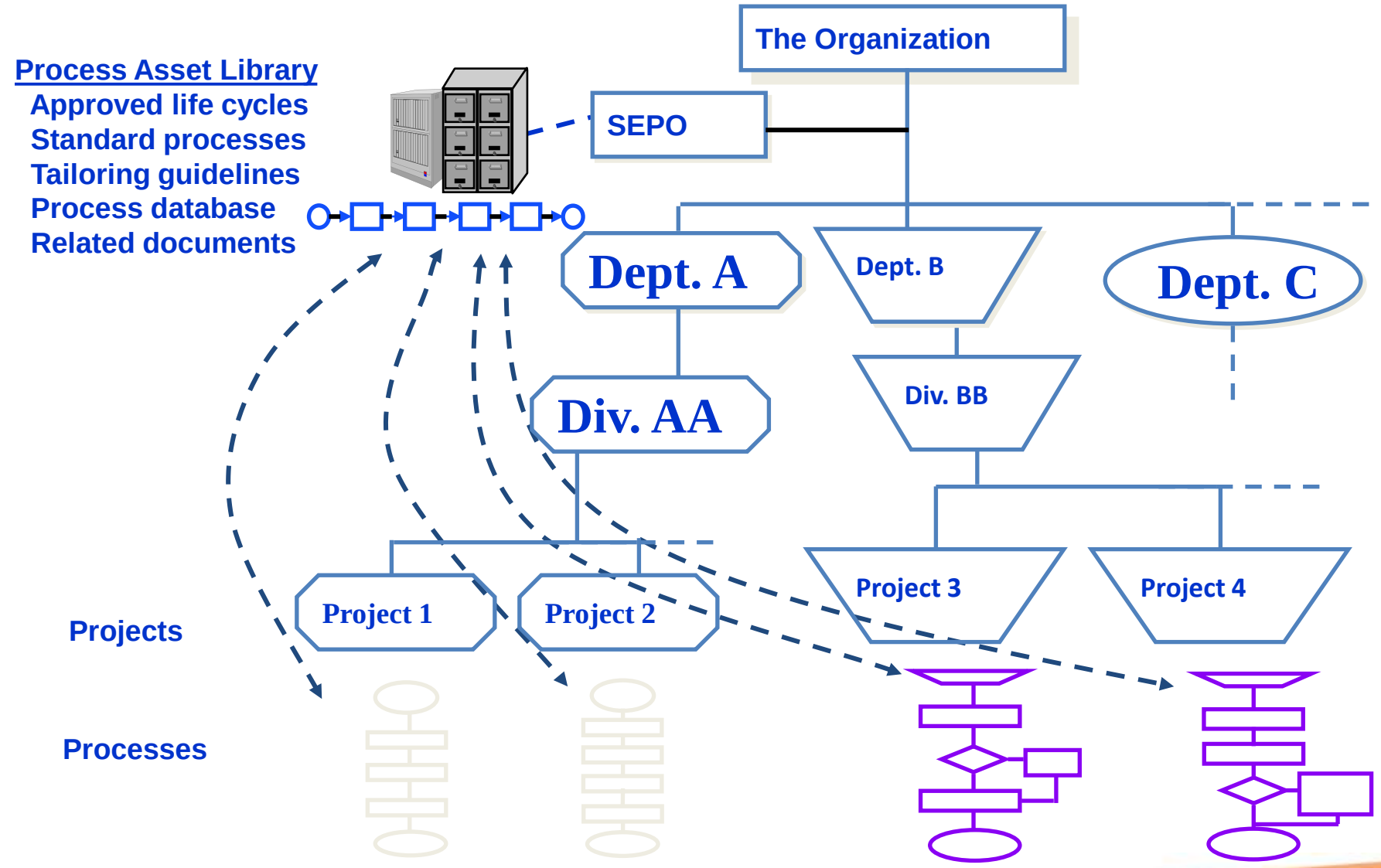
Sample Level 2 Organization

many processes in place; but they are project-specific



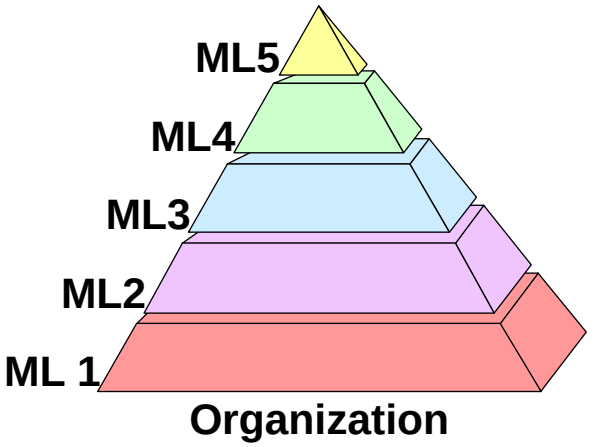
Sample Level 3 Organization

processes based on organization's Process Asset Library (PAL)



CMMI Representations

Staged



Maturity Level 5 OID, CAR

Maturity Level 4 OPP, QPM

Maturity Level 3 RM, TS, PI, VER, VAL, OPF, OPD, OT, IPM, RSKM, DAR

Maturity Level 2 REQM, PP, PMC, MA, PPQA, CM, SAM

Process Areas

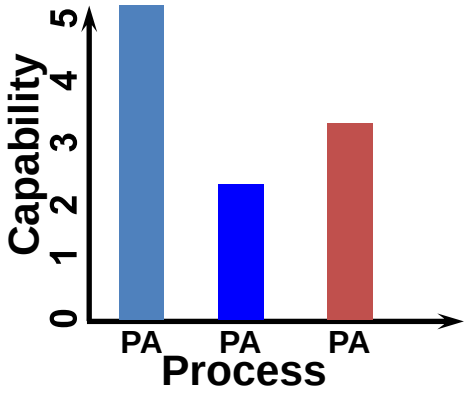
Organizational Innovation & Deployment (OID)
Causal Analysis and Resolution (CAR)

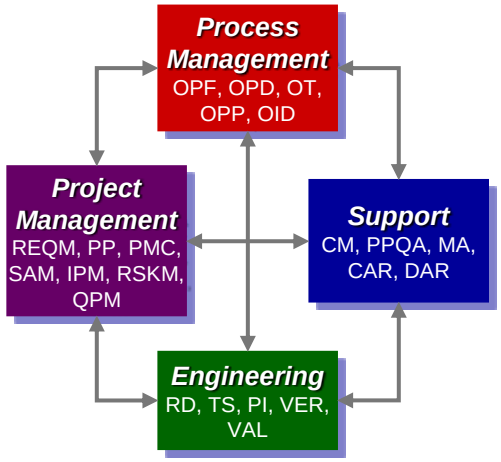
Organizational Process Performance (OPP)
Quantitative Project Management (QPM)

Requirements Development (RD)
Technical Solution (TS)
Product Integration (PI)
Verification (VER)
Validation (VAL)
Organizational Process Focus (OPF)
Organizational Process Definition (OPD) + IPPD
Organizational Training (OT)
Integrated Project Management (IPM) + IPPD
Risk Management (RSKM)
Decision Analysis and Resolution (DAR)

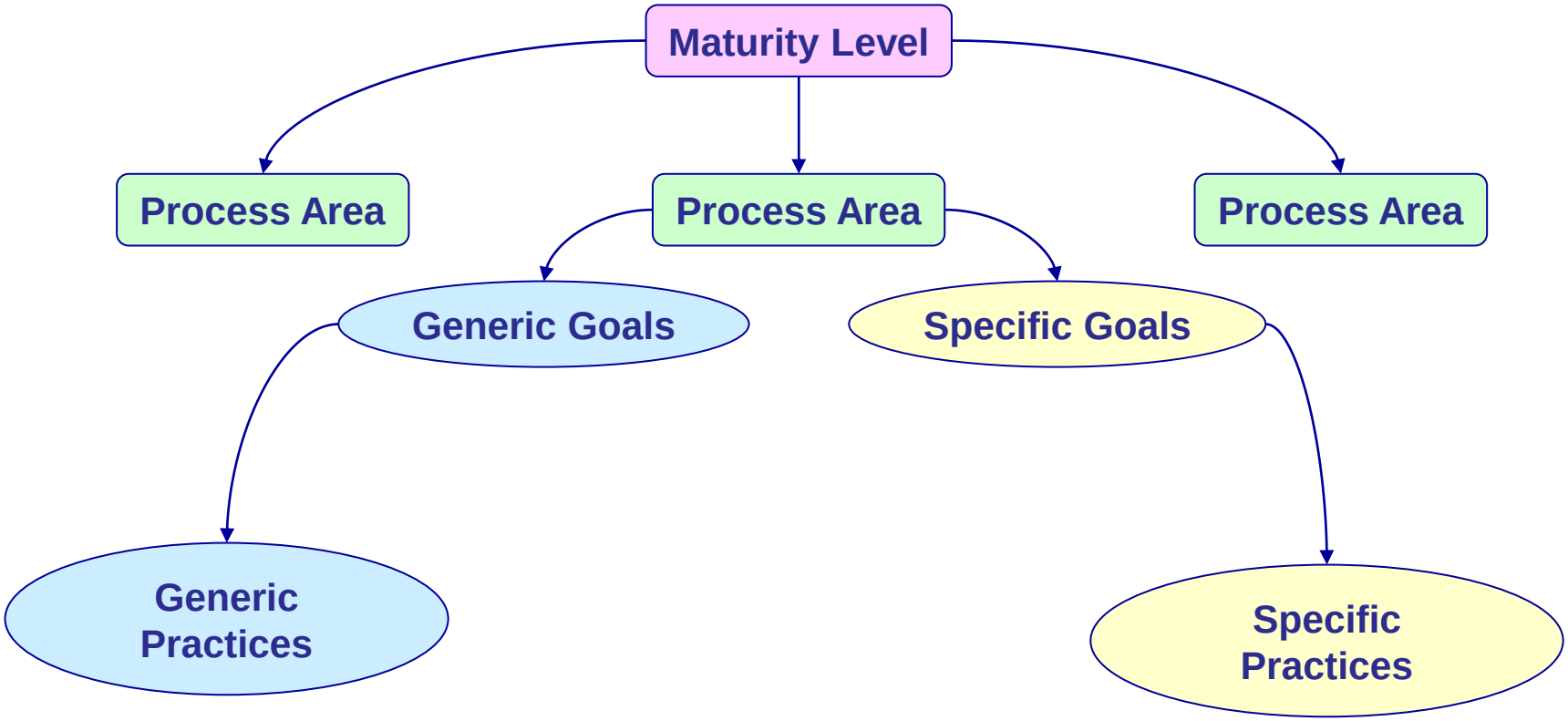
Requirements Management (REQM)
Project Planning (PP)
Project Monitoring and Control (PMC)
Supplier Agreement Management (SAM)
Measurement and Analysis (MA)
Process and Product Quality Assurance (PPQA)
Configuration Management (CM)

Continuous





Structure of the CMMI Staged Representation



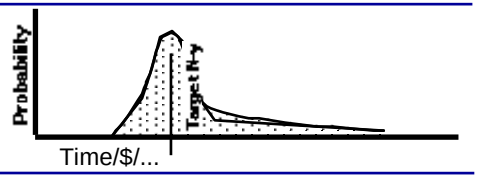
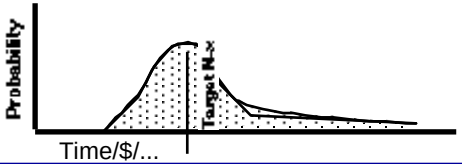
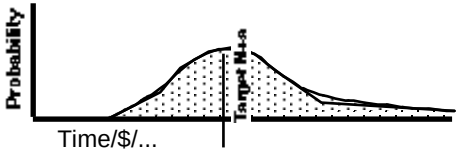
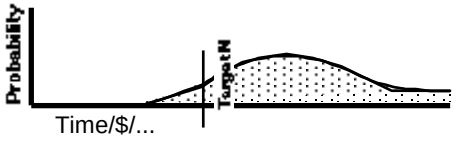
Maturity Levels Cannot Be Skipped

A level provides a necessary foundation for effective implementation of processes at the next level.

- Higher level processes are easily sacrificed without the discipline provided by lower levels.
- The effect of innovation is obscured in a noisy process.

Higher maturity level processes may be performed by organisations at lower maturity levels, with risk of not being consistently applied in a crisis.

Evolution of Process Capability

Level	Process Characteristics	Predicted Performance
5 Optimising	Process improvement is institutionalised	
4 Quantitatively Managed	Product and process are quantitatively controlled	
3 Defined	Software engineering and management processes are defined and integrated	
2 Managed	Project management system is in place; performance is repeatable	
1 Initial	Process is informal and unpredictable	

DO NOT FORGET!!!

