ESI European Software Institute Center Eastern Europe

Partner of: Software Engineering Institute | Carnegie Mellon

SEMP Software Engineering Management Program

МУК: Модели за управление на качеството. Курс на база на CMMI

SPI models. CMMI based course

PART 4: CMMI Maturity Level 3

SEMP (Software Engineering Management Program) course,
МУК-2009-2015, FMI/SU

Dr. George Sharkov, Ivaylo Georgiev, Krassimir Baylov
ESI Center Eastern Europe
gesha@esicenter.bg

www.esicenter.bg

SEMP: SOFTWARE ENGINEERING MANAGEMENT PROGRAM

The course is developed (and compiled) jointly by ESI Center (Eastern Europe) and CMU (Carnegie Mellon University) from the main lines and materials for SEMP, in partnership with SEI/CMU.

It introduces students to process improvement as a main factor for the quality of products and services.

Based on process-oriented models - CMMI, the "industrial" standard developed by SEI/CMU, project management (PMI/PM BOK), personal/team management (PSP/TSP BOK), strategic planning (Balanced ScoreCards), information security.

Augmented by modern methods and techniques – Agile CMMI, Six Sigma, etc. Mapping between main industrial models and standards. Implementation. Models for quality improvement in small settings and SMEs. Business aspects – cost of quality, what is "the right model for my company", why invest in PI, what is the return, who can help.

Информация, източници:

www.esicenter.bg >> general info and in “Resources”

links to CMMI models
http://cmmiinstitute.com/cmmi-solutions/
http://www.sei.cmu.edu/cmmi/tools/index.cfm

CMMI –DEV v 1.3 model (CMMI Institute, and SEI, Carnegie Mellon University)

http://cmmiinstitute.com/resource/cmmi-for-development-version-1-3/

*www.**sei**.cmu.edu/reports/10tr033.pdf*

General
www.sei.cmu.edu
www.cmmiinstitute.com

Съдържание

1	Увод в управление на качеството. Компоненти и цена на качеството. Процеси. Преглед на моделите за управление на качеството и подобряване на процесите. Методи за оценка на зрелостта на ИТ-интензивни и софтуерни организации. Стратегически карти/Балансирана система от показатели (balanced ScoreCards).
2	Модел CMMI (ver 1.2). История, внедряващи организации. Обща структура. Процесни области. Генерични и специфични цели и практики. Презентации – Maturity/Capability нива на Continuous и Staged representations. Категории процесни области: Process Management, Project Management, Engineering, Support.
3	Процесни области от ниво 2 на CMMI. Детайлно представяне на: REQM – Requirements Management PP – Project Planning MA – Measurement and Analysis PPQA – Process and Product Quality Assurance CM – Configuration Management PMC – Project Monitoring and Control Преглед на:SAM-Supplier Agreement Management
4	**Процесни области от ниво 3 на CMMI. Детайлно представяне на:** **RD – Requirements Development** **VAL - Validation** **VER - Verification** **RSKM - Risk Management** **TS - Technical Solution** **Преглед на: DAR - Decision Analysis and Resolution , IPM - Integrated Project Management , OPD - Organizational Process Definition , OPF - Organizational Process Focus, OT - Organizational Training , PI - Product Integration** **Преглед на Maturity Level 4 и 5.** **Обобщение на връзките между процесните области: Tying all together**
5	Внедряване на програма за подобряване на процесите на база CMMI. Адаптирани подходи – Agile CMMI, CMMI/ISO. Нови модели CMMI – CMMI for Services, CMMI for Acquisition. Оценка (SCAMPI), роли.
6	Подобряване на процесите в малки фирми – IT Mark. Компненти на зрелостта – бизнес, организация/процеси, информационна сигурност. Оценка на нивото и план за подобрения.

Part 4: Maturity Level 3

Процесни области от ниво 3 на CMMI. Представяне на:

RD – Requirements Development

VAL - Validation

VER - Verification

RSKM - Risk Management

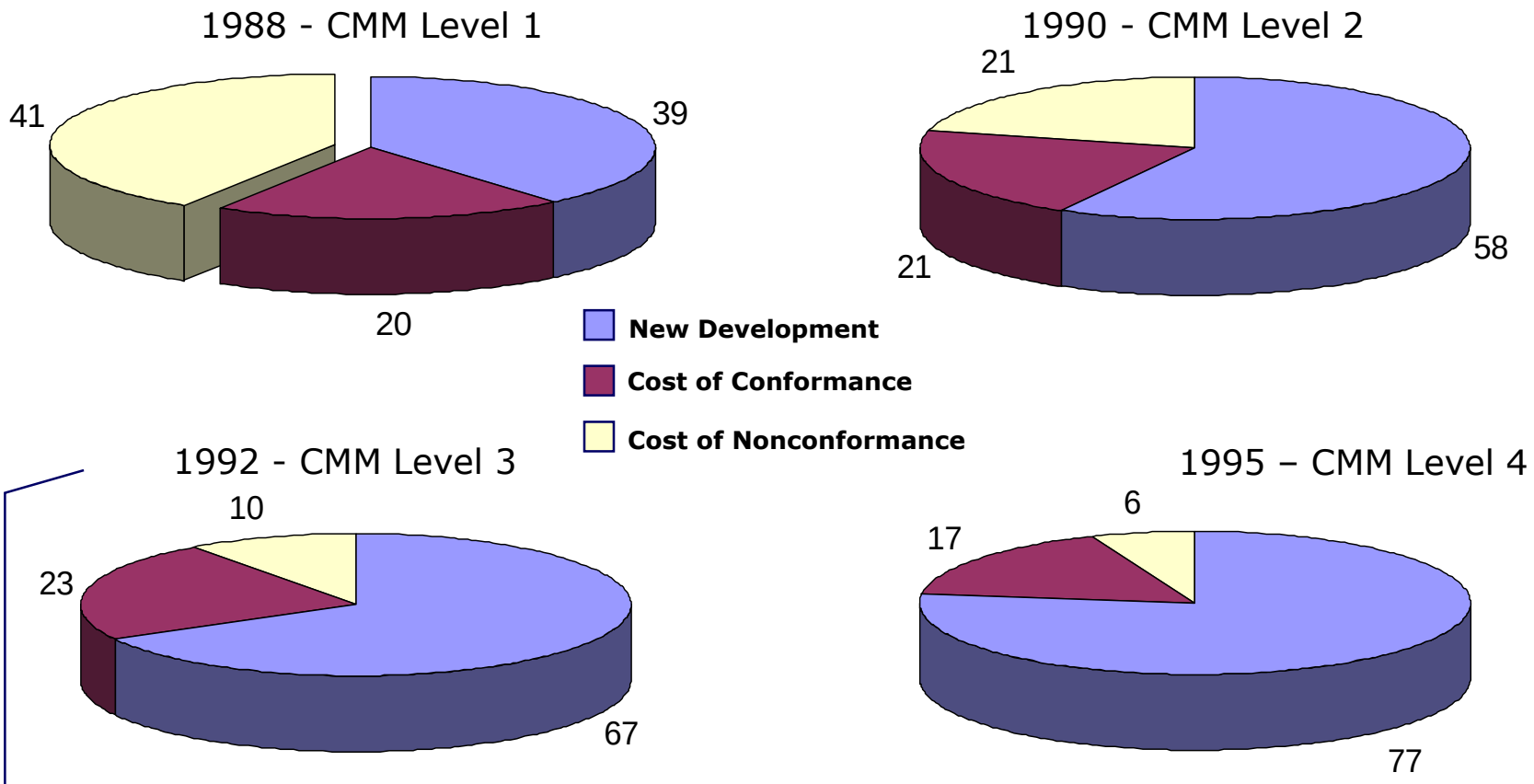
TS - Technical Solution

Преглед на: DAR - Decision Analysis and Resolution , IPM - Integrated Project Management , OPD - Organizational Process Definition , OPF - Organizational Process Focus, OT - Organizational Training , PI - Product Integration

Преглед на Maturity Level 4 и 5.

Обобщение на връзките между процесните области: Tying all together

Remember: 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 since ages...?

Bridges and Software

"The Roman bridges of antiquity were very inefficient structures. By modern standards, they used too much stone, and as a result, far too much labor to build.

Over the years we have learned to build bridges more efficiently, using fewer materials and less labor to perform the same task."

-- Tom Clancy, The Sum of All Fears

Chaos Report (Standish Research Group Report)

"The Roman bridges of antiquity were very inefficient structures. By modern standards, they used too much stone, and as a result, far too much labor to build. Over the years we have learned to build bridges more efficiently, using fewer materials and less labor to perform the same task." Tom Clancy (The Sum of All Fears)

Bridges are normally built on-time, on-budget, and do not fall down

Software "never" comes in on-time or on-budget. It always breaks down.

Bridge building did not always have such a stellar record – 3,000 years of experience, failures investigated & reported.

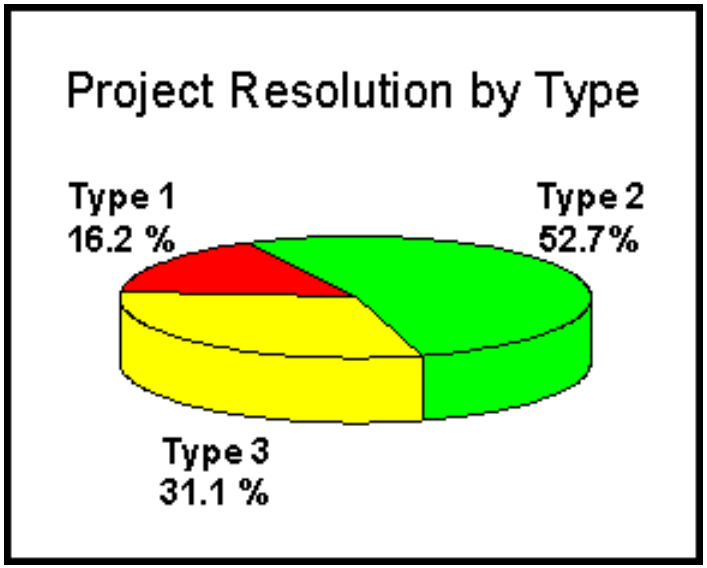
Computer industry failures are covered up, ignored, and/or rationalized. – mistakes repeated over and over again.

(1986, Alfred Spector, president of Transarc Corporation)

Project Success: Type 1. The project is completed on-time and on-budget, with all features and functions as initially specified. **(2000: 28%)**

Project Challenged: Type 2. The project is completed and operational but over-budget, over the time estimate, and offers fewer features and functions than originally specified. **(2000: 49%)**

Project Impaired: Type 3. The project is canceled at some point during the development cycle. **(2000: 23%)**



http://www.standishgroup.com/sample_research/chaos_1994_1.php

Bridge Building	Software Development
3000 years of experience in bridge building	30 years of software development experience
Solid Theoretical understanding of the physics of bridge structures	Some theoretical understanding of the software structures and logic. However real-world software is too complex to allow for proof of correctness.
Accurate and detailed modeling and design based on well established and mathematically proven laws and well-defined specifications.	There are hardly ever well-defined specifications or a frozen designs before the software is built. Software is viewed as "flexible" enough to keep up with today's fast moving business environment.
Knowledge of tolerances and load capacities of bridges is gathered before building the bridge	Difficult to determine resource and time consumption of software before it is built.
Bridge failures are thoroughly investigated and causes of failure are reported.	Software failures are covered up, ignored, and/or rationalized

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

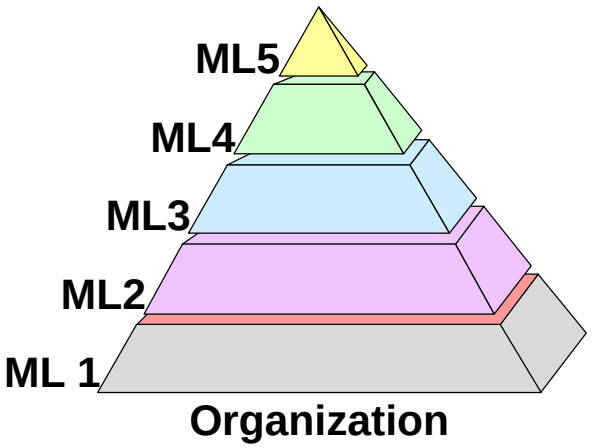
1 Process unpredictable, poorly controlled and reactive

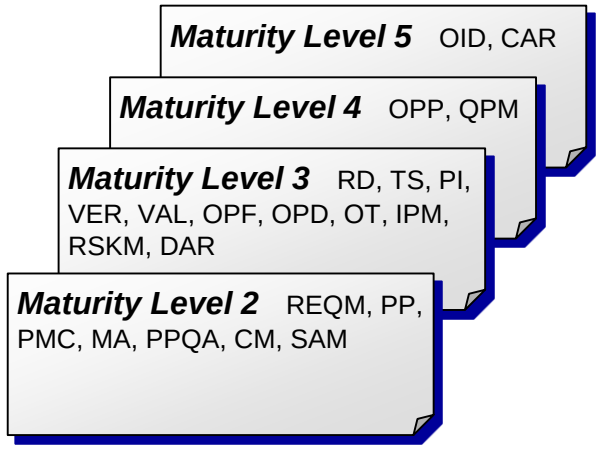
Performed
- Requirements flow in.
- A product is (sometimes) produced by some amorphous process.
- The product flows out and (we hope) works.