Process ≠ Bureaucracy

Process = Work

ML2: Managing the Project Involves

Understand and commit to the requirements

Estimating the scope and work that needs to be performed

Developing mechanisms to acquire identified products

Developing a project plan

Getting commitments to the plan

Working with suppliers to acquire identified products

Monitoring progress against the plan

Identifying and analyzing risks

Taking action to address significant deviations from the plan

Taking action to appropriately mitigate risks

Project Management PAs (overview)

Requirements management (REQM)

- **SG1: Manage requirements**

Project Planning (PP)

- SG1: Establish Estimates
- SG2: Develop a project plan
- SG3: Obtain Commitment to the plan

Project Monitoring and Control (PMC)

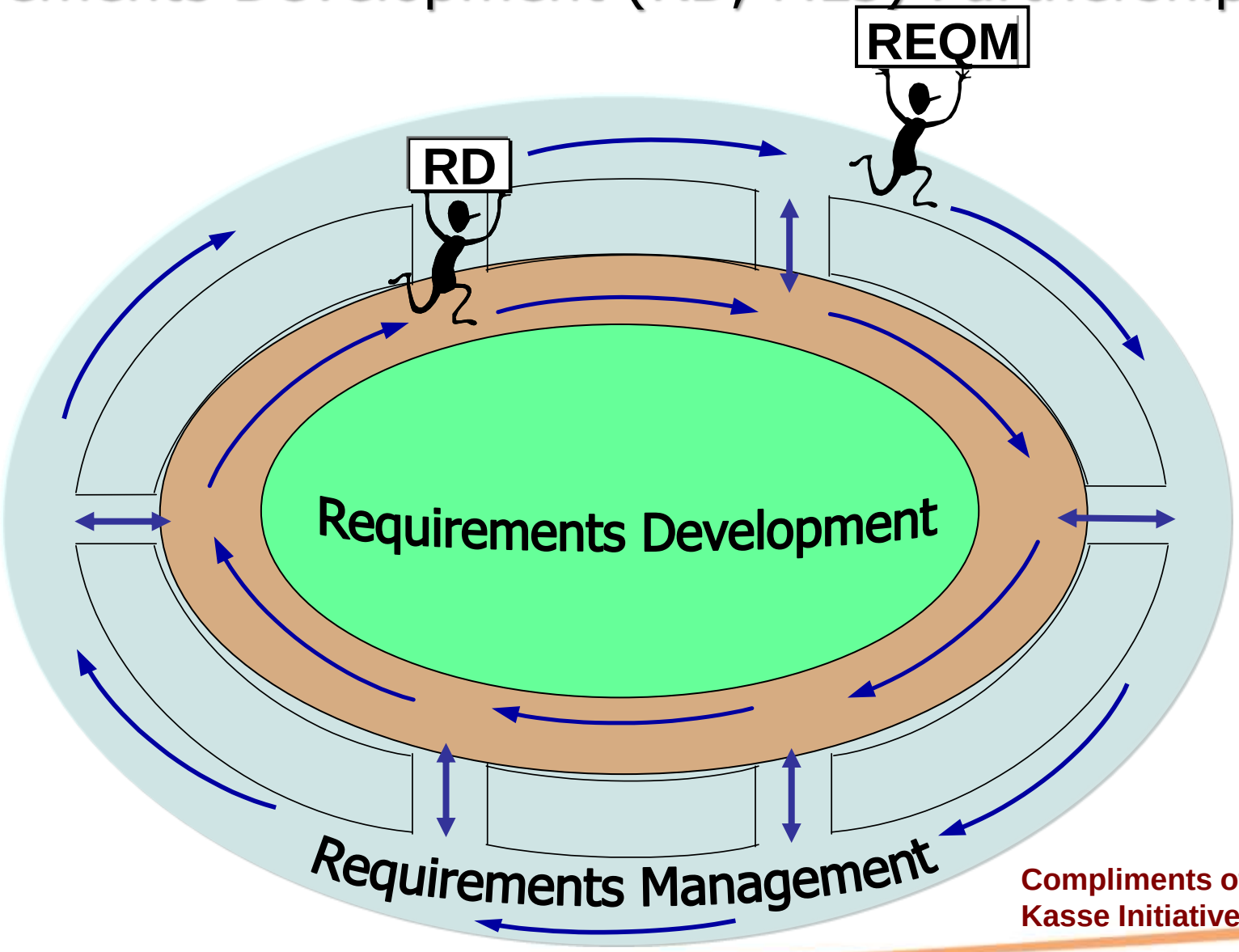
- SG1: Monitor Project Against Plan
- SG2: Manage Corrective action to closure

Think about: What Product/SW Development Needs?

Establishing and maintaining sets of requirements

- customer requirements
- product requirements
- product component requirements
- managing the requirements as the product evolves

The Requirements Management (REQM, ML2) and Requirements Development (RD, ML3) Partnership



Compliments of
Kasse Initiatives, LLC

Requirements Management (REQM)

*The purpose of Requirements Management (REQM) is to **manage the requirements** of the project's products and product components and to identify inconsistencies between those requirements and the project's plans and work products.*

SG1: Manage Requirements
Requirements are managed and inconsistencies with project plans and work products are identified.

The process area also has generic goals to support institutionalization.

When Requirements Management Is Not Done Well…

Requirements are accepted by staff from **any source** they deem to be authoritative.

The project experiences a high level of **requirements changes**.

There are high levels **of rework** throughout the project.

There is an inability to prove that the **product meets the approved requirements**.

Lack of requirements traceability often results **in incomplete or incorrect testing** of the product.

Relevant Terminology

Requirements traceability

A discernable association between requirements and related requirements, implementations, and verifications.

Bidirectional traceability

An association among two or more logical entities that is discernable in either direction (i.e., to and from an entity).

Requirements Management (REQM) Specific Practices

SP 1.1 Obtain an **Understanding** of Requirements

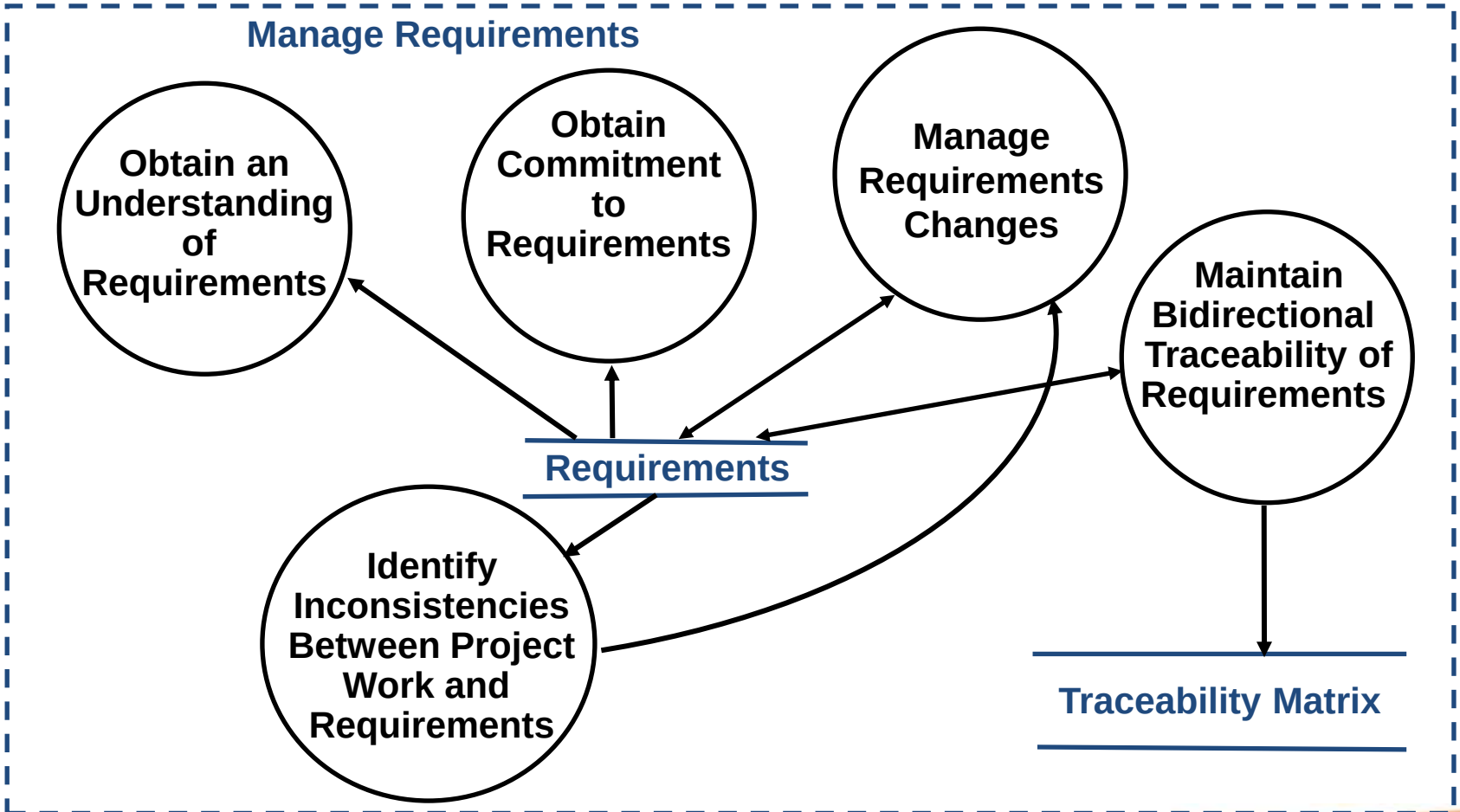
SP 1.2 Obtain **Commitment** to Requirements

SP 1.3 Manage Requirements **Changes**

SP 1.4 Maintain **Bidirectional Traceability** of Requirements

SP 1.5 **Identify Inconsistencies** between project work and requirements

Requirements Management (REQM)



Bidirectional traceability

An association among two or more logical entities that is discernable in either direction (i.e., to and from an entity).

Remember: Why do we need bidirectional traceability???

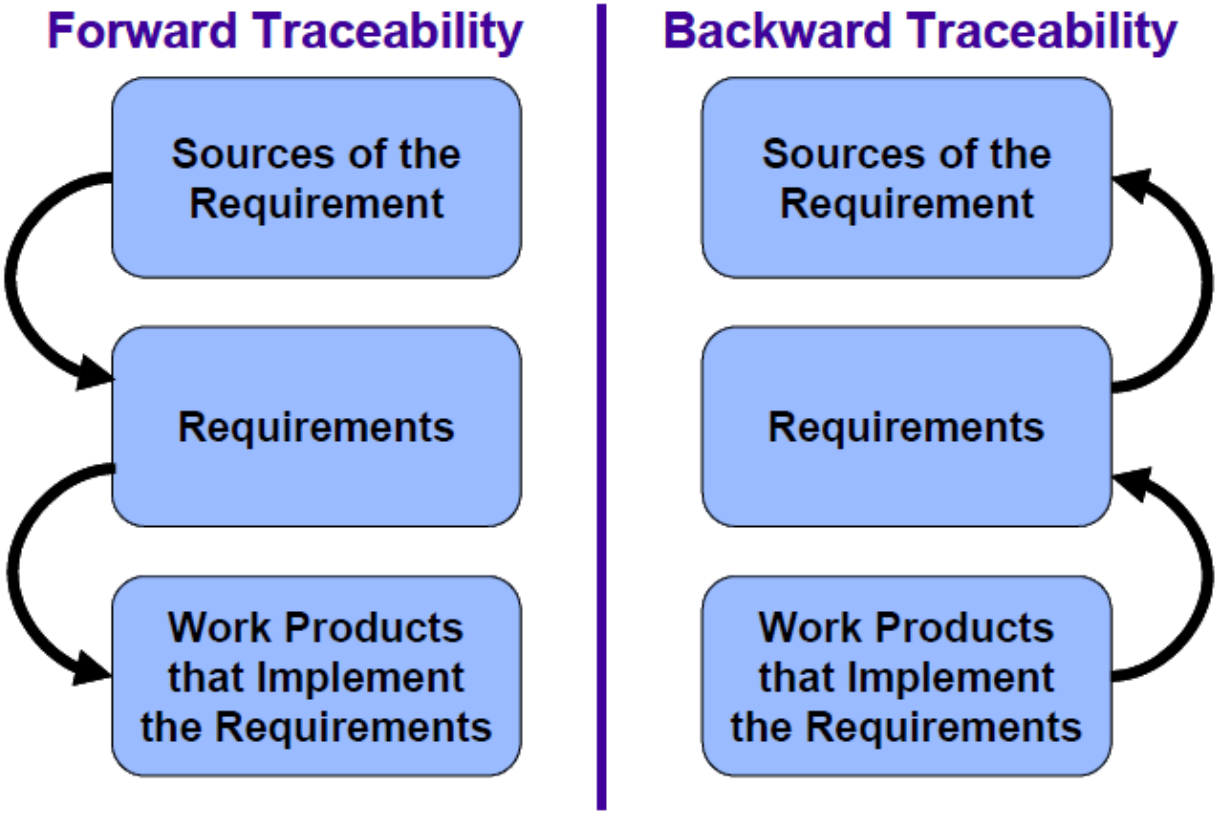


Figure 1: Bidirectional (Forward & Backward) Traceability

Benefits:

Analyze the impact of a change

- All **work products affected** by a changed requirement
- All **requirements affected** by a change or defect in a work product

Assess current status of the requirements and the project

- Identify **missing** requirements
- Identify **gold plating** (overdoing)

REQM Practices implementation:

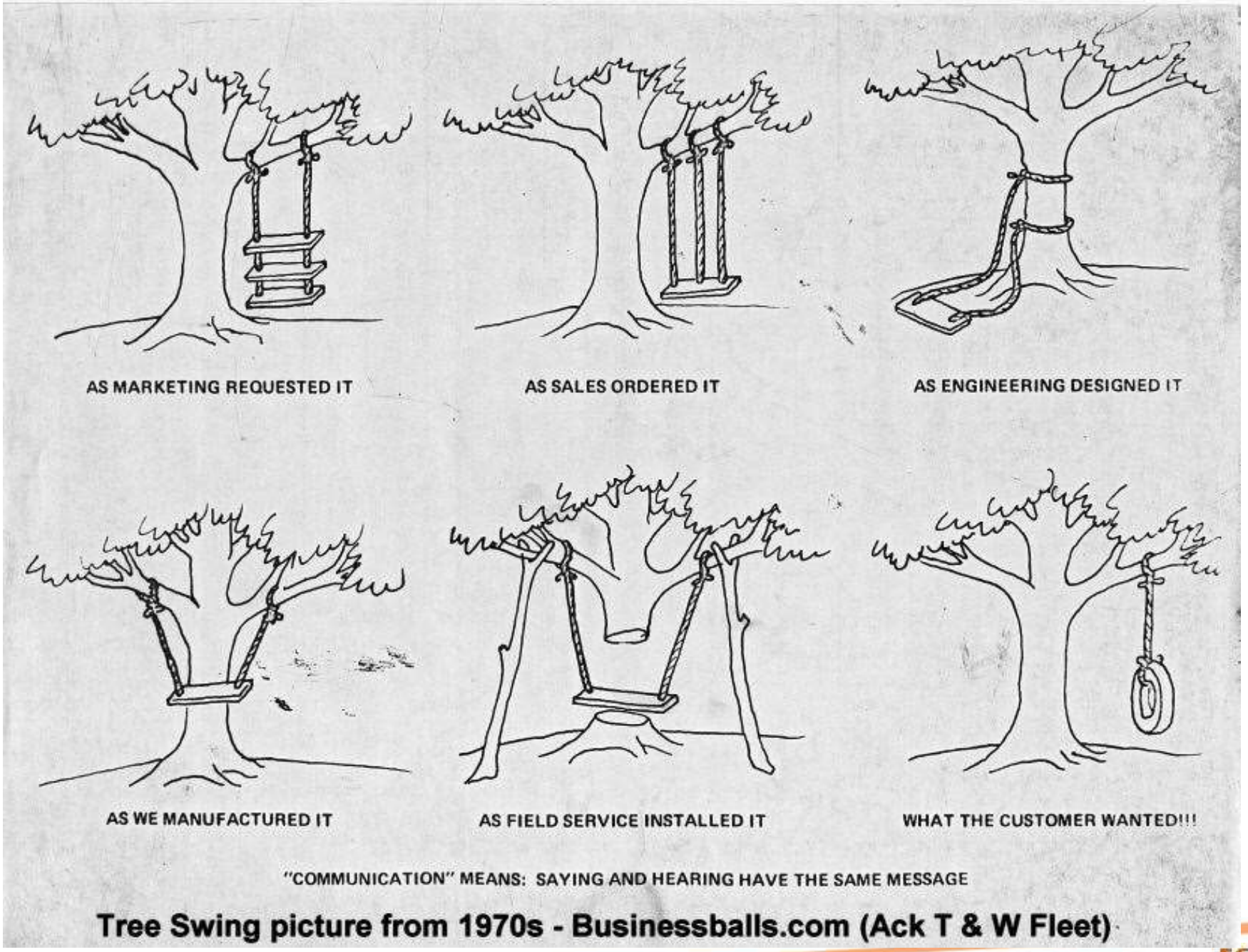
- Acceptance criteria in place?
- Requirements comply to criteria?
- Is understanding reached and is it documented? How?
- Who are the relevant stakeholders?
- Did they agree to requirements?
- Is the commitment documented? How?
- All requirements and their changes documented?
- Requirements change history and rationale documented?
- Are changes evaluated by affected stake holders?
- Bi-directional traceability among the requirements and the project plans and work products maintained?
- Are the project plan/activities/work products reviewed to assess the consistency with the (changed) requirements?
- If inconsistencies have been are corrective actions initiated to solve them?