Maturity levels can not be skipped

CMMI DEV v 1.3 Presentaion

Staged





Process Areas

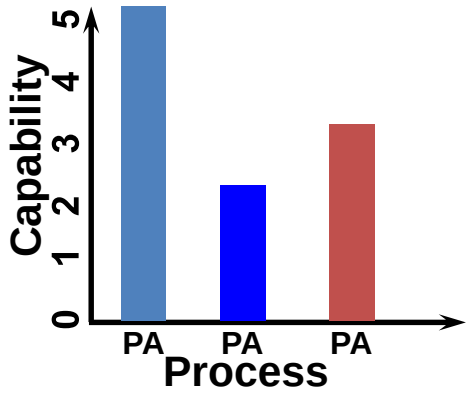
Organizational Innovation & Deployment (OID)
Causal Analysis and Resolution (CAR)

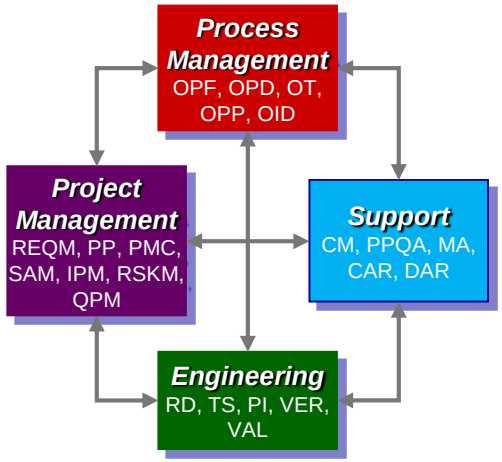
Organizational Process Performance (OPP)
Quantitative Project Management (QPM)

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

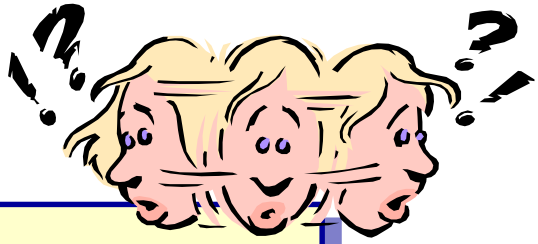
Requirements Management (REQM)
Project Planning (PP)
Project Monitoring and Control (PMC)
Supplier Agreement Management (SAM)
Measurement and Analysis (MA)
Process and Product Quality Assurance (PPQA)
Configuration Management (CM)

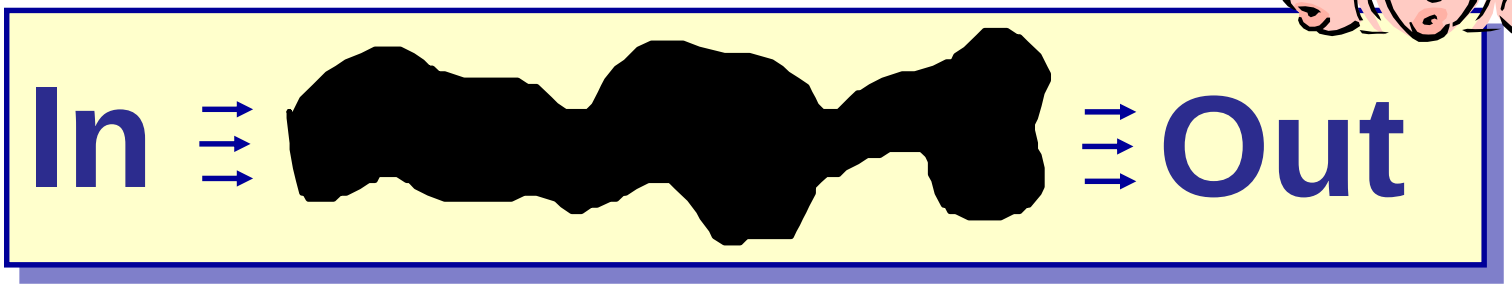
Continuous





ML1: Performance Is Unpredictable





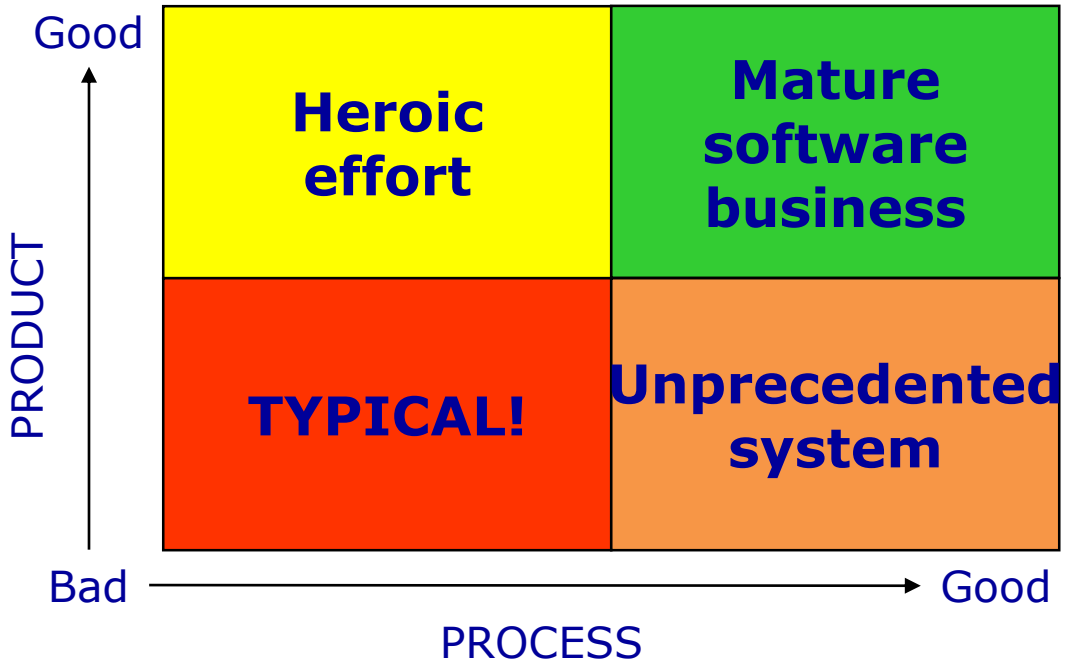
Requirements flow in.

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

The product flows out and (we hope) works.

REMEMBER? Corporate excellence – INTERNAL

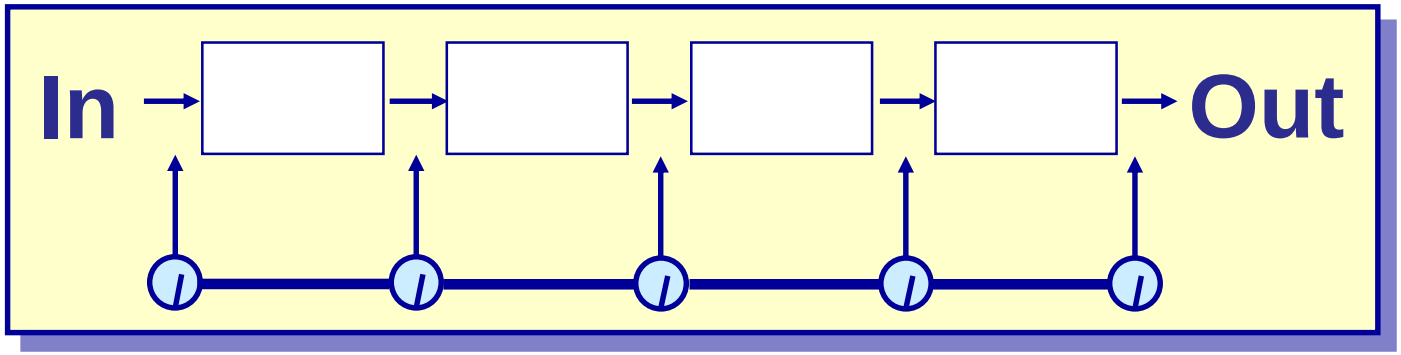
The corporate excellence is BASED on good internal processes



"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.

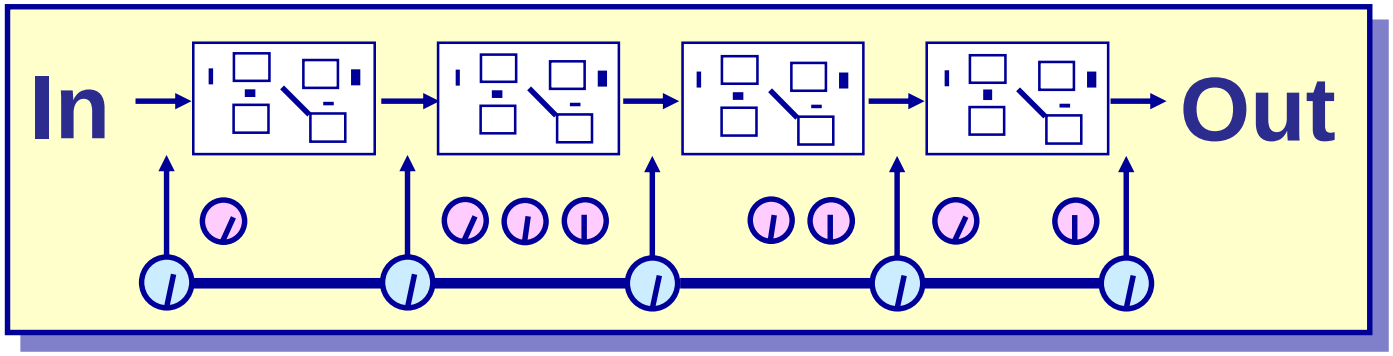
Plans are developed in accordance with policies.

Activities are performed in accordance with plans.

Measurements and reviews occur at defined points.

The product flows out and (usually) works.

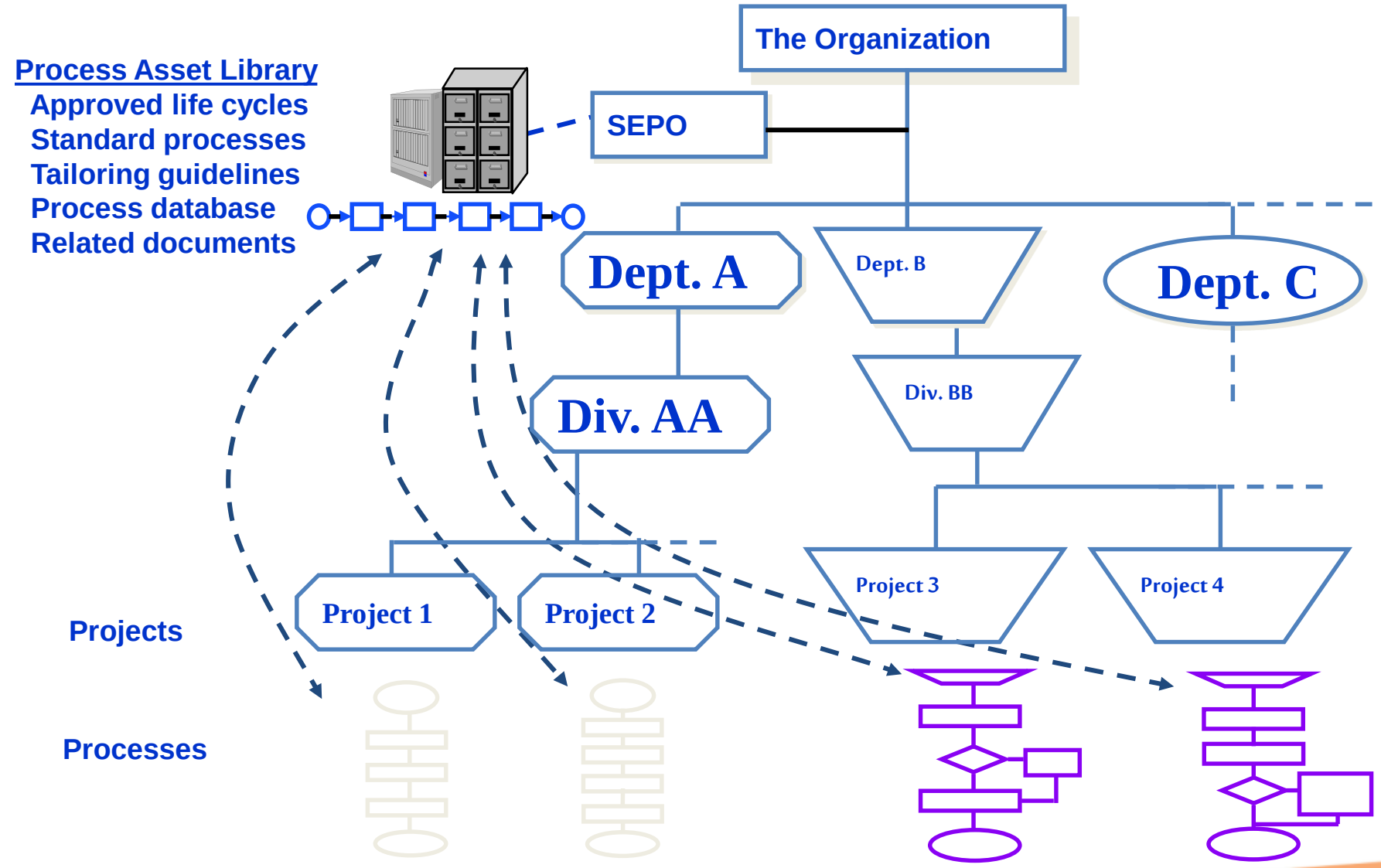
ML3: Managed According to a Defined Process



Commonality among projects allows more uniform estimation of performance.