Remember:

Defects - Insertion Pattern & Cost of Removal

	Require-ments	Design	Code	Software Test	System Test	Field Use
Where Defects are Introduced	10%	40%	50%			
Relative Cost to Fix	$1	$1	$1	$6	$12	$100

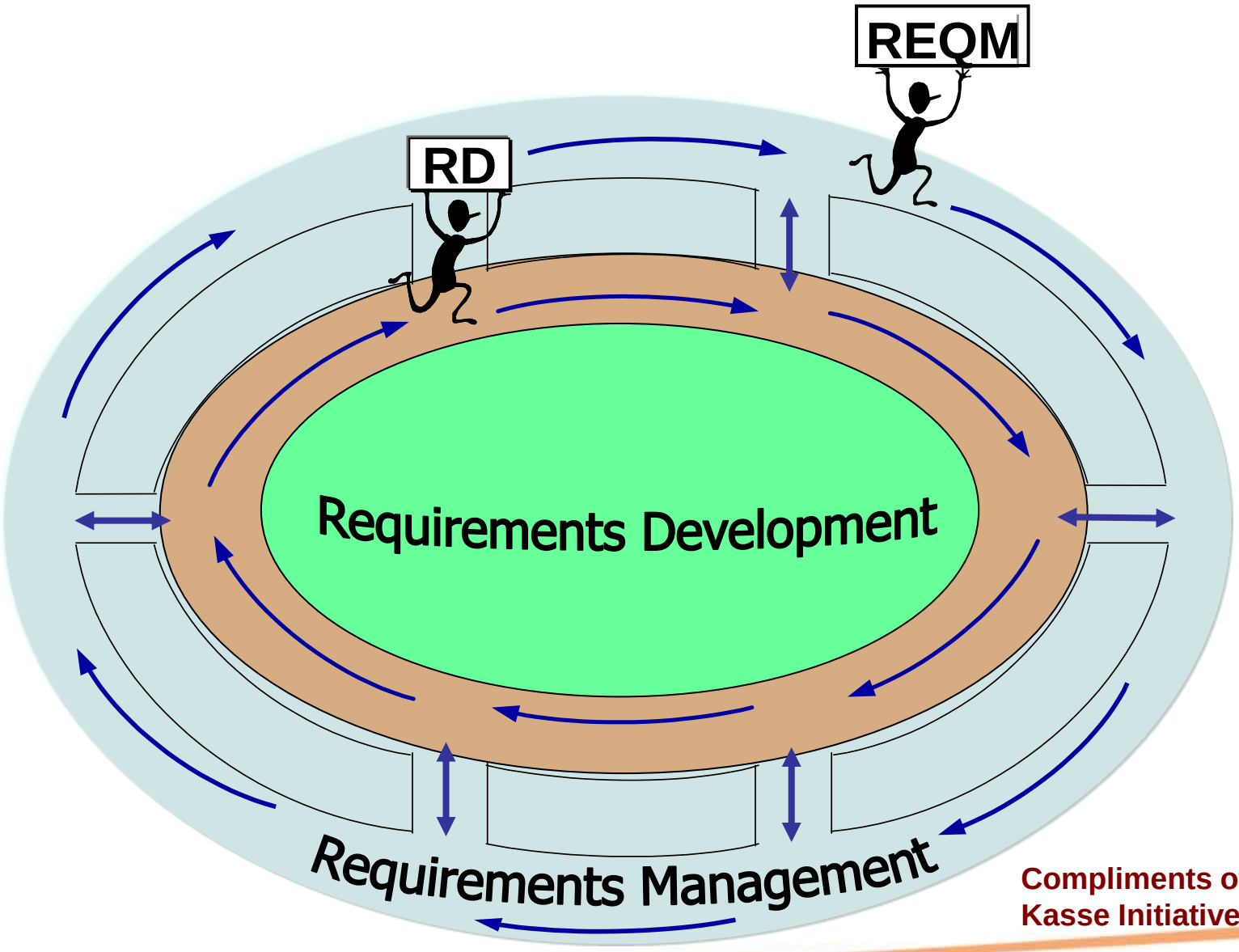
Source: SEPG Asia Pacific 2009
presented by Ravindra Nath, KUGLER MAAG CIE GmbH


AS MARKETING REQUESTED IT
AS SALES ORDERED IT
AS ENGINEERING DESIGNED IT
AS WE MANUFACTURED IT
AS FIELD SERVICE INSTALLED IT
WHAT THE CUSTOMER WANTED!!!
"COMMUNICATION" MEANS: SAYING AND HEARING HAVE THE SAME MESSAGE
Tree Swing picture from 1970s - Businessballs.com (Ack T & W Fleet)

ML3: Requirements Development

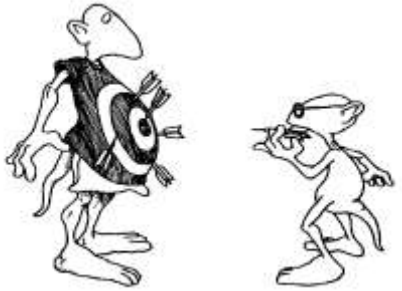
*The purpose of Requirements Development (RD) is to produce and analyze **customer, product**, and product **component** requirements.*

Requirements Management and Requirements Development



Compliments of
Kasse Initiatives, LLC

GOALS



SG 1 Develop Customer Requirements

Stakeholder needs, expectations, constraints, and interfaces are collected and translated into customer requirements.

SG 2 Develop Product Requirements

Customer requirements are refined and elaborated to develop product and product component requirements.

SG 3 Analyze and Validate Requirements

The requirements are analyzed and validated, and a definition of required functionality is developed.

Importance of Requirements Development

Present complete **clear validated** requirements **understood by all parties**

Establish solid **foundation** for downstream activities

Benefits of Proper Requirements Development

Development team and customer share the same vision of what is to be developed, tested and supported

Requirements are easily traceable to/from downstream work products

Acceptance by customer of downstream products is easy & swift

Low risk of increased costs to meet customer needs and expectations

Terminology

Allocated Requirement - Requirement that levies all or part of the performance and functionality of a higher level requirement on a lower level architectural element or design component.

Derived Requirement - Requirements that are not explicitly stated in the customer requirements, but are inferred (1) from contextual requirements (e.g., applicable standards, laws, policies, common practices, and management decisions), or (2) from requirements needed to specify a product component. Derived requirements can also arise during analysis and design of components of the product or system. (See also "product requirements.")

Terminology II

Customer Requirement - The result of eliciting, consolidating, and resolving conflicts among the needs, expectations, constraints, and interfaces of the product's relevant stakeholders in a way that is acceptable to the customer. (See also "customer.")

Product Requirement - A refinement of the customer requirements into the developers' language, making implicit requirements into explicit derived requirements. (See also "derived requirements" and "product component requirements.") The developer uses the product requirements to guide the design and building of the product.

Product Component Requirements - A complete specification of a product component, including fit, form, function, performance, and any other requirement.

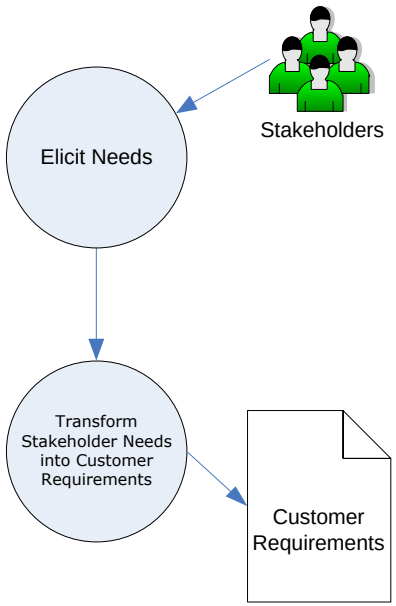
SG 1 Develop Customer Requirements

SP 1.1 Elicit Needs

Elicit stakeholder needs, expectations, constraints, and interfaces for all phases of the product lifecycle.

SP 1.2 Transform Stakeholder Needs into Customer Requirements

Transform stakeholder needs, expectations, constraints, and interfaces into customer requirements.



SG 2 Develop Product Requirements

SP 2.1 Establish Product and Product Component Requirements

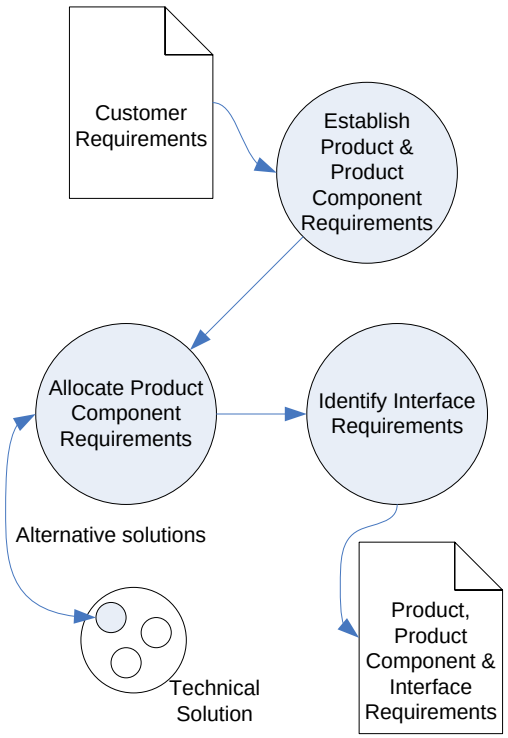
Establish and maintain product and product component requirements, which are based on the customer requirements.

SP 2.2 Allocate Product Component Requirements

Allocate the requirements for each product component.

SP 2.3 Identify Interface Requirements

Identify interface requirements.



SG 3 Analyze and Validate Requirements

SP 3.1 Establish Operational Concepts and Scenarios

Establish and maintain operational concepts and associated scenarios.

SP 3.2 Establish a Definition of Required Functionality

Establish and maintain a definition of required functionality.

SP 3.3 Analyze Requirements

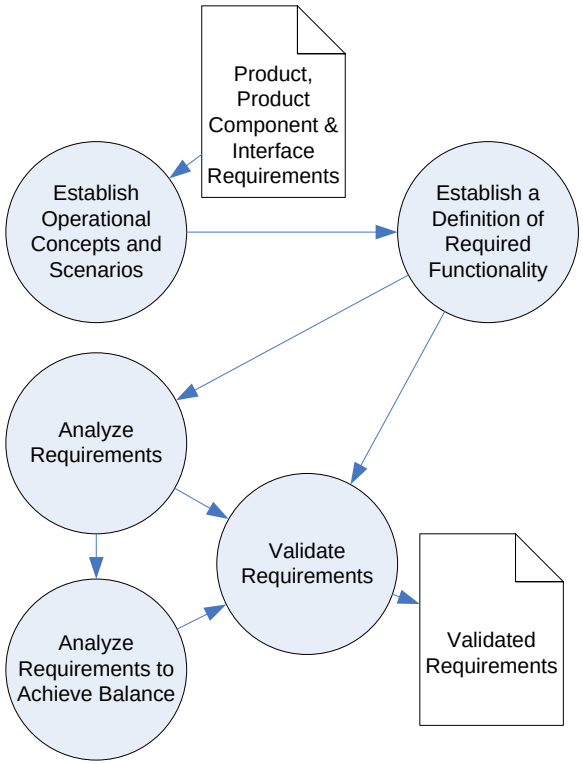
Analyze requirements to ensure that they are necessary and sufficient.

SP 3.4 Analyze Requirements to Achieve Balance

Analyze requirements to balance stakeholder needs and constraints.

SP 3.5 Validate Requirements

Validate requirements to ensure the resulting product will perform as intended in the user's environment.



How Requirements Development interacts with other Process Areas

Who does RD depend upon?

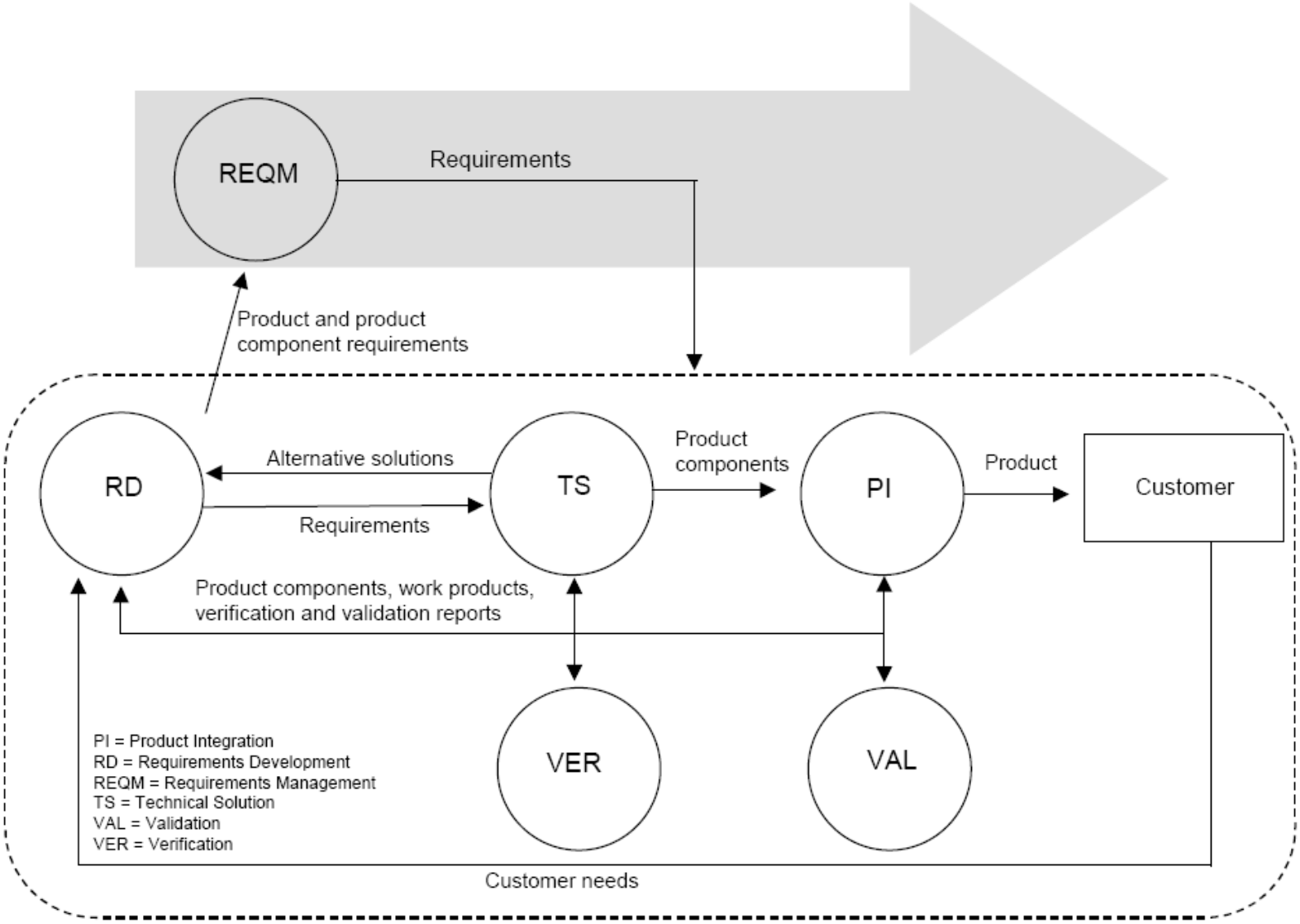
- Requirements Management (ML2:REQM) for managing requirements
- Technical Solution (ML3:TS) for development of alternative solutions and identification of product components
- Risk Management (ML3:RSKM) for identification and management of requirements risks

Who depends on RD?

- Requirements Management (ML2:REQM) takes requirements from RD
- Product Integration (ML3:PI) takes interface requirements
- Verification & Validation (ML3: VER & VAL)

Where Requirements Development stands in the model?

- **Maturity level 3**
- **Enginee ng** proce area



CMMI V2.0 [Next Generation CMMI]

REQUIREMENTS DEVELOPMENT AND MANAGEMENT (RDM)

- Combined REQM (ML2) and RD (ML3)
- New approach to ML and indicators
- Adapted to Agile organizations
- Process areas >>> **Practice** areas

CMMI V2.0 REQUIREMENTS DEVELOPMENT AND MANAGEMENT (RDM)

Level 1

- RDM 1.1 Record requirements

Level 2

- RDM 2.1 Elicit stakeholder needs, expectations, constraints, and interfaces or connections. (RD SP 1.1)
- RDM 2.2 Transform stakeholder needs, expectations, constraints, and interfaces or connections into prioritized customer requirements. (RD SP 1,1, 3.2)
- RDM 2.3 Develop an understanding with the requirements providers on the meaning of the requirements. (REQM SP 1.1.)
- RDM 2.4 Obtain commitment from project participants that they can implement the requirements. (REQM SP 1.2)
- RDM 2.5 Develop, record, and maintain bidirectional traceability among requirements and activities or work products. (REQM SP 1.4)
- RDM 2.6 Ensure that plans and activities or work products remain consistent with requirements. (REQM SP 1.4)

Level 3

- RDM 3.1 Develop and keep requirements updated for the solution and its components. (RD SP 2.1)
- RDM 3.2 Develop operational concepts and scenarios. (RD SP 3.1, 3.2)
- RDM 3.3 Allocate the requirements to be implemented. (RD SP2.2)
- RDM 3.4 Identify, develop, and keep updated interface or connection requirements. (RD SP 2.3)
- RDM 3.5 Ensure that requirements are necessary and sufficient. (RD SP 3.3)
- RDM 3.6 Balance stakeholder needs and constraints. (RD SP 3.4)
- RDM 3.7 Validate requirements to ensure the resulting solution will perform as intended in the target environment. (RD SP 3.5)

PP: Project Planning

*The purpose of Project Planning (PP) is to **establish and maintain plans** that define project activities.*

SG1: Establish Estimates
Estimates of project planning parameters are established and maintained.

SG2: Develop a Project Plan
A project plan is established and maintained as the basis for managing the project.

SG3: Obtain Commitment to the Plan
Commitments to the project plan are established and maintained.

When Project Planning Is Not Done Well…

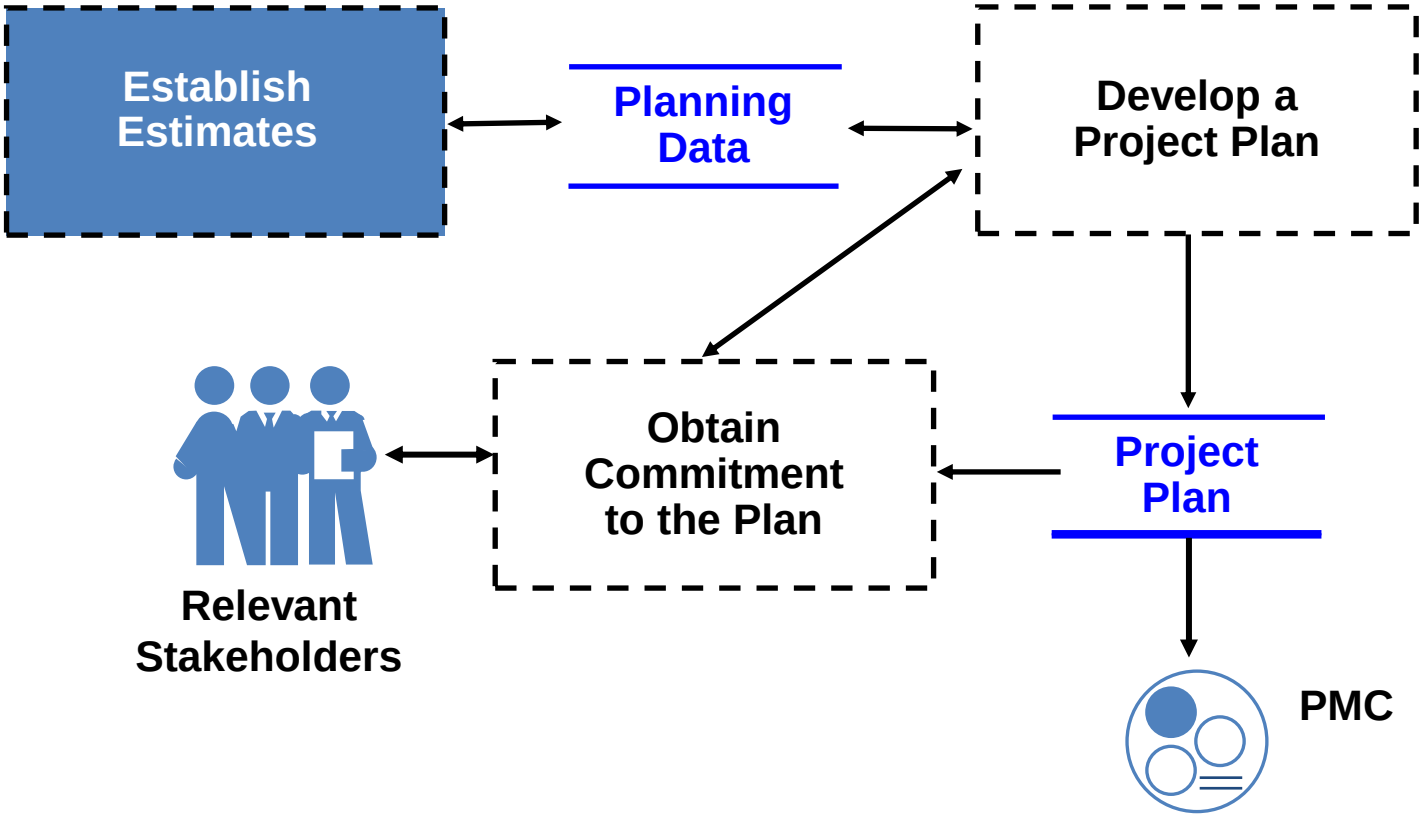
Estimates of project attributes are inaccurate.

It is difficult to **identify deviations** from poorly documented plans.

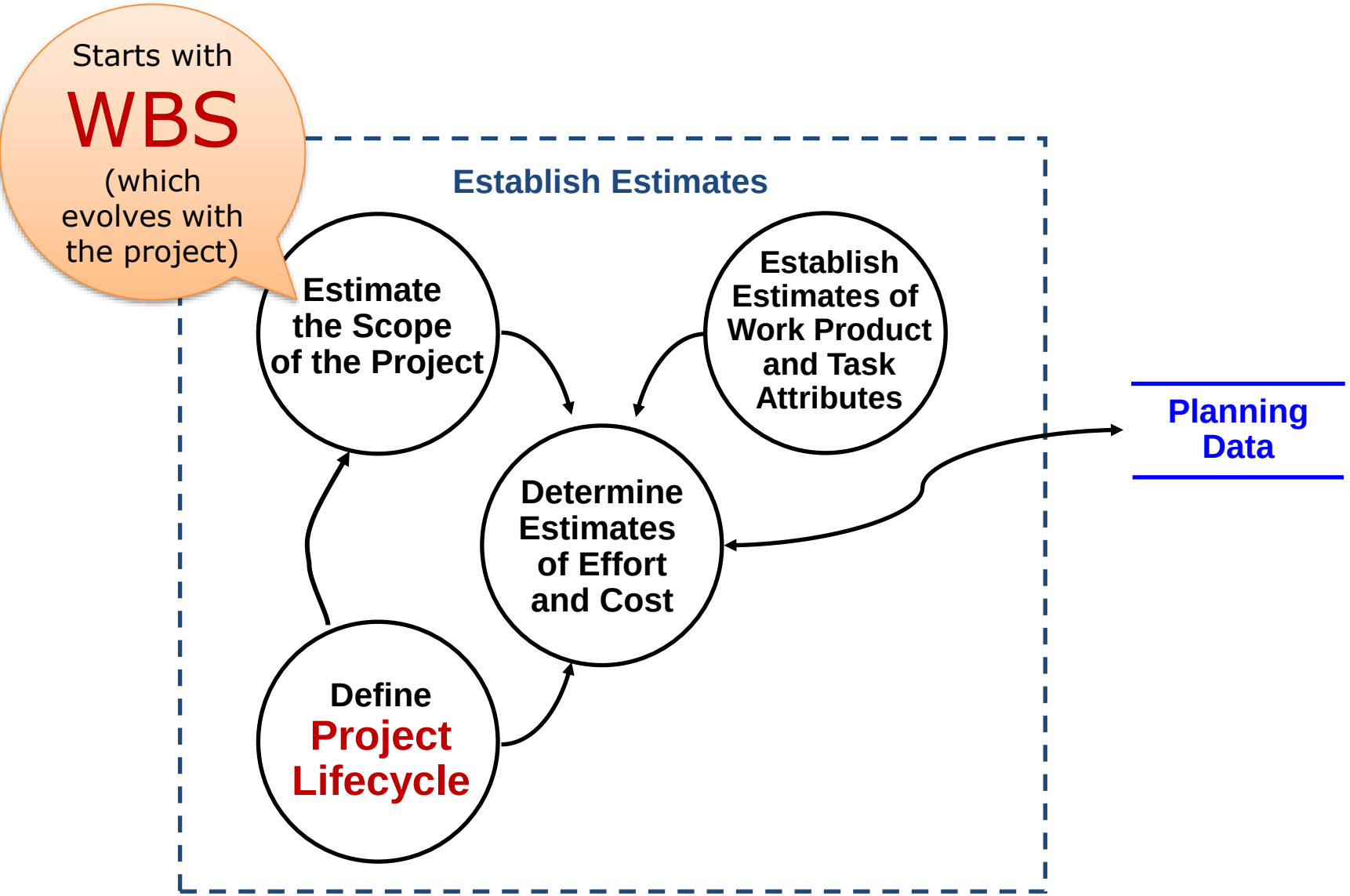
Resources are not available/applied when needed.

Future projects cannot learn from completed projects because there are **no lessons learned**.