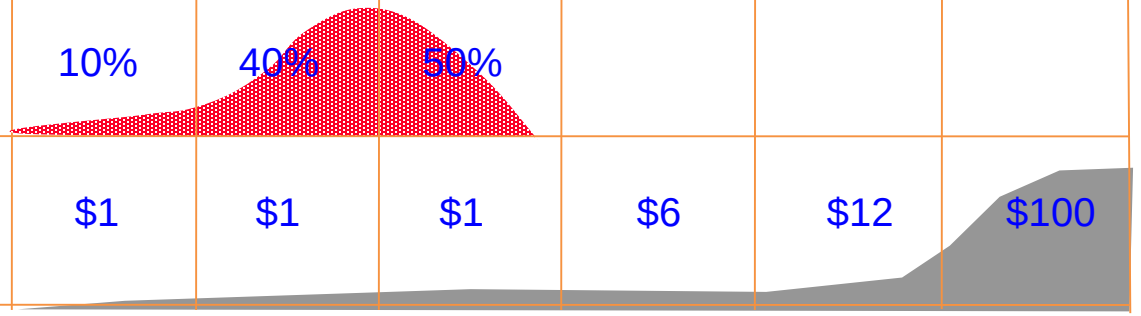
Sample Level 3 Organization

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



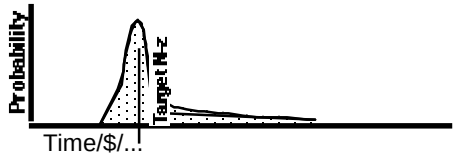
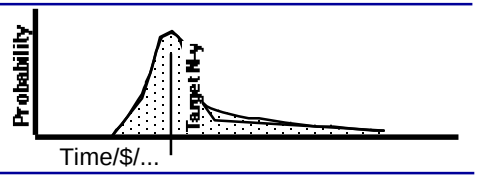
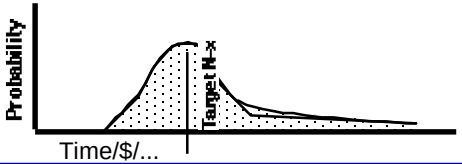
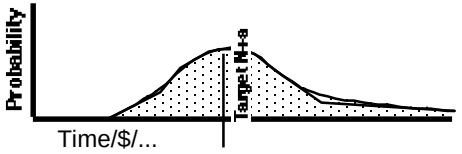
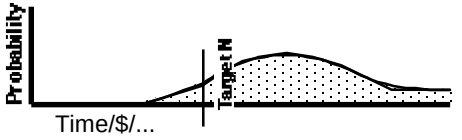
Remember:
Defects - Insertion Pattern & Cost of Removal

	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

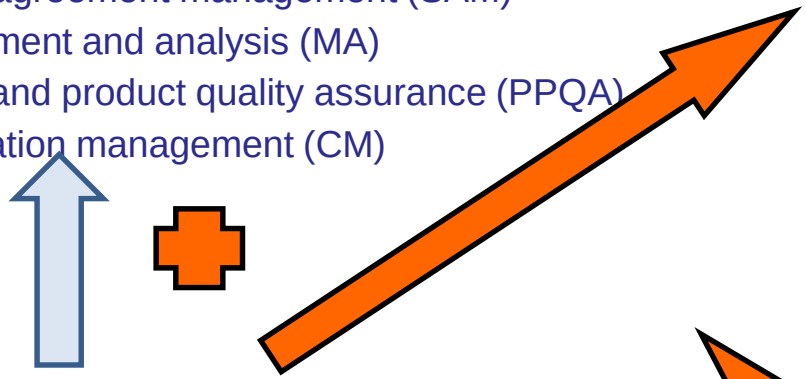
Evolution of Process Capability

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	

Maturity levels can not be skipped: generic + specific goals and practices

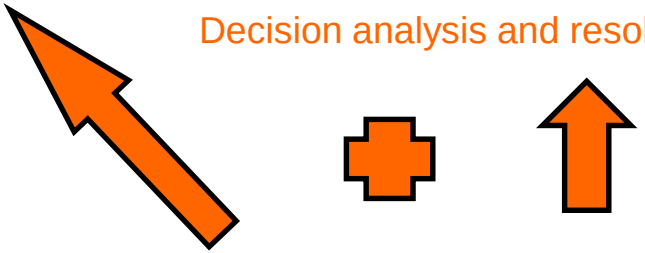
Maturity Level 2

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)



Maturity Level 3

Requirements development (RD)
Technical solution (TS)
Product integration (PI)
Verification (VER)
Validation (VAL)
Organizational process focus (OPF)
Organizational process definition (OPD)
Organizational training (OT)
Integrated project management (IPM)
Risk management (RSKM)
Decision analysis and resolution (DAR)



GG2 (ML2): Institutionalize a Managed Process

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

GG3 (ML3): Institutionalize a Defined Process

GP 3.1 Establish a defined process

GP 3.2 Collect process related experience *(improvement information)*

About Generic Goals and Institutionalization

The degree of institutionalization is embodied in the generic goals and expressed in the names of the processes associated with each goal as indicated below.

	Generic Goal and Title	Progression of Processes
GG 3	Institutionalize a Defined Process	**Defined Process**
GG 2	Institutionalize a Managed Process	**Managed Process**
GG 1	Achieve Specific Goals*	Performed Process

* This generic goal is only used in the continuous representation.

FROM ML2 > GG2 > GPs

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

GG2: Institutionalize a Managed Process

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 *(ex. manage configuration)*

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

FROM ML3 > GG3 > GPs

What is added to GG2 and applied to all PAs (from ML2 and up):

GG3 (ML3): Institutionalize a Defined Process

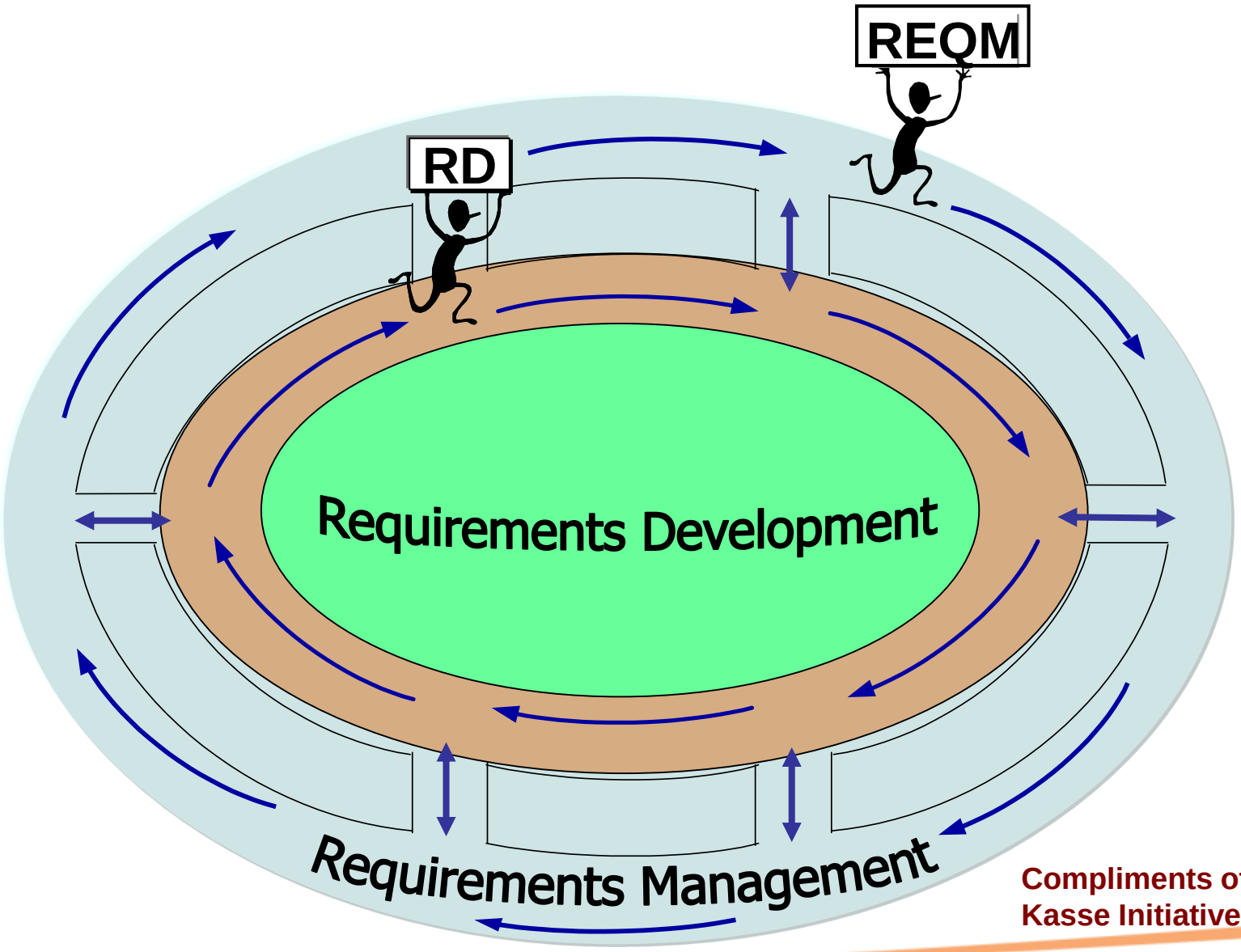
GP 3.1 Establish a defined process

GP 3.2 Collect process related experience *(ex. improvement information)*

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

Importance of Requirements Development

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

Establish solid **foundation** for downstream activities

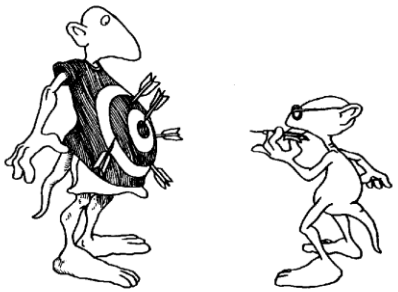
Benefits of Proper Requirements Development

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



Specific goals of RD

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.

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.

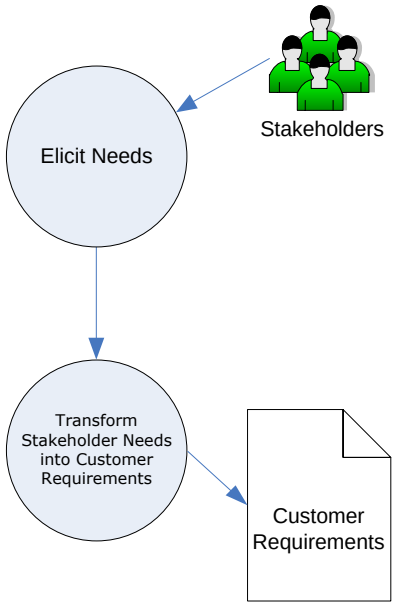
SG 1 Develop Customer Requirements

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

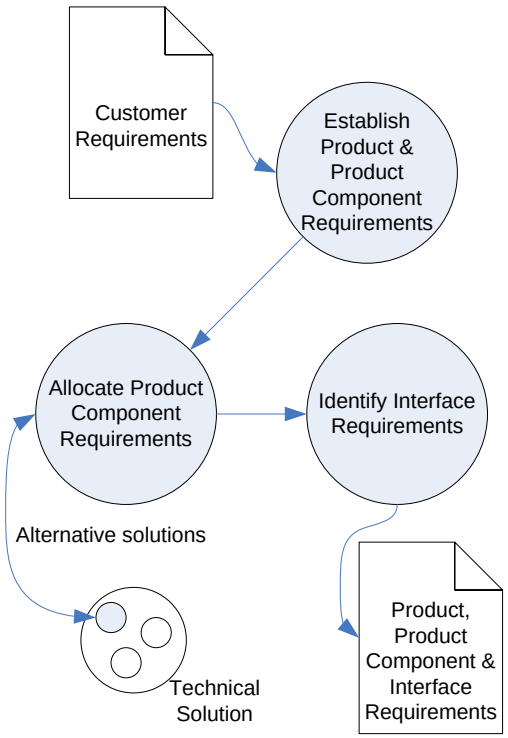
Establish and maintain product and product component requirements, which are based on the customer requirements.

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

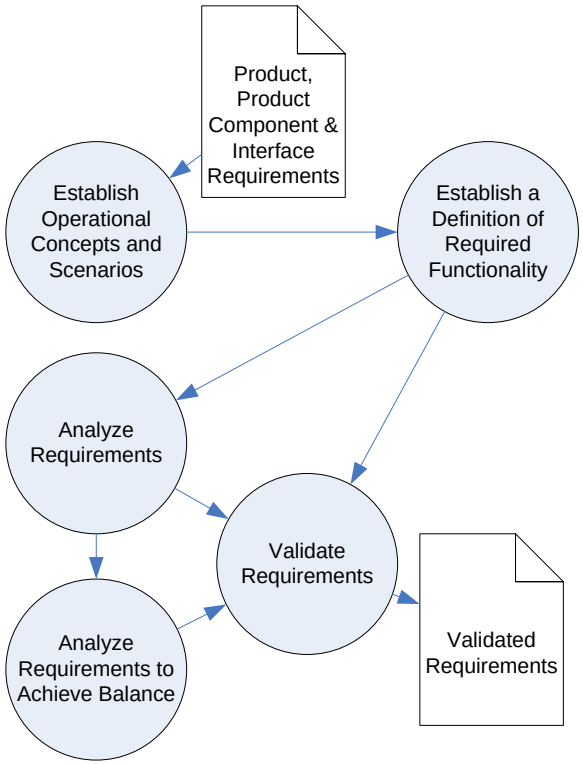
Analyze requirements to ensure that they are necessary and sufficient.

SP 3.4 Analyze Requirements to Achieve Balance

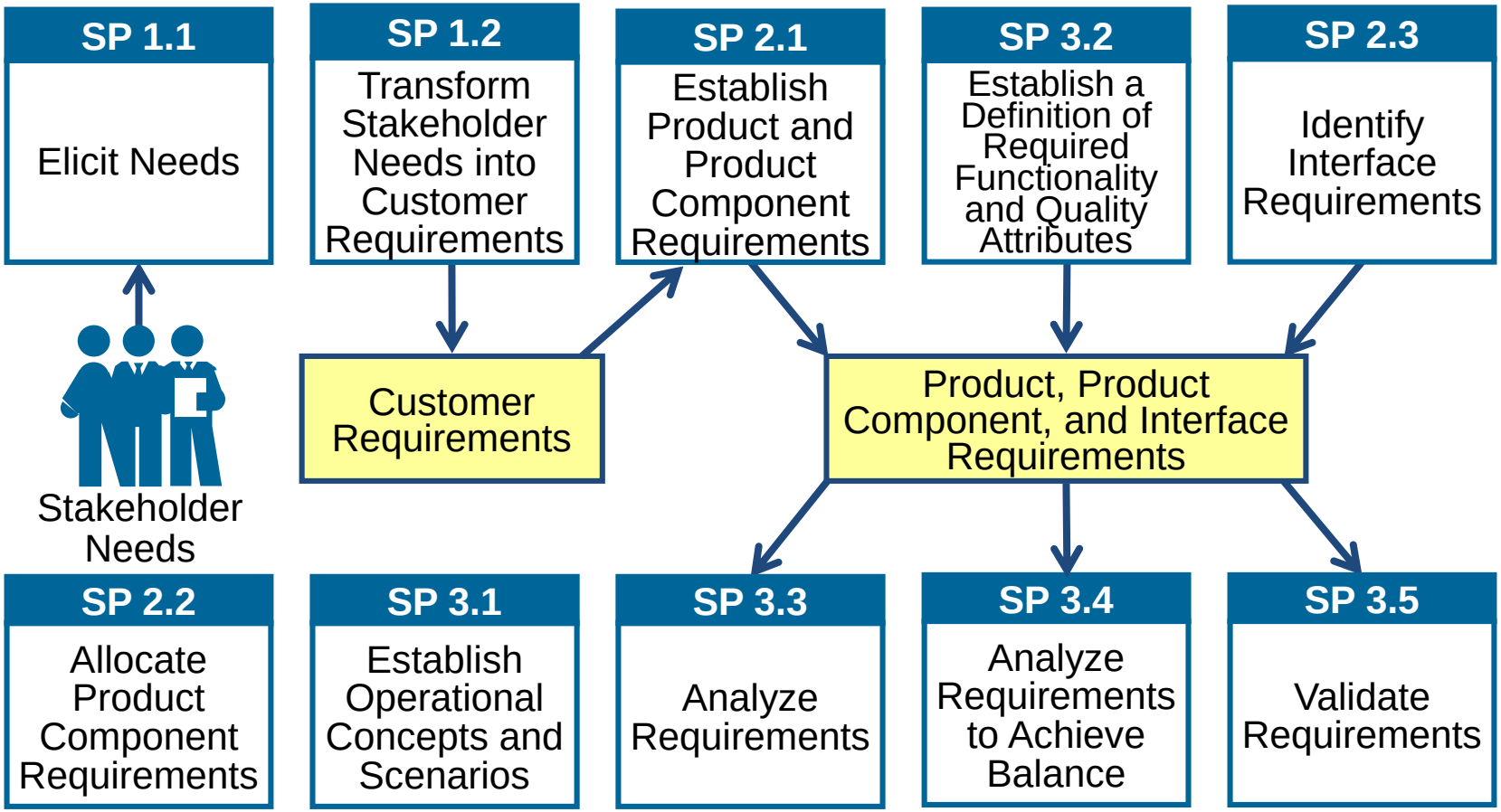
Analyze requirements to balance stakeholder needs and constraints.

SP 3.5 Validate Requirements

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



Requirements Development
Sampling of Work Products



How Requirements Development interacts with other Process Areas

Who does RD depend upon?

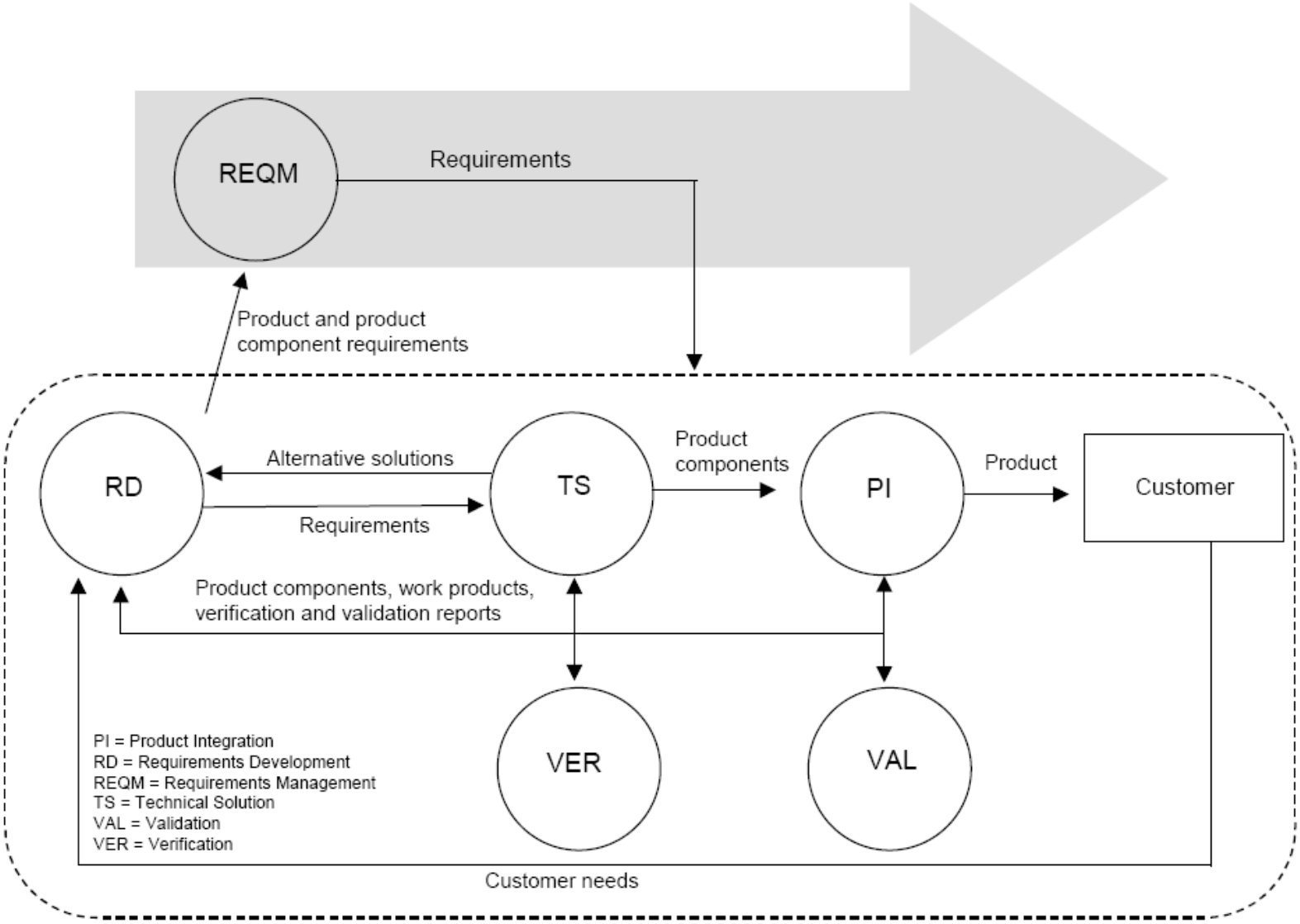
- Requirements Management (ML2:REQM) for managing requirements
- Technical Solution (ML3:TS) for development of alternative solutions and identification of product components
- Risk Management (ML3:RSKM) for identification and management of requirements risks

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



Verification Versus Validation

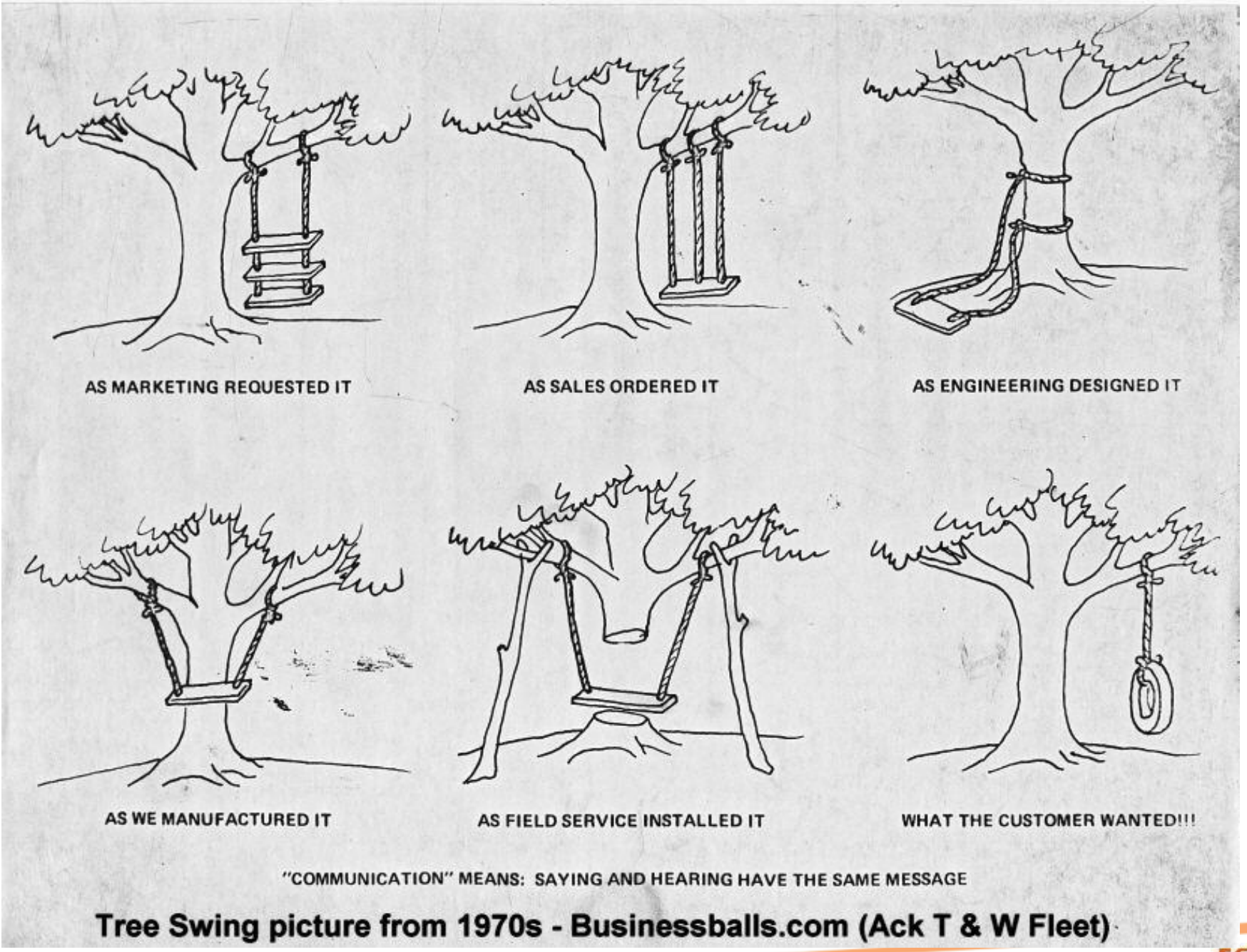
Verification (ML3:VER)

- Are you building the **product right**?
- That is, are you meeting the specified requirements?