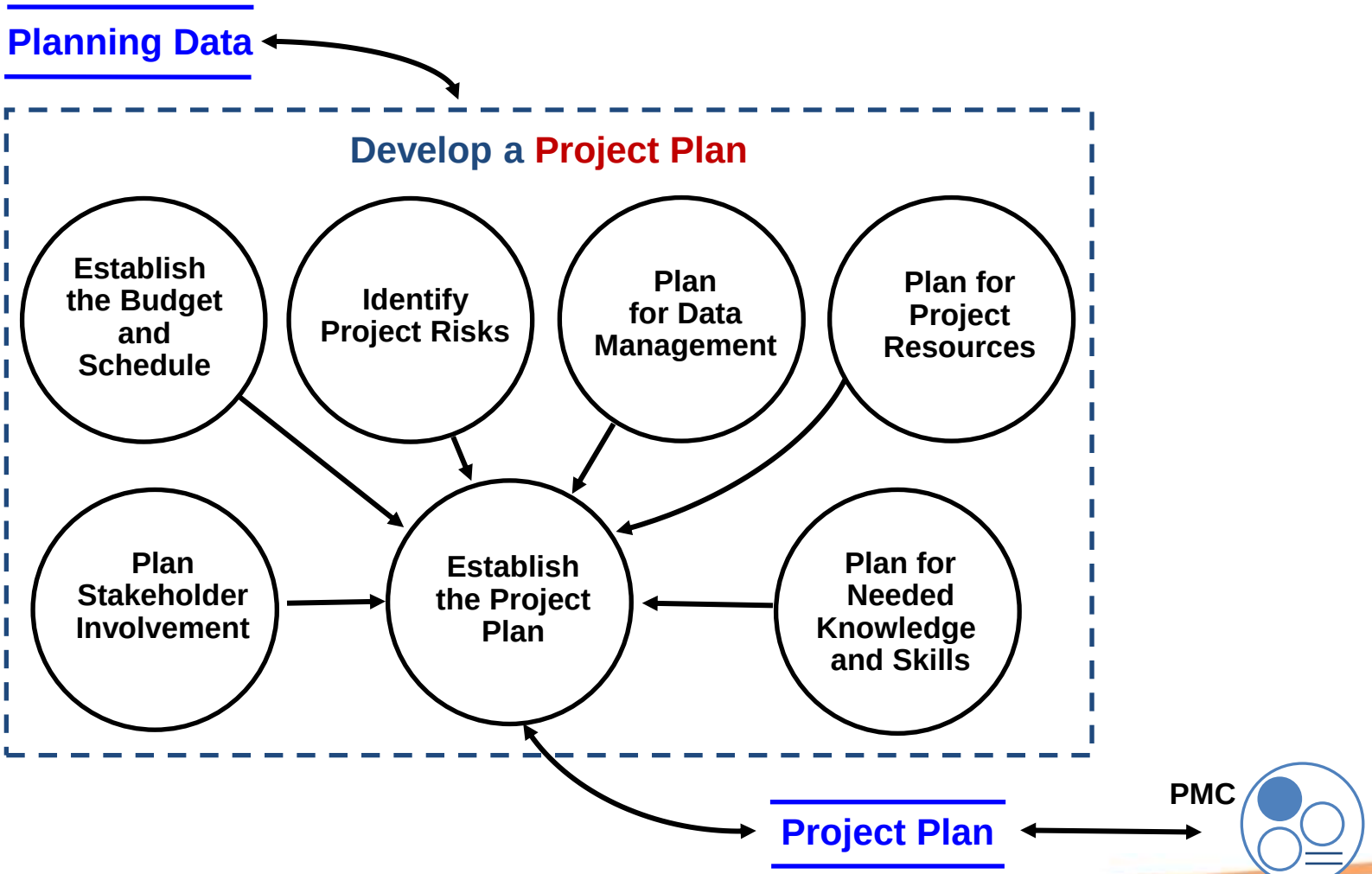
Project Planning (PP)



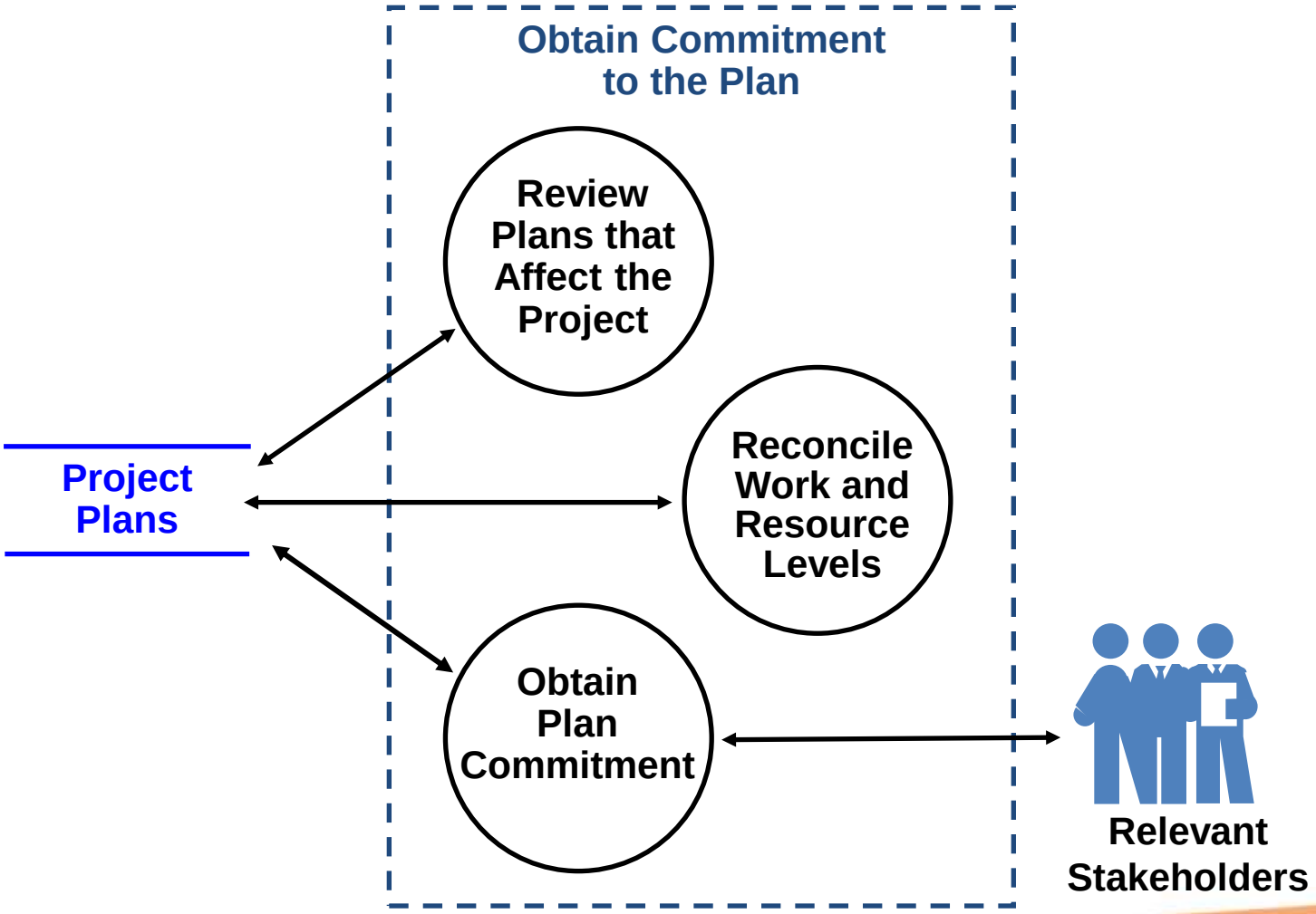
PP-1


Starts with
WBS
(which evolves with the project)
Establish Estimates
Estimate the Scope of the Project
Establish Estimates of Work Product and Task Attributes
Determine Estimates of Effort and Cost
Define Project Lifecycle
Planning Data

PP-2


Planning Data
Develop a Project Plan
Establish the Budget and Schedule
Identify Project Risks
Plan for Data Management
Plan for Project Resources
Plan Stakeholder Involvement
Establish the Project Plan
Plan for Needed Knowledge and Skills
Project Plan
PMC

PP-3


Obtain Commitment to the Plan
Review Plans that Affect the Project
Reconcile Work and Resource Levels
Obtain Plan Commitment
Project Plans
Relevant Stakeholders

SW Project Plan – example - 1 Gantt Chart

Figure 1. Gantt Chart Example: Planning a custom-written computer project

Task	Earliest start	Length	Type	Dependent on...
A. High level analysis	Week 0	1 week	Sequential	
B. Selection of hardware platform	Week 1	1 day	Sequential	A
C. Installation and commissioning of hardware	Week 1.2	2 weeks	Parallel	B
D. Detailed analysis of core modules	Week 1	2 weeks	Sequential	A
E. Detailed analysis of supporting modules	Week 3	2 weeks	Sequential	D
F. Programming of core modules	Week 3	2 weeks	Sequential	D
G. Programming of supporting modules	Week 5	3 weeks	Sequential	E
H. Quality assurance of core modules	Week 5	1 week	Sequential	F
I. Quality assurance of supporting modules	Week 8	1 week	Sequential	G
J. Core module training	Week 6	1 day	Parallel	C,H
K. Development and QA of accounting reporting	Week 5	1 week	Parallel	E
L. Development and QA of management reporting	Week 5	1 week	Parallel	E
M. Development of Management	Week 6	1 week	Sequential	I

Internet | Protected M

Step 1. List all activities in the plan

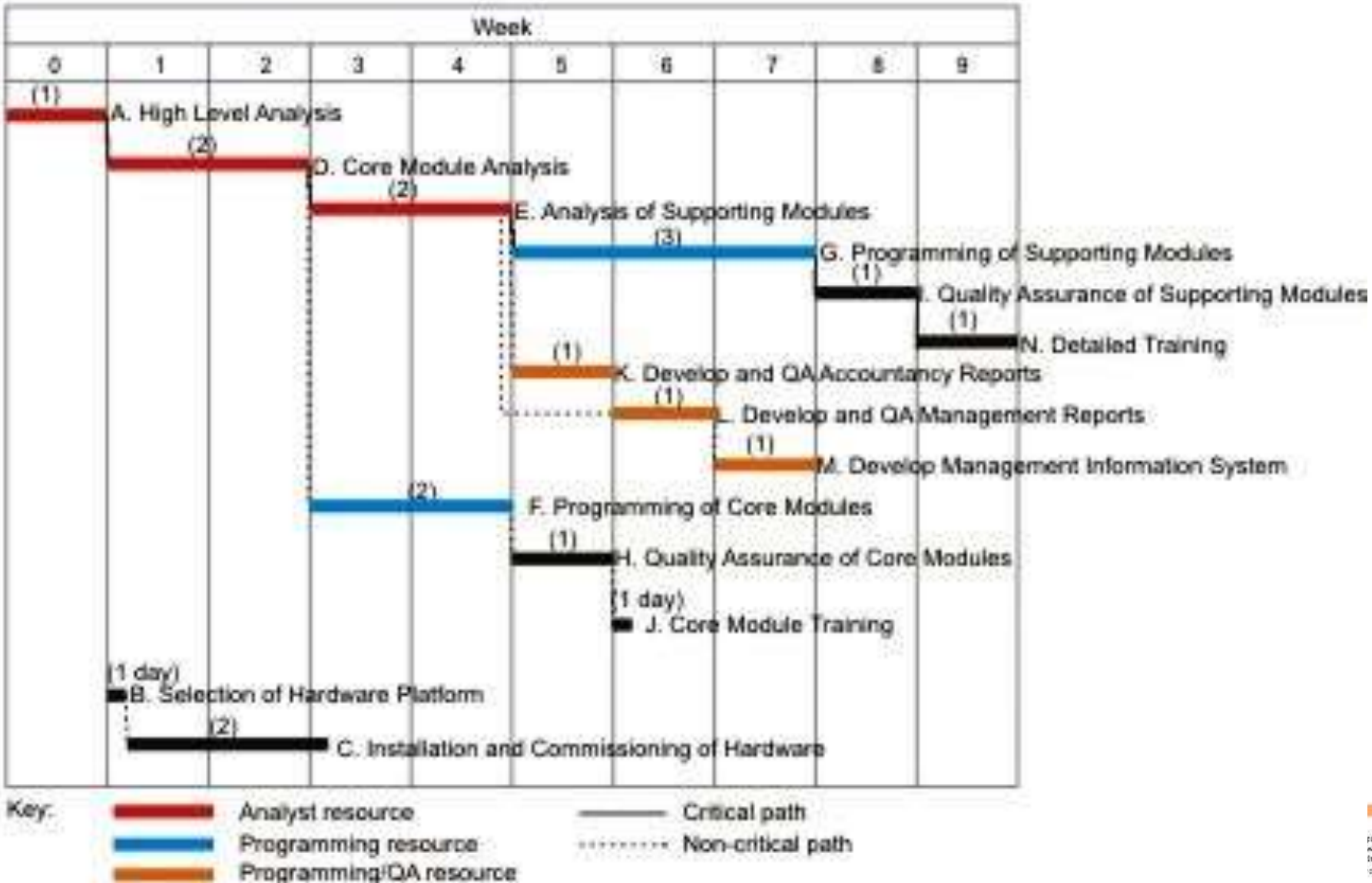
Sequential and parallel activities

Courtesy to www.mindtools.com

Courtesy to www.mindtools.com

SW Project Plan – example - 2
Gantt Chart and Critical Path

Figure 3: Critical Path Analysis: Activities Scheduled on a Gantt Chart



Sampling the Generic Practices

GP 2.2: Plan the Process

Establish and maintain the plan for performing the project planning process.

Or …. PLAN THE PLAN !!!

Elaboration for Project Planning

Refer to Table 6.2 in Generic Goals and Practices in Part Two for more information about the relationship between generic practice 2.2 and the Project Planning process area.