Validation (ML3:VAL)

- Are you building the **right product**?
- That is, are you meeting the operational need?

Both are applicable throughout the product development lifecycle.


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:Verification (VER)

Purpose

Ensure that selected work products meet their specified requirements.

When Verification Is Not Done Well…

There is **disagreement** among technical staff as to whether the different components meet the requirements.

The product being tested **does not meet design requirements**.

Product reliability suffers because **defects** are not detected or corrected **prior to customer release**.

Added **rework** occurs because defects that could have been caught early escape into later lifecycle phases.

Verification Goals

SG 1: Prepare for Verification

Preparation for verification is conducted.

SG 2: Perform **Peer Reviews**

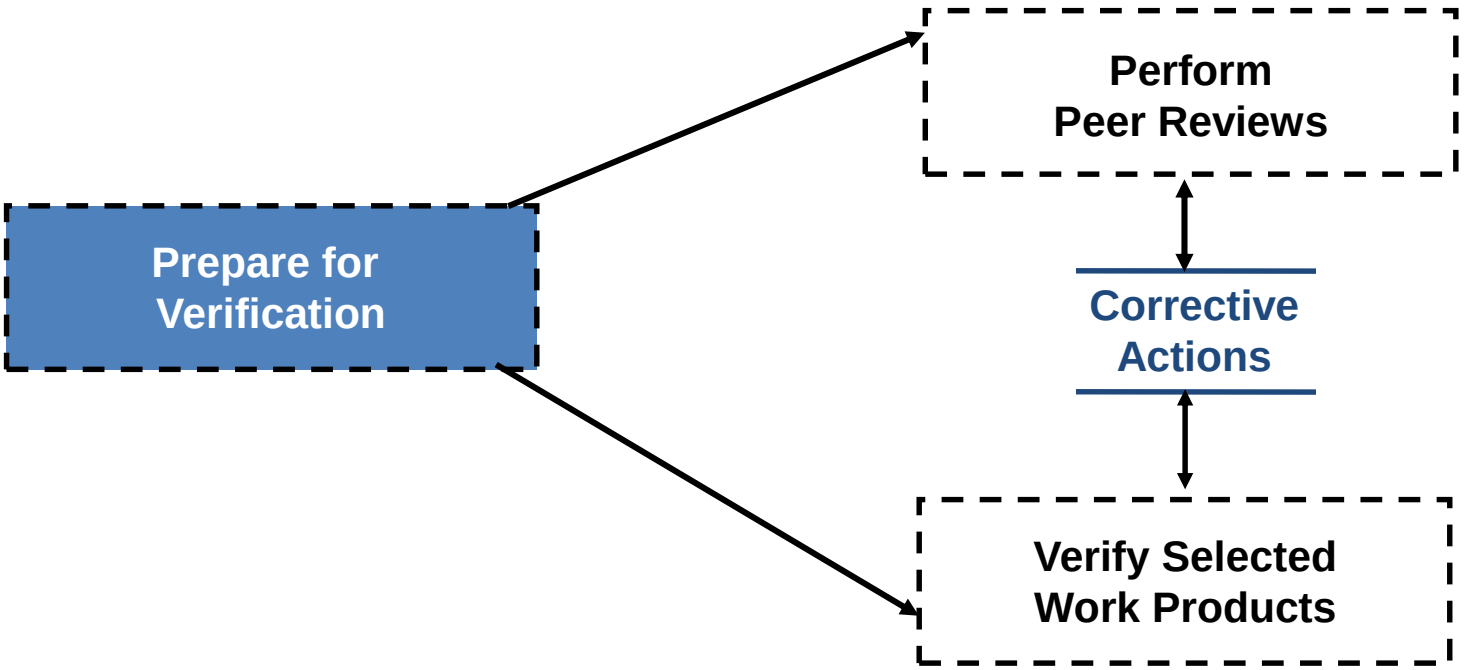
Peer reviews are performed on selected work products.

SG 3: Verify Selected Work Products

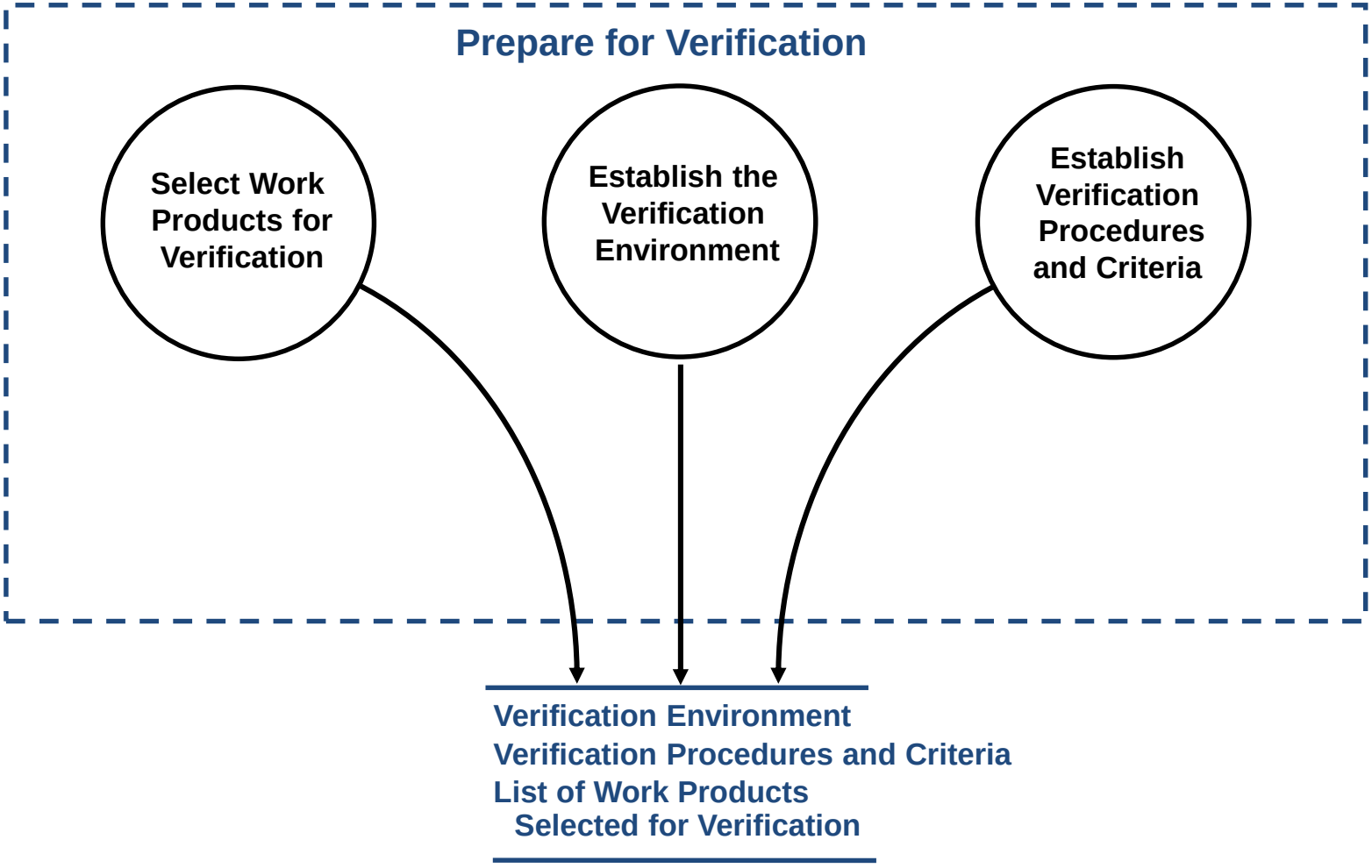
Selected work products are verified against their specified requirements.

The process area also has generic goals to support institutionalization.