Example: A typical Project Plan includes

- Resources
- Budget
- Schedule (Milestones)
- Stakeholders
- Commitments – dependencies, deliverables
- Data Plan
- Knowledge and skills - Training
- Risks

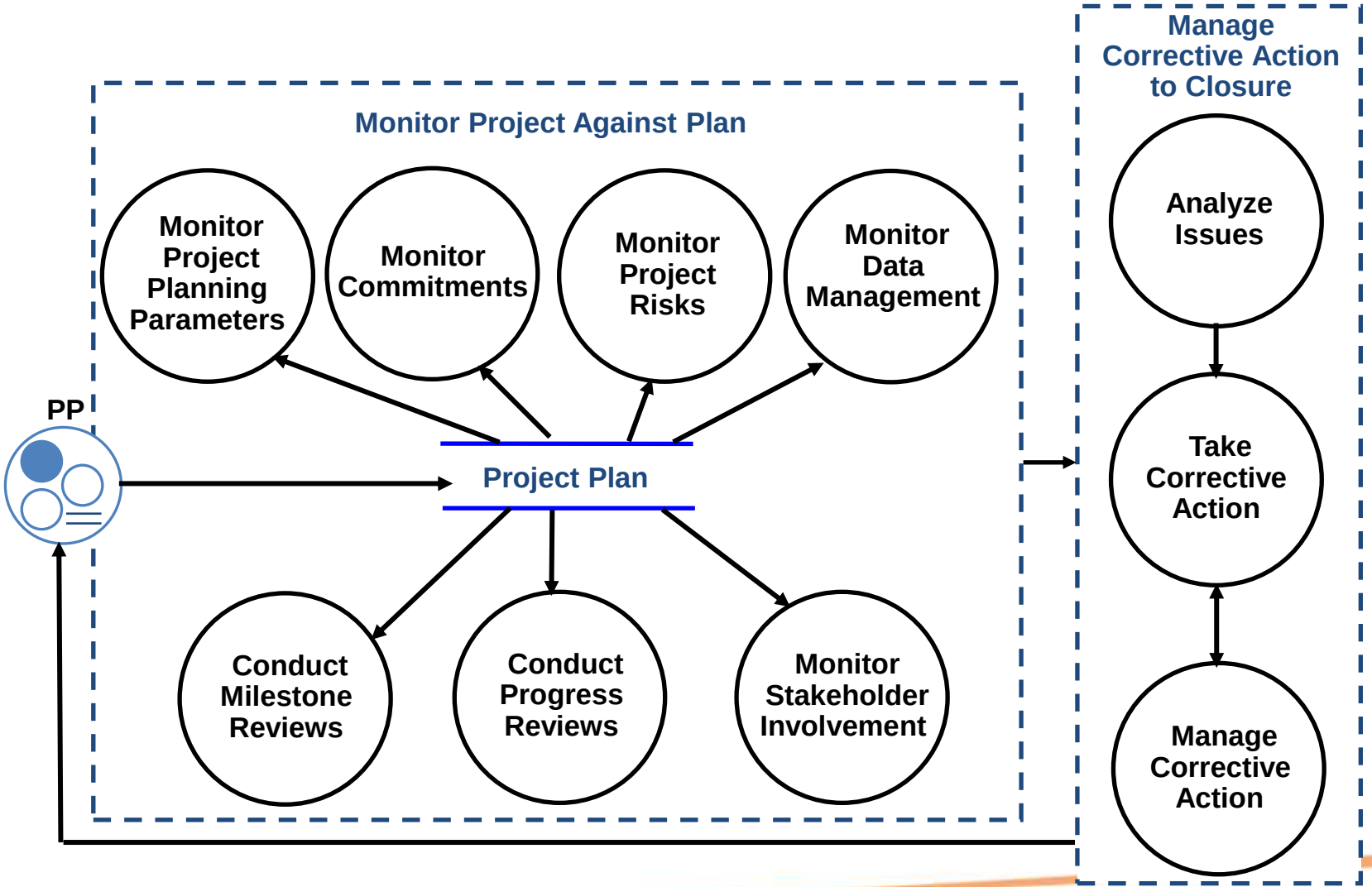
PMC: Project Monitoring and Control

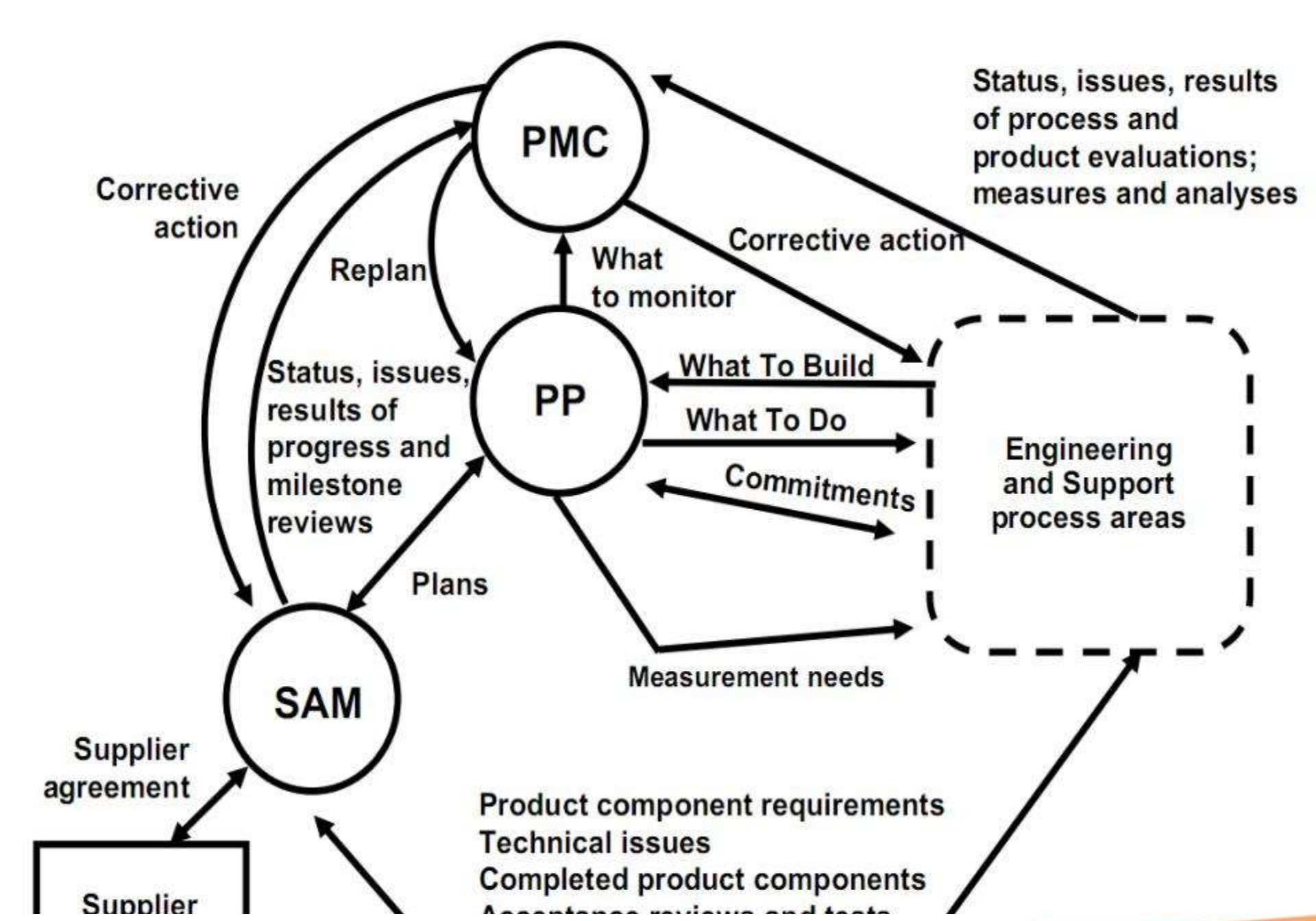
*The purpose of Project Monitoring and Control (PMC) is to provide an understanding of the **project's progress** so that appropriate **corrective actions** can be taken when the project's performance deviates significantly from the plan.*

SG 1: Monitor Project Against Plan
Actual performance and progress of the project are monitored against the project plan.

SG 2: Manage Corrective Action to Closure
Corrective actions are managed to closure when the project's performance or results deviate significantly from the plan.

Project Monitoring and Control (PMC)

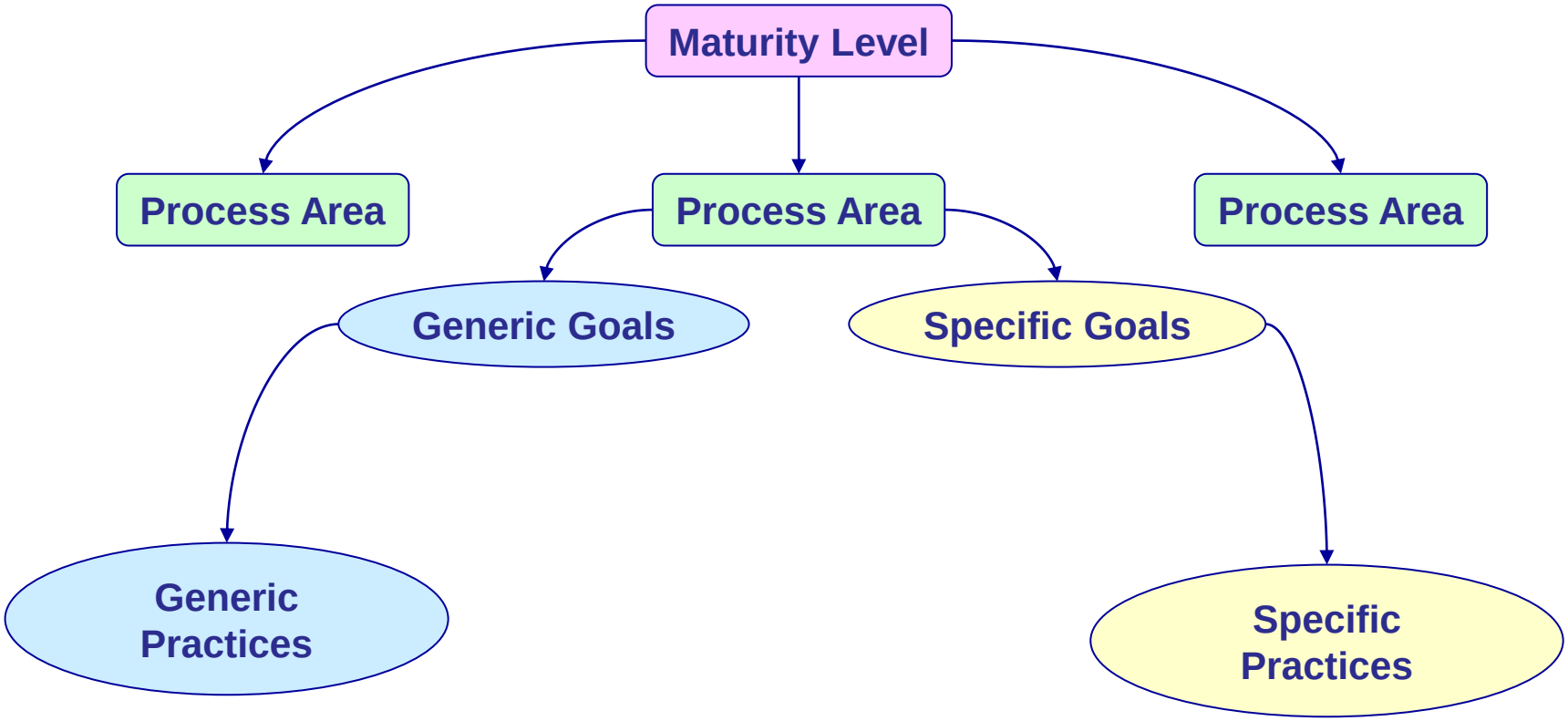



PMC
Status, issues, results of process and product evaluations; measures and analyses
Corrective action
Replan
What to monitor
Corrective action
PP
What To Build
What To Do
Commitments
Status, issues, results of progress and milestone reviews
Engineering and Support process areas
Plans
SAM
Measurement needs
Supplier agreement
Product component requirements
Technical issues
Completed product components
Supplier

Нива на зрялост, генерични цели и генерични практики

Maturity Level 2 & 3 – Generic goals and practices

Structure of the CMMI Staged Representation



GG (Generic goals) = Institutionalization

GG2 (ML2): Institutionalize a Managed Process

The process is institutionalized as a managed process.

- A managed process is a performed process that is planned and executed in accordance with policy; employs skilled people having adequate resources to produce controlled outputs; involves relevant stakeholders; is monitored, controlled, and reviewed; and is evaluated for adherence to its process description.
- Management of the process is concerned with institutionalization and the achievement of specific objectives established for the process, such as cost, schedule, and quality objectives.

ML2 GG&GPs

GG2: Institutionalize a Managed Process

What should be applied **to all PAs** (from ML2 and up):

- GP2.1: Establish an Organizational Policy
- GP2.2: Plan the Process
- GP2.3: Provide Resources
- GP2.4: Assign Responsibility
- GP2.5: Train People
- GP2.6: Control Work Products
- GP2.7: Identify and Involve Relevant Stakeholders
- GP2.8: Monitor and Control the Process
- GP2.9: Objectively Evaluate Adherence
- GP2.10: Review Status with Higher Level Management

Maturity levels: generic and specific practices

Maturity Level 2

- Requirements management
- Project planning
- Project monitoring and control
- Supplier agreement management
- Measurement and analysis
- Process and product quality assurance
- Configuration management