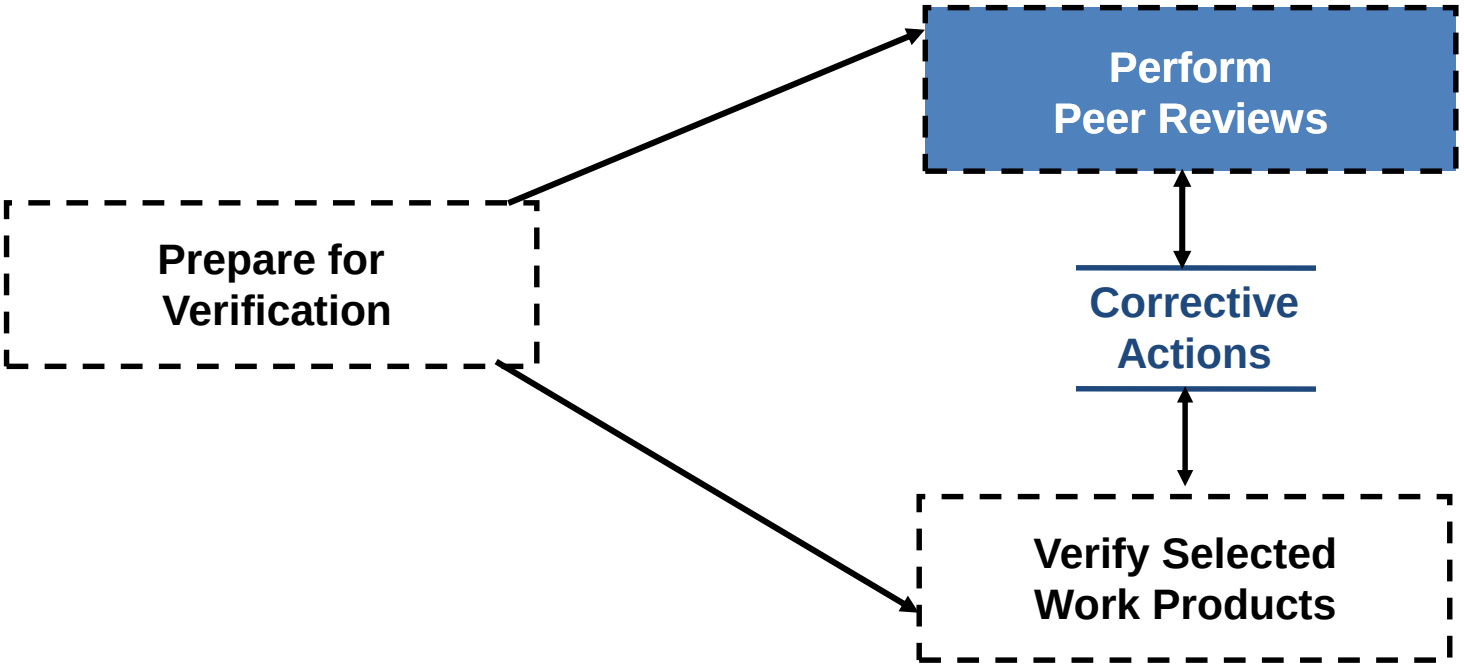
Verification Context -1



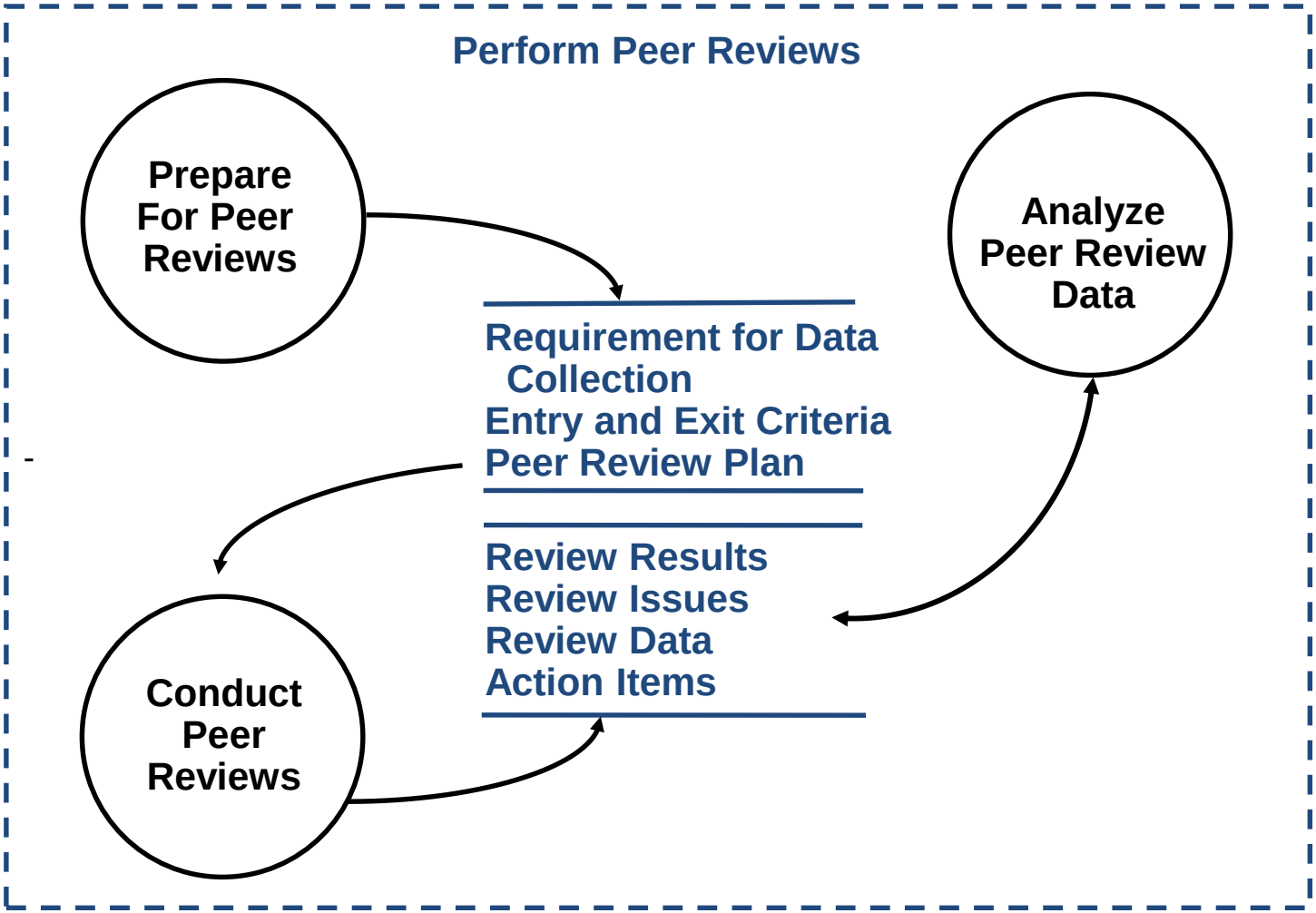
Verification Context -2



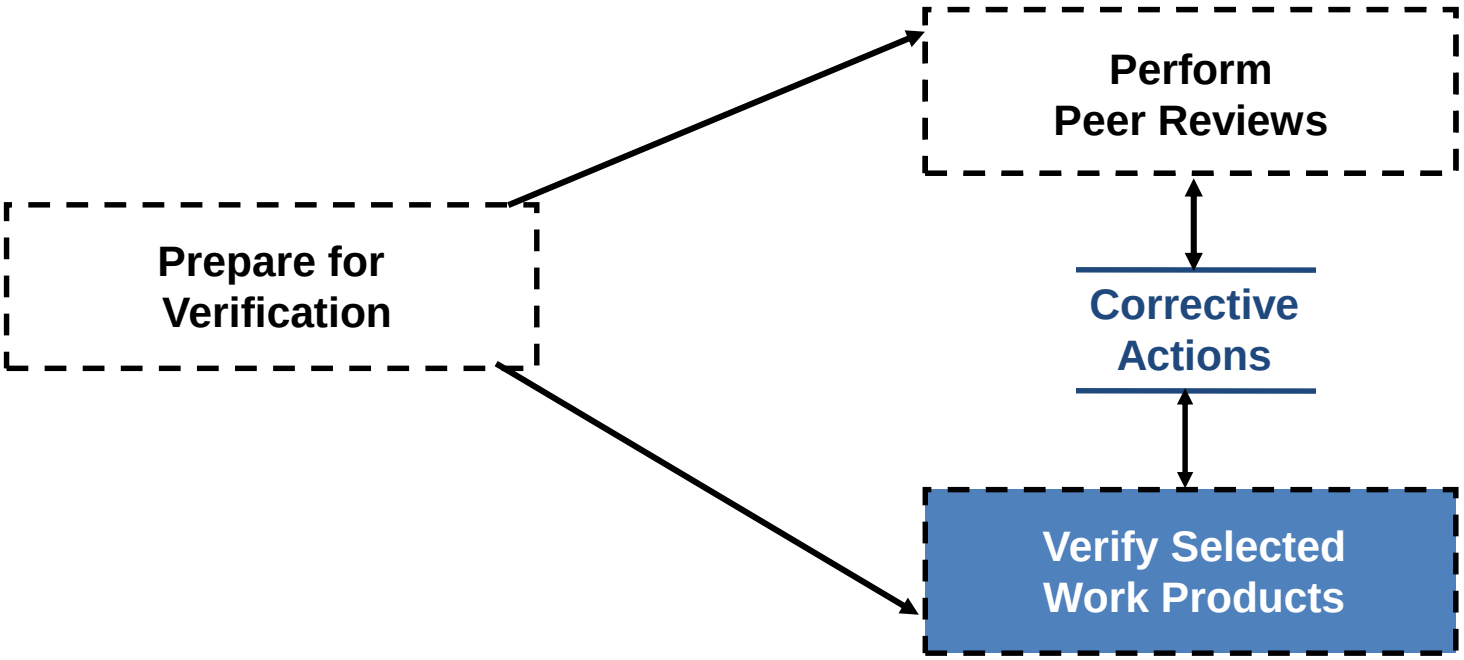
Verification Context -3



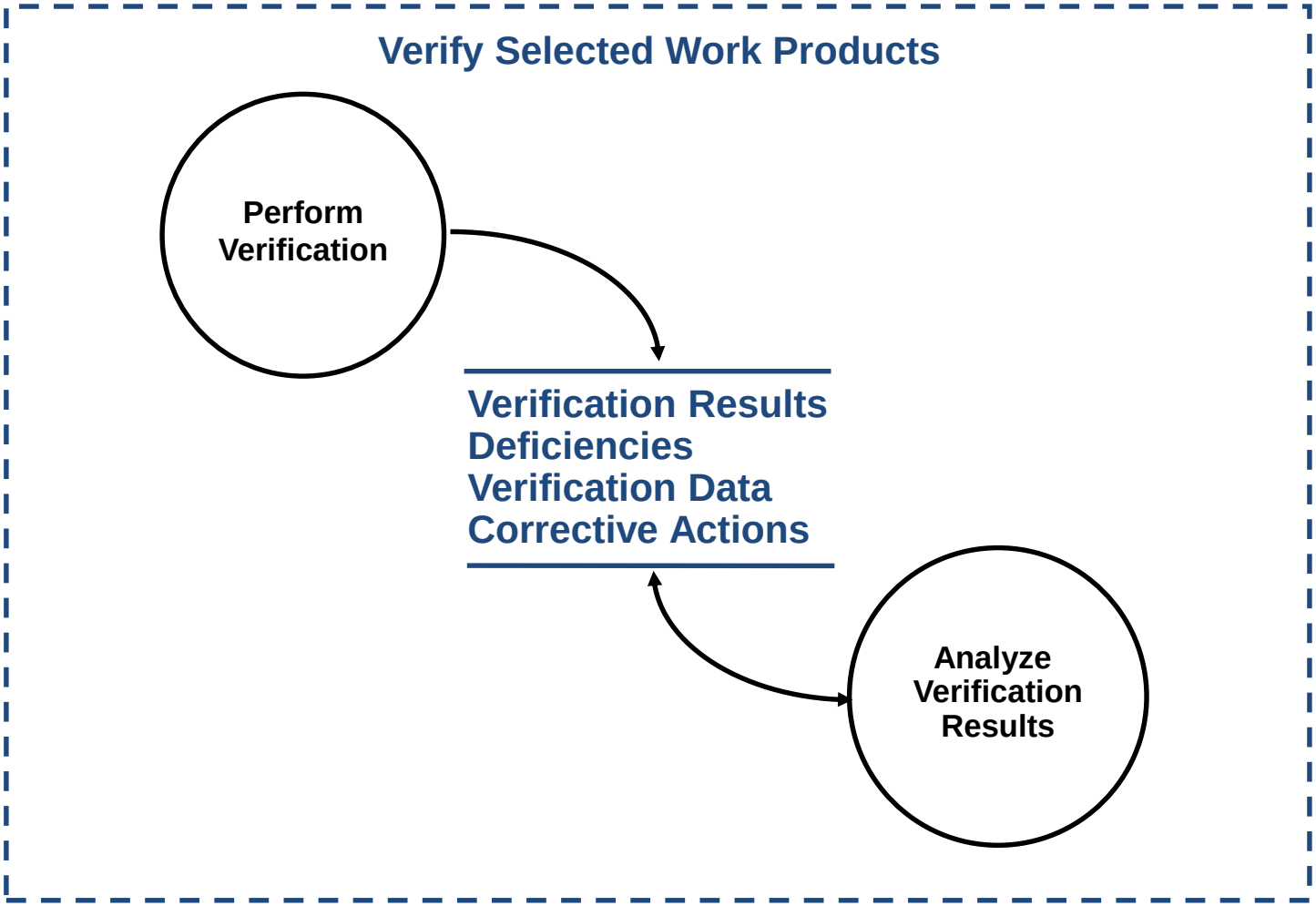
Verification Context -4



Verification Context -5



Verification Context -6



ML3: Validation (VAL)

Purpose

Demonstrate that a product or product component fulfills its intended use when placed in its intended environment.

When Validation Is Not Done Well…

There are **arguments** among the technical staff as to **what the user really wants**.

The released product **does not meet user expectations**.

Customers do not pay for products that do not meet their needs.

End users refuse to use the product as delivered.

Validation Goals

SG 1: Prepare for Validation

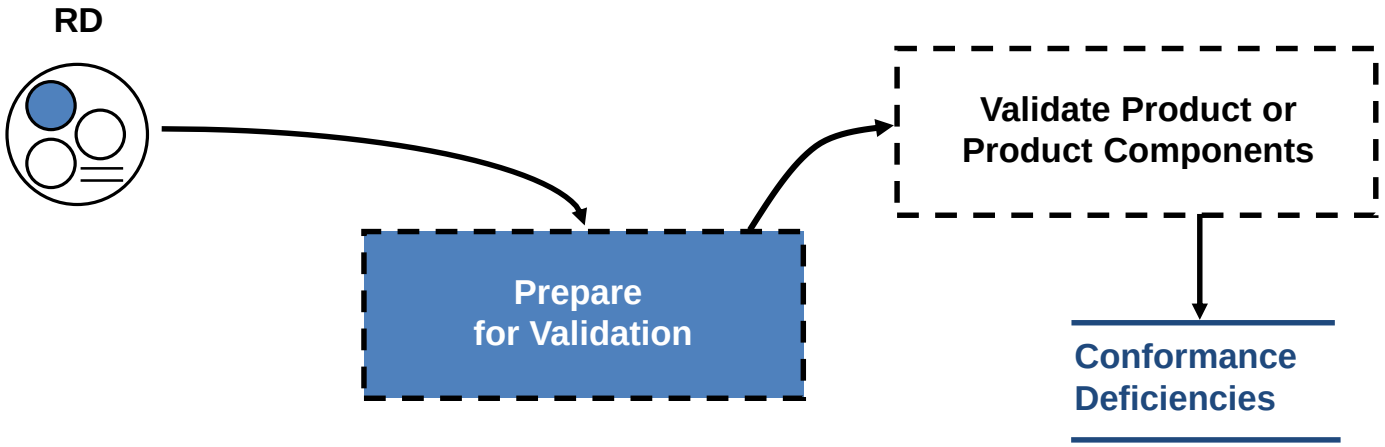
Preparation for validation is conducted.