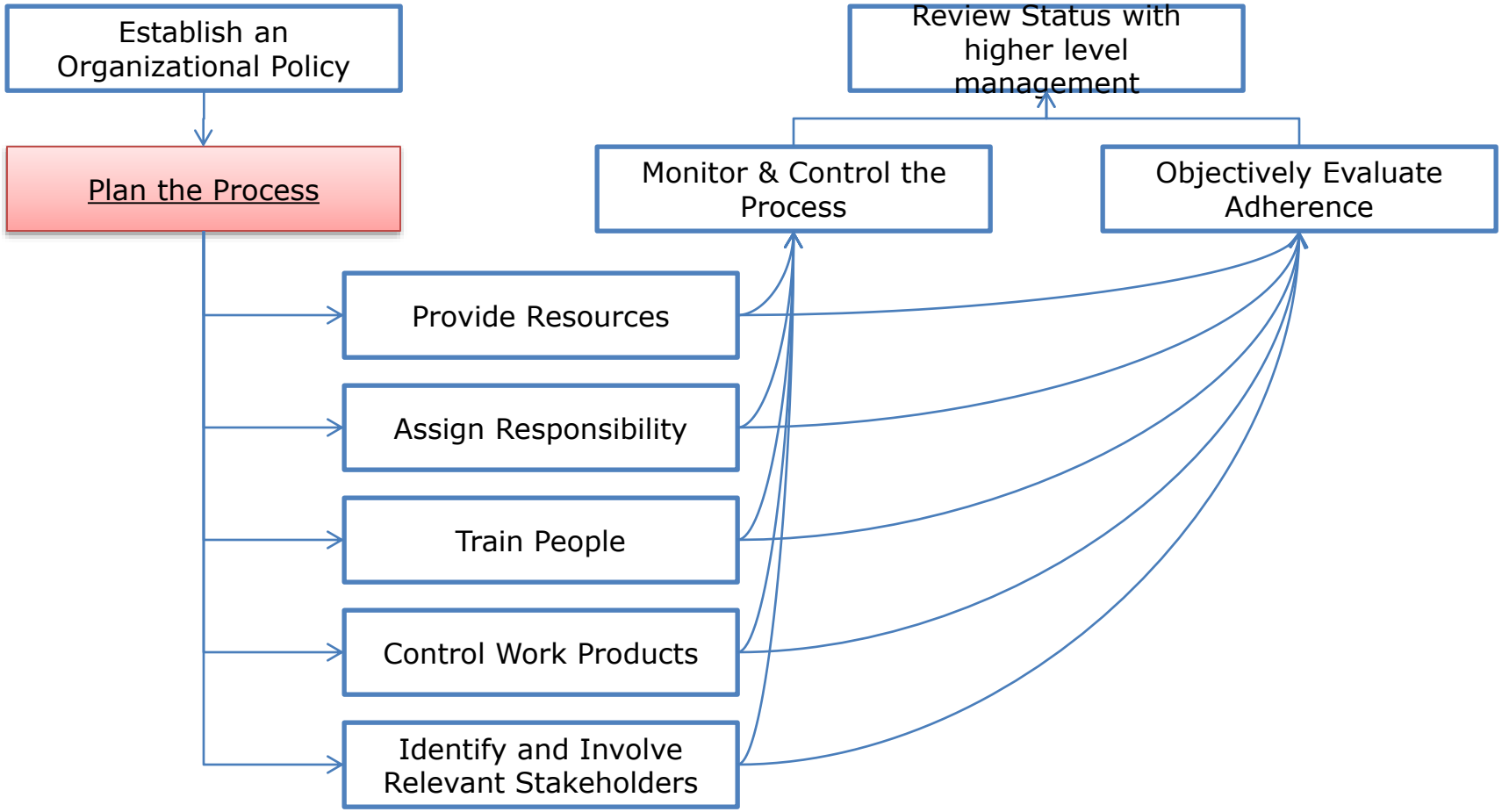
Maturity Level 3

- Requirements development
- Technical solution
- Product integration
- Verification
- Validation
- Organizational process focus
- Organizational process definition + IPPD
- Organizational training
- Integrated project management + IPPD
- Risk management
- Decision analysis and resolution

GP 2.1 Establish organizational policy
GP 2.2 Plan the process
GP 2.3 Provide resources
GP 2.4 Assign responsibility
GP 2.5 Train people
GP 2.6 Control Work Products (Manage configuration)
GP 2.7 Identify and involve relevant stakeholders
GP 2.8 Monitor and control the process
GP 2.9 Objectively evaluate adherence
GP 2.10 Review status with higher level management

GP 3.1 Establish a defined process
GP 3.2 Collect improvement information

How PAs relate to Generic Practices?



Source: Kiril Karaatanasov, ESI Center Bulgaria

Next: Supporting PAs ML2:

- Requirements Management
- Project Planning
- Project Monitoring & Control
- **Process and Product Quality Assurance**
- **Measurement & Analysis**
- **Configuration Management**
- Supplier Agreement Management

What PPQA provides?

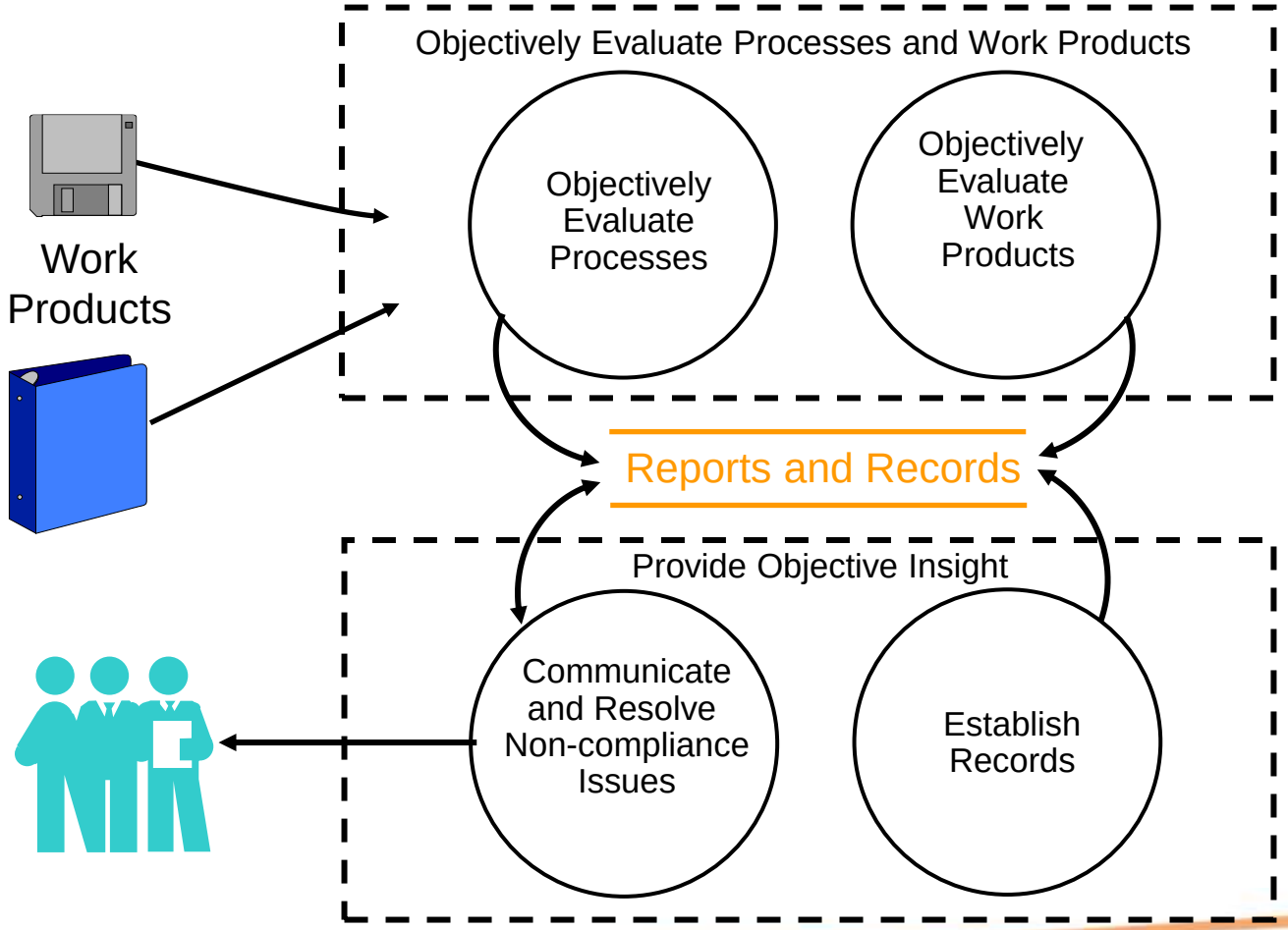
Management knows if **process assets** are being used

Failures to follow process that may **endanger projects become visible early on**

Problems with **process definitions** are uncovered and addressed

Process descriptions are **followed**

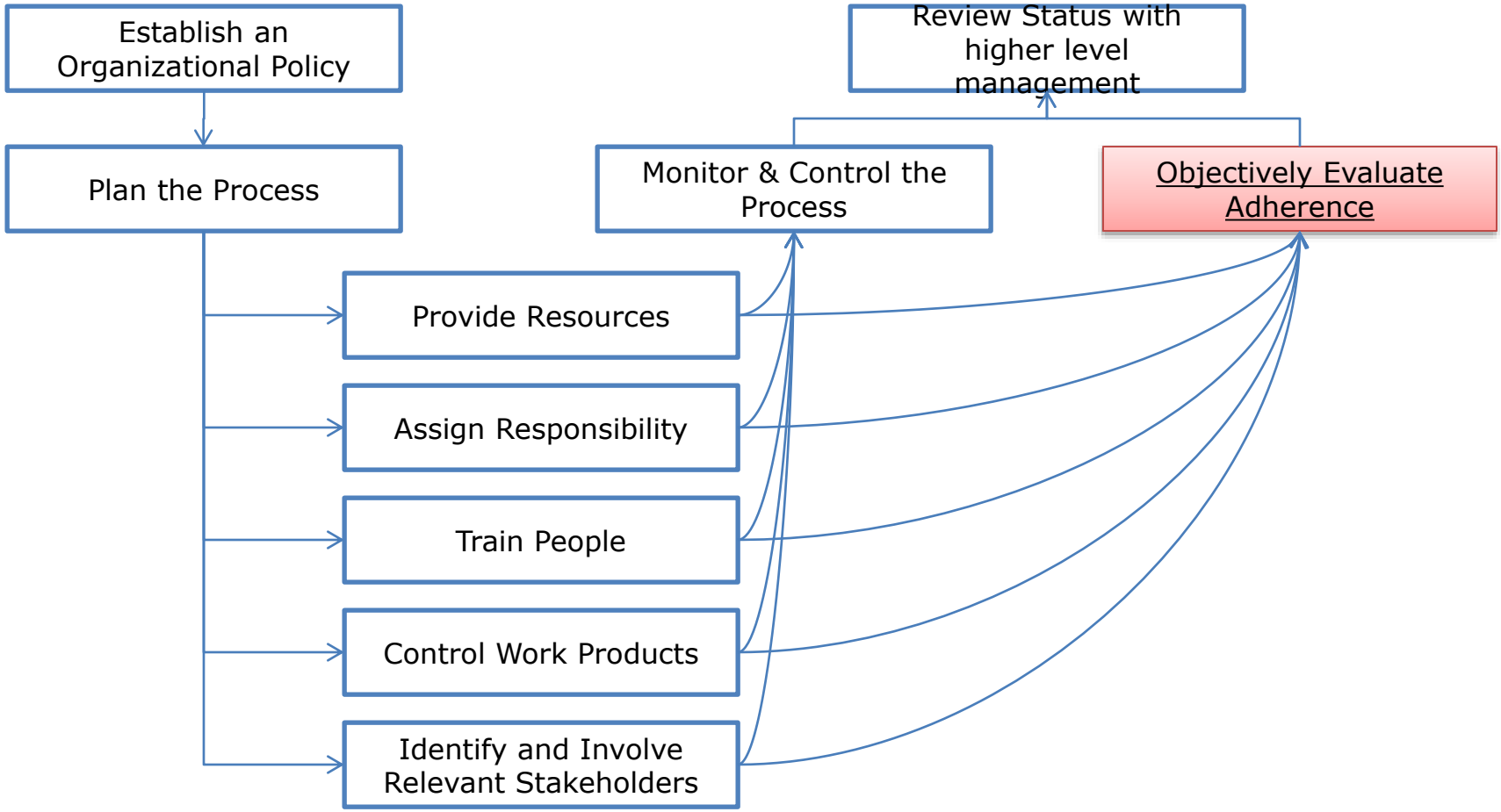
Process and Product Quality Assurance - Context



PPQA Practices translated:

- Are QA evaluations performed on processes/workproducts according to predefined criteria?
- Performed processes adhere to the standards, process descriptions and procedures?
- Non-compliance identified during the QA evaluations of processes/work products?
- Lessons learned collected?
- Non-compliances resolved within the project/escalated?
- Relevant stakeholders aware of the results of the QA evaluations?
- Management reviews on non-compliances on periodic basis?
- Non-compliances tracked until closure?
- QA activities documented in sufficient detail?
- QA status and results known?

How PPQA relates to Generic Practices?



Source: Kiril Karaatanasov, ESI Center Bulgaria

Configuration Management (CM)

The purpose of Configuration Management (CM) is to establish and maintain the integrity of work products using configuration identification, configuration control, configuration status accounting, and configuration audits.

SG 1: Establish Baselines
Baselines of identified work products are established.

SG 2: Track and Control Changes
Changes to the work products under configuration management are tracked and controlled.

SG 3: Establish Integrity
Integrity of baselines is established and maintained.

What does CM Provide?

State of components is known and there is confidence what and when can be released

When needed baselines can be recovered

Changes from baseline are identifiable

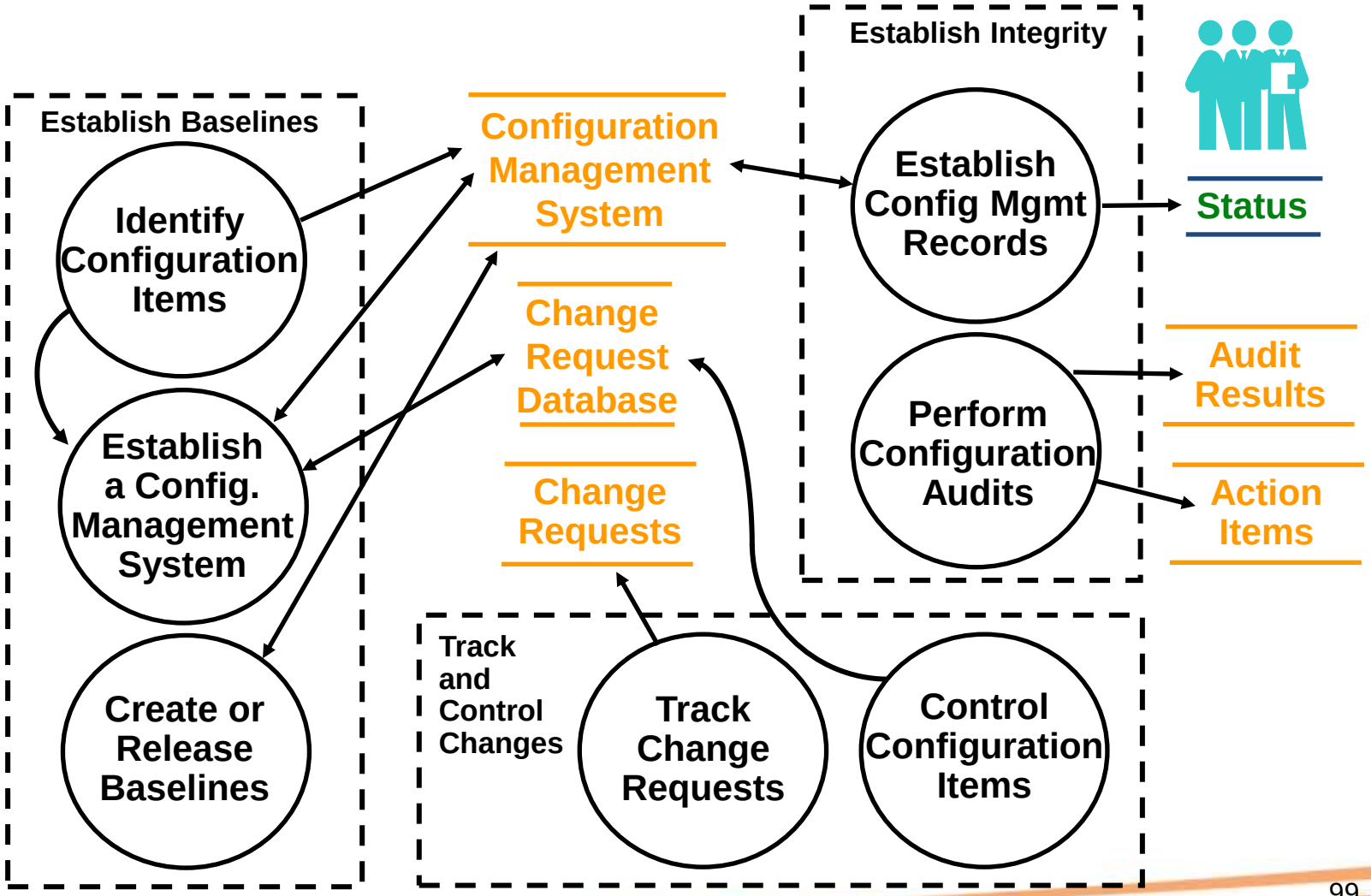
Past product releases can be rebuilt

Reasons for changes to plans are clear

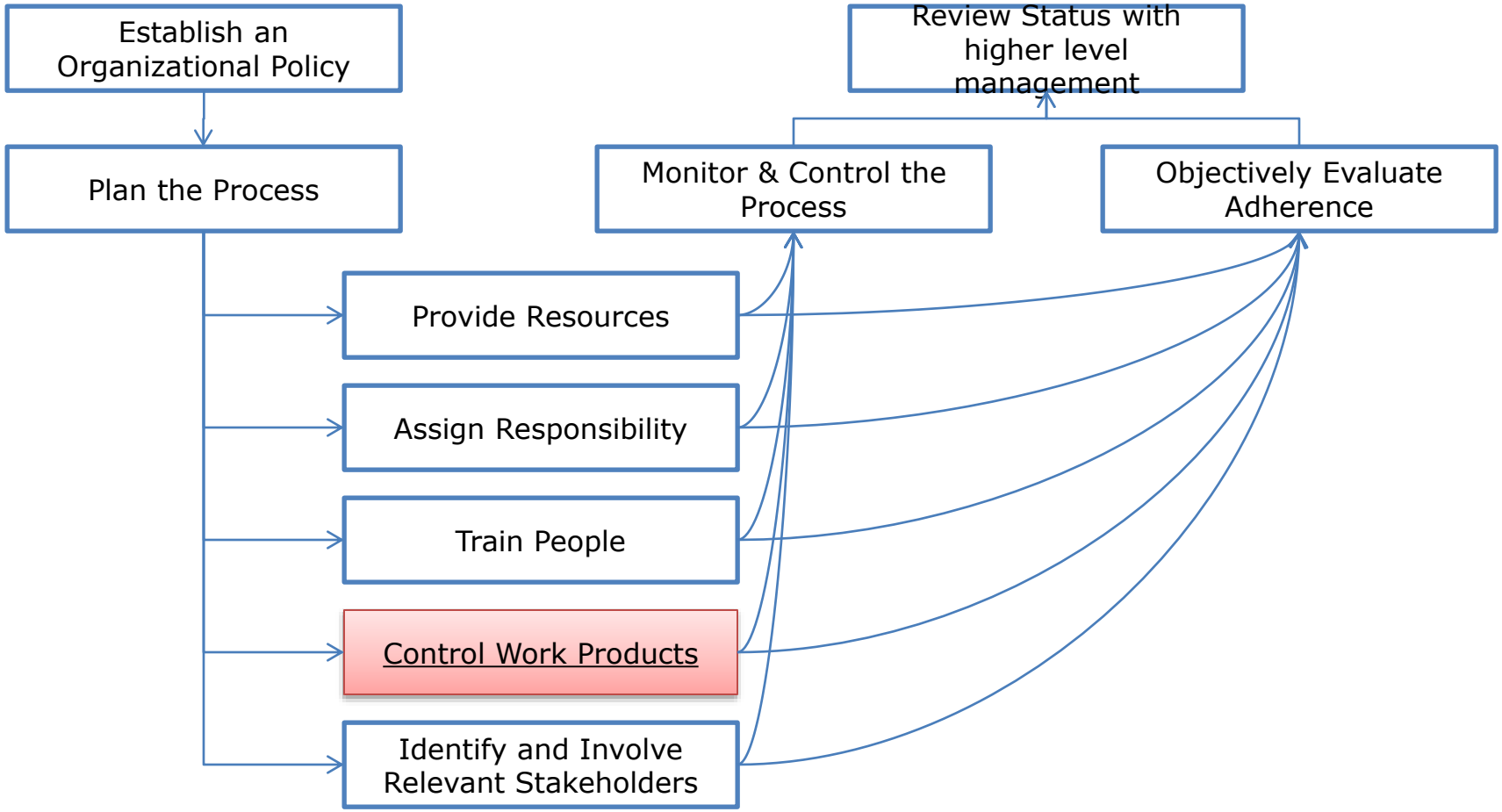
Baseline

- A set of specifications or work products that has been formally reviewed and agreed on, which thereafter serves as the basis for further development, and which can be changed only through change control procedures. (See also “configuration baseline” and “product baseline.”)

Configuration Management - Context



How CM relates to Generic Practices?



Source: Kiril Karaatanasov, ESI Center Bulgaria

Measurement & Analysis

The purpose of Measurement and Analysis (MA) is

to **develop and sustain**

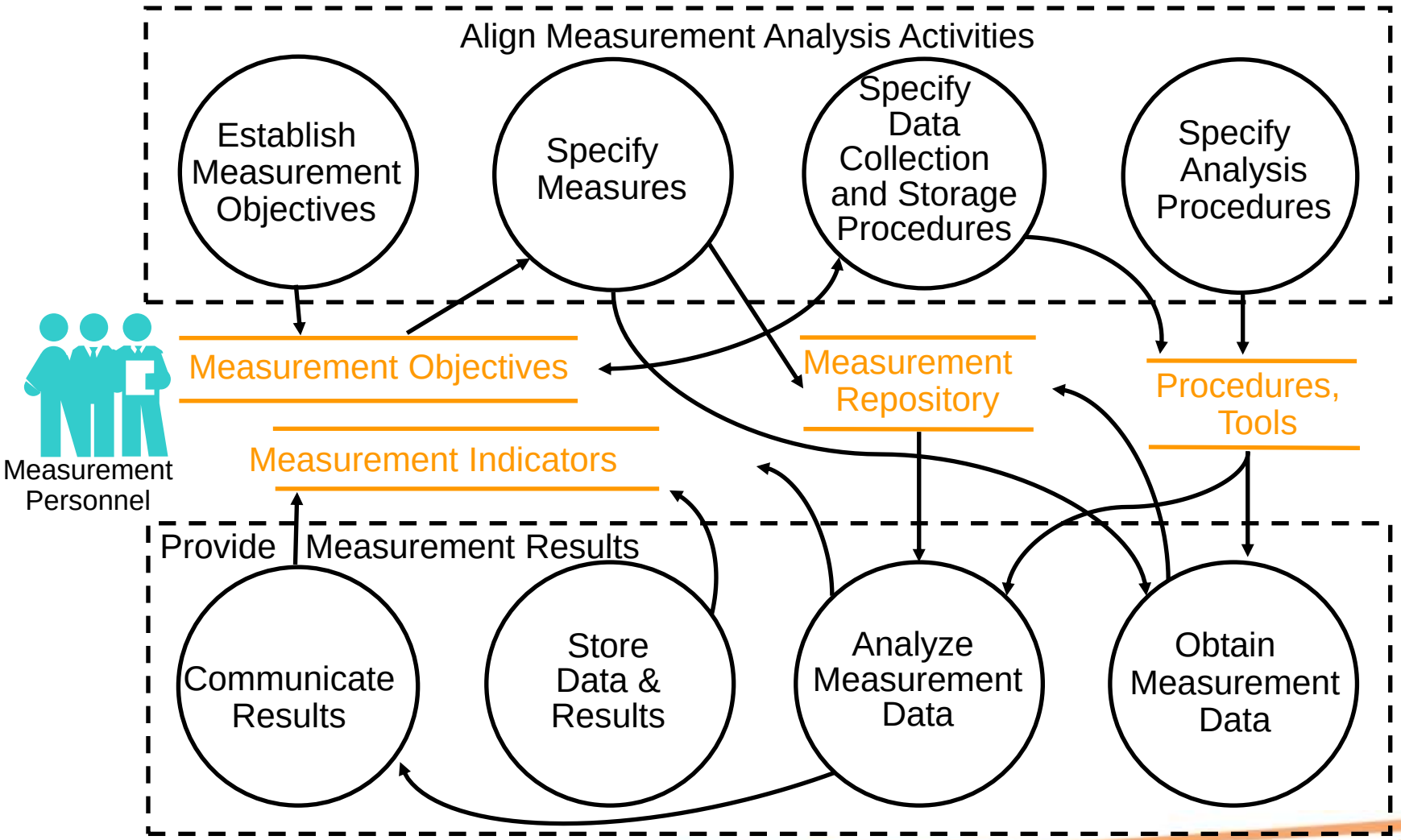
a measurement capability

that is used to support

management

information needs.

Measurement & Analysis - Context

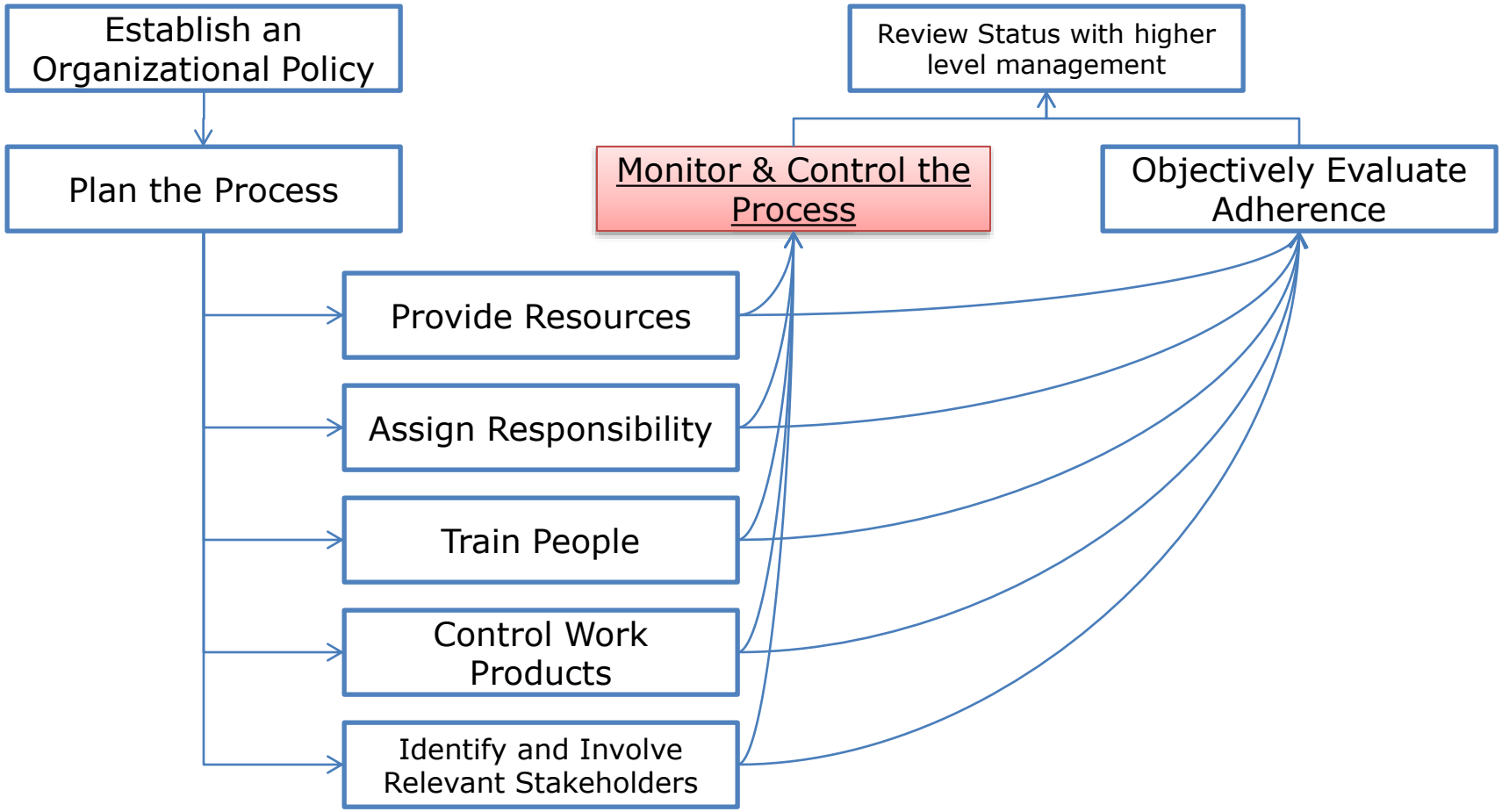


MA – Metrics example 1

REQM (Requirements Management)

1. **Requirements volatility (percentage of requirements changes)**
2. Number of requirements by type or status (defined, reviewed, approved, and implemented)
3. Cumulative number of changes to the allocated requirements, including total number of changes proposed, open, approved, and incorporated into the system baseline
4. Number of changes requests per month, compared to the original number of requirements for the project
5. Number of time spent, effort spent, cost of implementing change requests
6. **Number and size of change requests after the Requirements phase is finished**
7. Cost of implementing a change request
8. Number of change requests versus the total number of change requests during the life of the project
9. Number of change requests accepted but not implemented
10. **Number of requirements (changes and additions to the baseline)**

How MA relates to Generic Practices?



Source: Kiril Karaatanasov, ESI Center Bulgaria