SG 2: Validate Product or Product Components

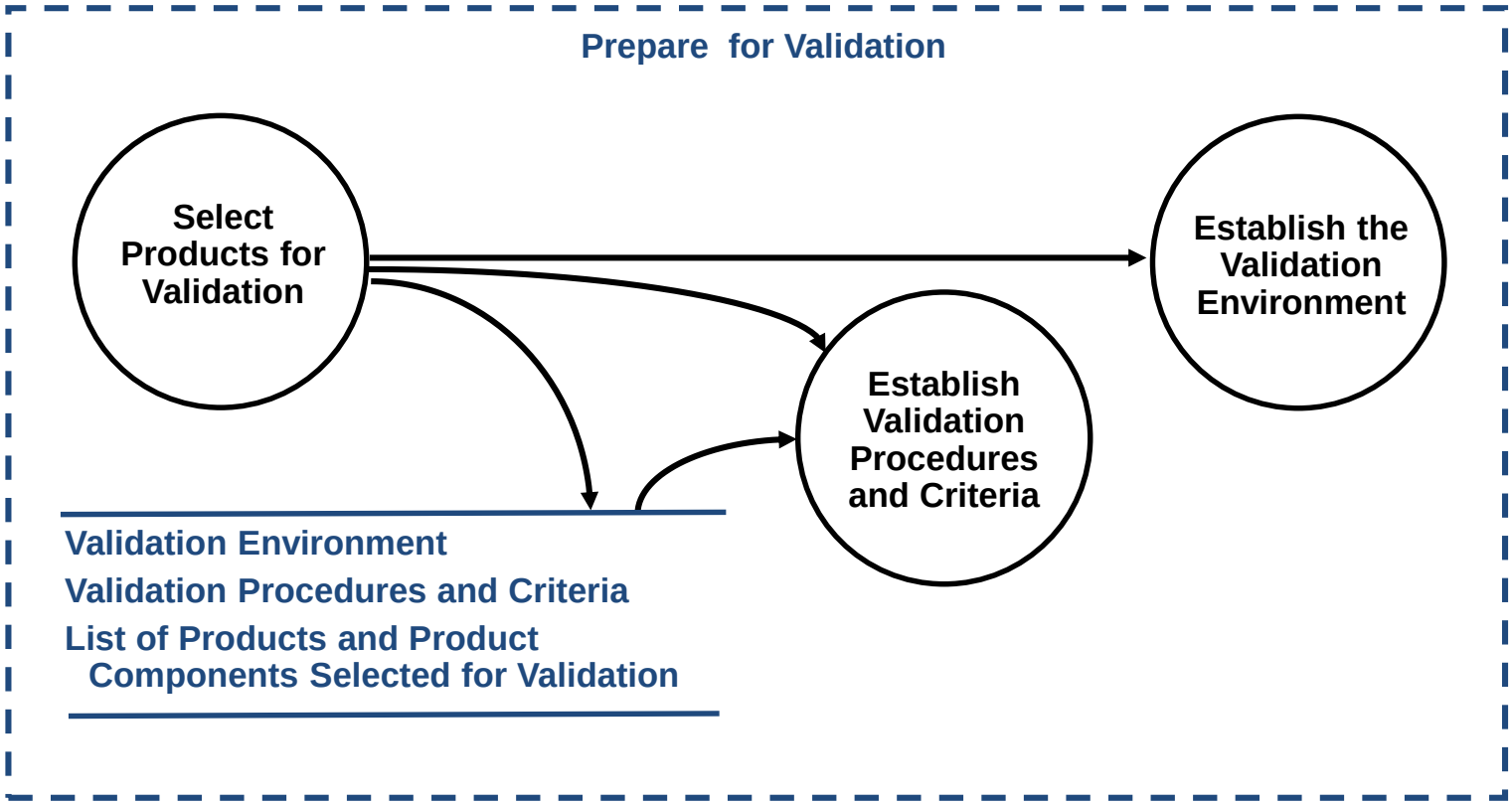
The product or product components are validated to ensure that they are suitable for use in their intended operating environment.

The process area also has generic goals to support institutionalization.

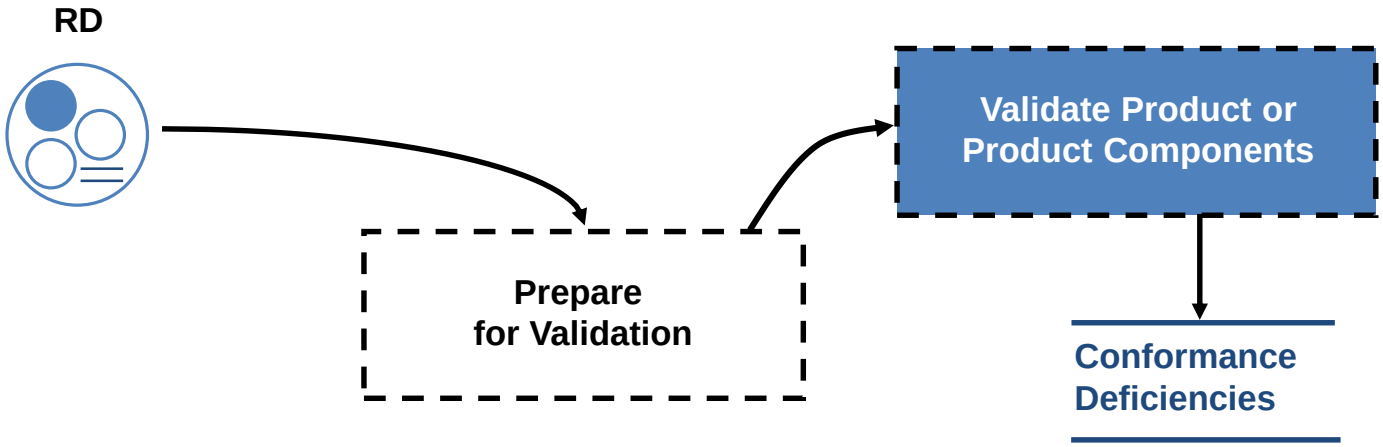
Validation Context -1



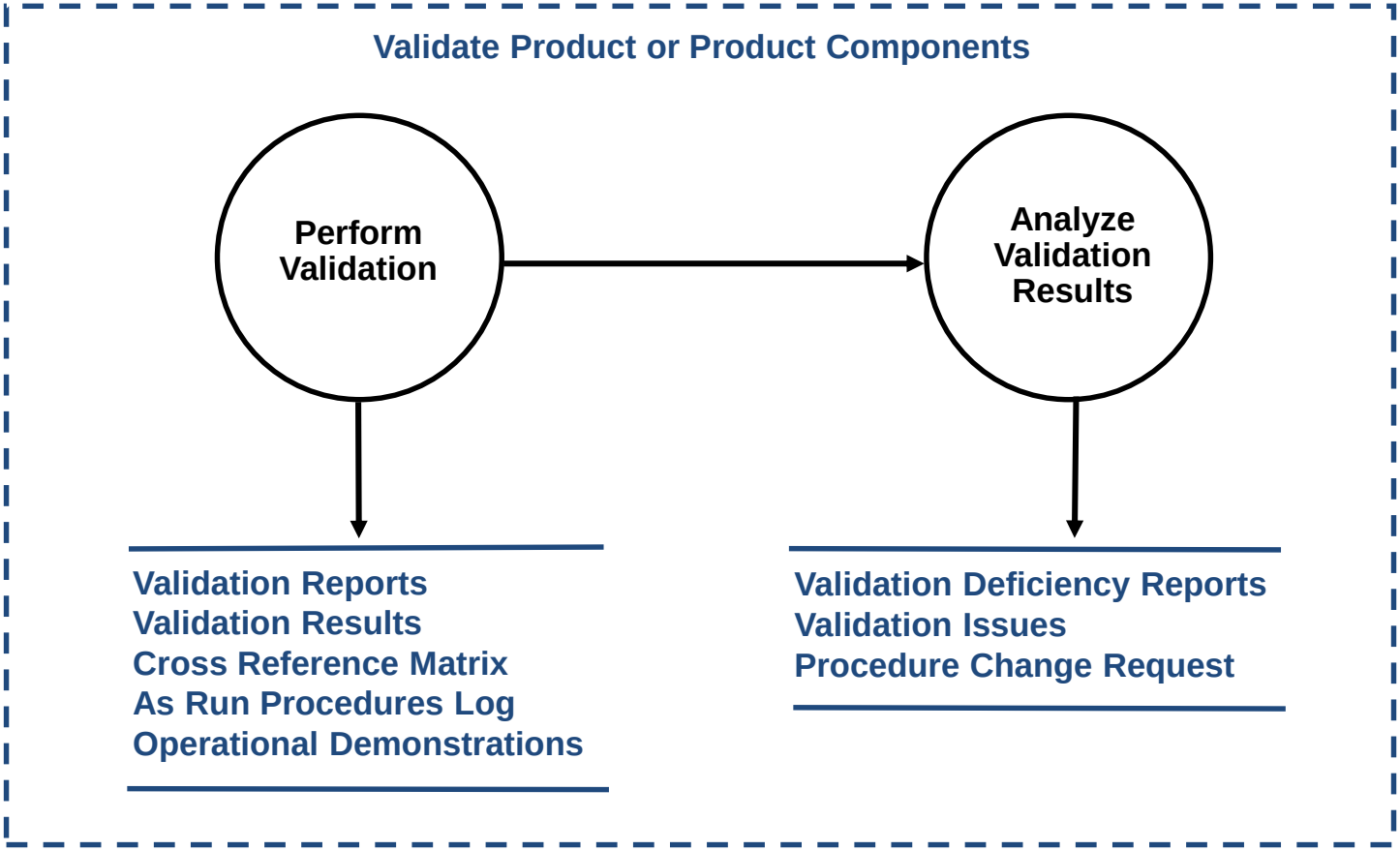
Validation Context -2



Validation Context -3



Validation Context -4



Sampling the Generic Practices

GP 2.3: Provide Resources

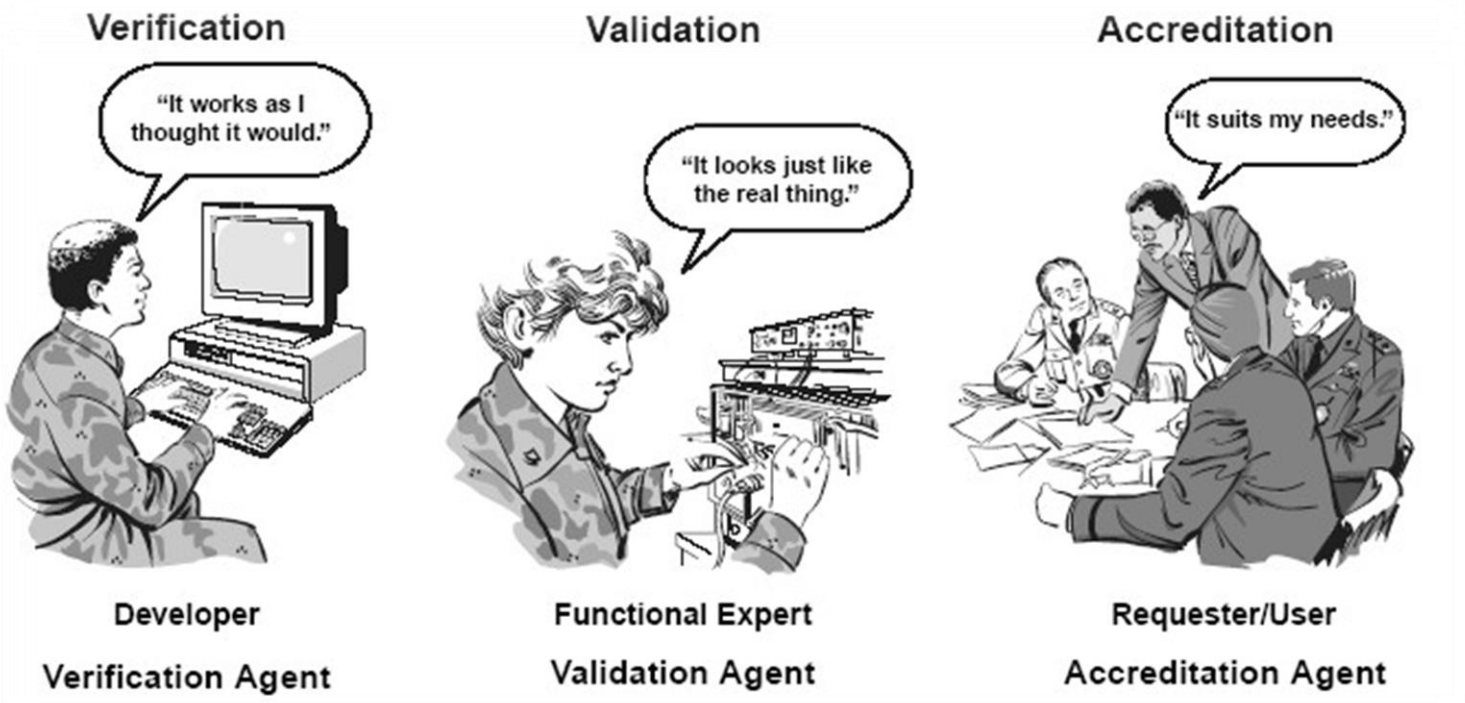
Provide adequate resources for performing the validation process, developing the work products, and providing the services of the process.

Elaboration for Validation

Examples of other resources provided include the following tools:

- test-management tools
- test-case generators
- test-coverage analyzers
- simulators
- load, stress, and performance tools

Verification - Validation



http://commons.wikimedia.org/wiki/File:Verification_Validation_Accreditation.jpg

ML3: Technical Solution (TS)

Purpose

Design, develop, and implement solutions to requirements. Solutions, designs, and implementations encompass products, product components, and product-related lifecycle processes either singly or in combinations as appropriate.

When Technical Solution Is Not Done Well…

An **ineffective solution** is chosen.

Products **may not meet** technical performance requirements or user needs.

Increased testing and rework is required to resolve design issues.

The product may not be able to accommodate **technology upgrades and future growth** if the technical solution is not well conceived.

Technical Solution Goals

SG 1: Select Product Component Solutions

Product or product component solutions are selected from **alternative solutions**.

SG 2: Develop the Design

Product or product **component designs** are developed.

SG 3: Implement the Product Design

Product components, and associated support documentation, **are implemented from their designs**.

The process area also has generic goals to support institutionalization.

Relevant Terminology

Product-related lifecycle processes

Processes associated with a product throughout one or more phases of its life (e.g., from conception through disposal), such as the manufacturing and support processes.

Sustainment

The processes used to ensure that a product can be utilized operationally by its end users or customers. Sustainment ensures that maintenance is done such that the product is in an operable condition whether or not the product is in use by customers or end users.

ML3: Risk Management (RSKM)

Purpose

Identify potential problems before they occur so that risk-handling activities can be planned and invoked as needed across the life of the product or project to mitigate adverse impacts on achieving objectives.

When Risk Management Is Not Done Well…

It is easy to **ignore risks** when they are not being tracked.

Risks that are known to project staff are often **not known to management**.

Repeated project failures due to unforeseen (but predictable) risks **can cost you business**.

Risk Management Goals

SG 1: Prepare for Risk Management

Preparation for risk management is conducted.

SG 2: Identify and Analyze Risks

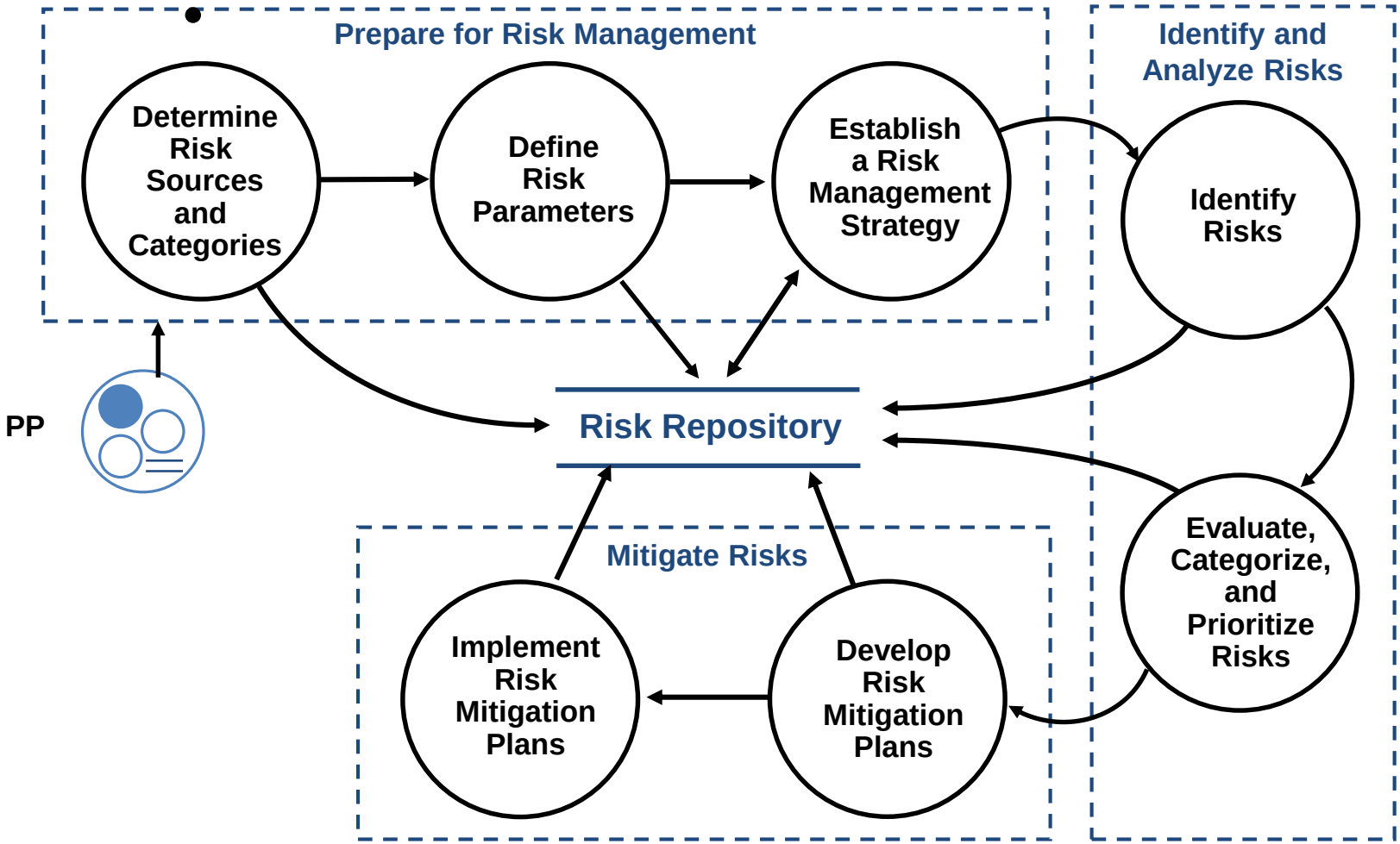
Risks are identified and analyzed to determine their relative importance.

SG 3: Mitigate Risks

Risks are handled and mitigated, where appropriate, to reduce adverse impacts on achieving objectives.

The process area also has generic goals to support institutionalization.

Risk Management Context



Sampling the Generic Practices

GP 2.10: Review Status with Higher Level Management

Review the activities, status, and results of the risk management process with higher level management and resolve issues.

Elaboration for Risk Management

Reviews of the project risk status are held on a periodic and event-driven basis with appropriate levels of management, to provide visibility into the potential for project risk exposure and appropriate corrective action.

Typically, these reviews will include a summary of the most critical risks, key risk parameters (such as likelihood and consequence of these risks), and the status of risk mitigation efforts.

Higher Levels of (CMMI) Maturity Lead to Lower Risk

Level 2 expects a **start at risk management**

- Project Planning SP 2.2 **Identify and analyze project risks**

Level 3 provides the **Risk Management Process Area**

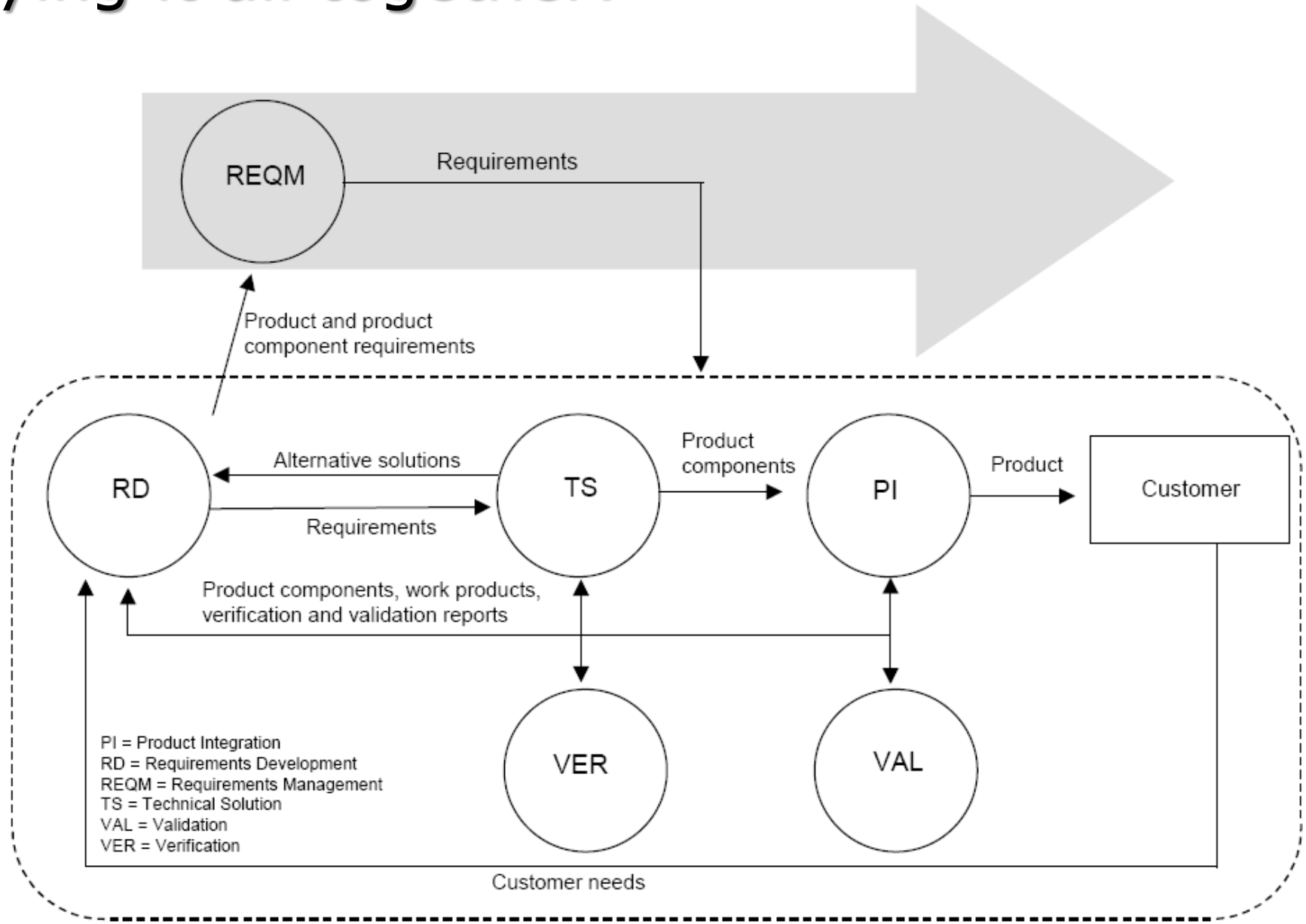
- Establishes a defined process with additional breadth of subject and organizational coverage
- Risk sources and categories used to more effectively identify and handle risks.

Level 4 quantitatively defines the **impact of risk on project success**

- Process volatility a major source of risk
- Data allows better prioritization and control of risks

Level 5 activities produce action proposals which often **address sources of high risk**

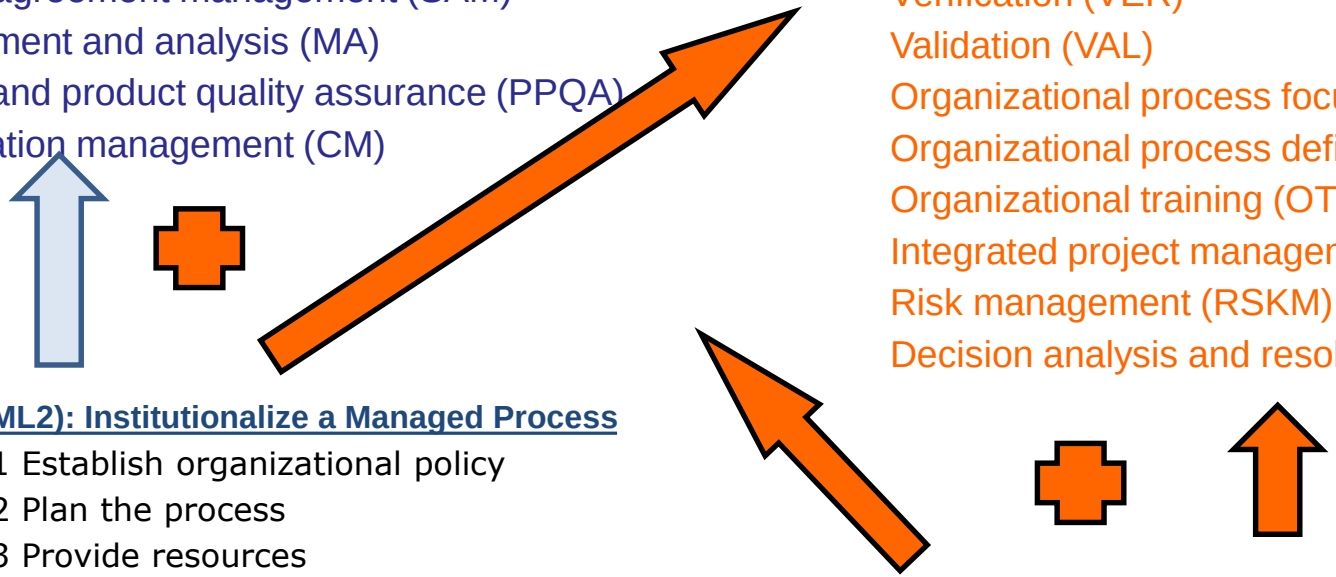
Tying it all together:



REMEMBER:
Maturity levels can not be skipped!!!

Maturity Level 2

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)



Maturity Level 3

Requirements development (RD)
Technical solution (TS)
Product integration (PI)
Verification (VER)
Validation (VAL)
Organizational process focus (OPF)
Organizational process definition (OPD)
Organizational training (OT)
Integrated project management (IPM)
Risk management (RSKM)
Decision analysis and resolution (DAR)

GG2 (ML2): Institutionalize a Managed Process

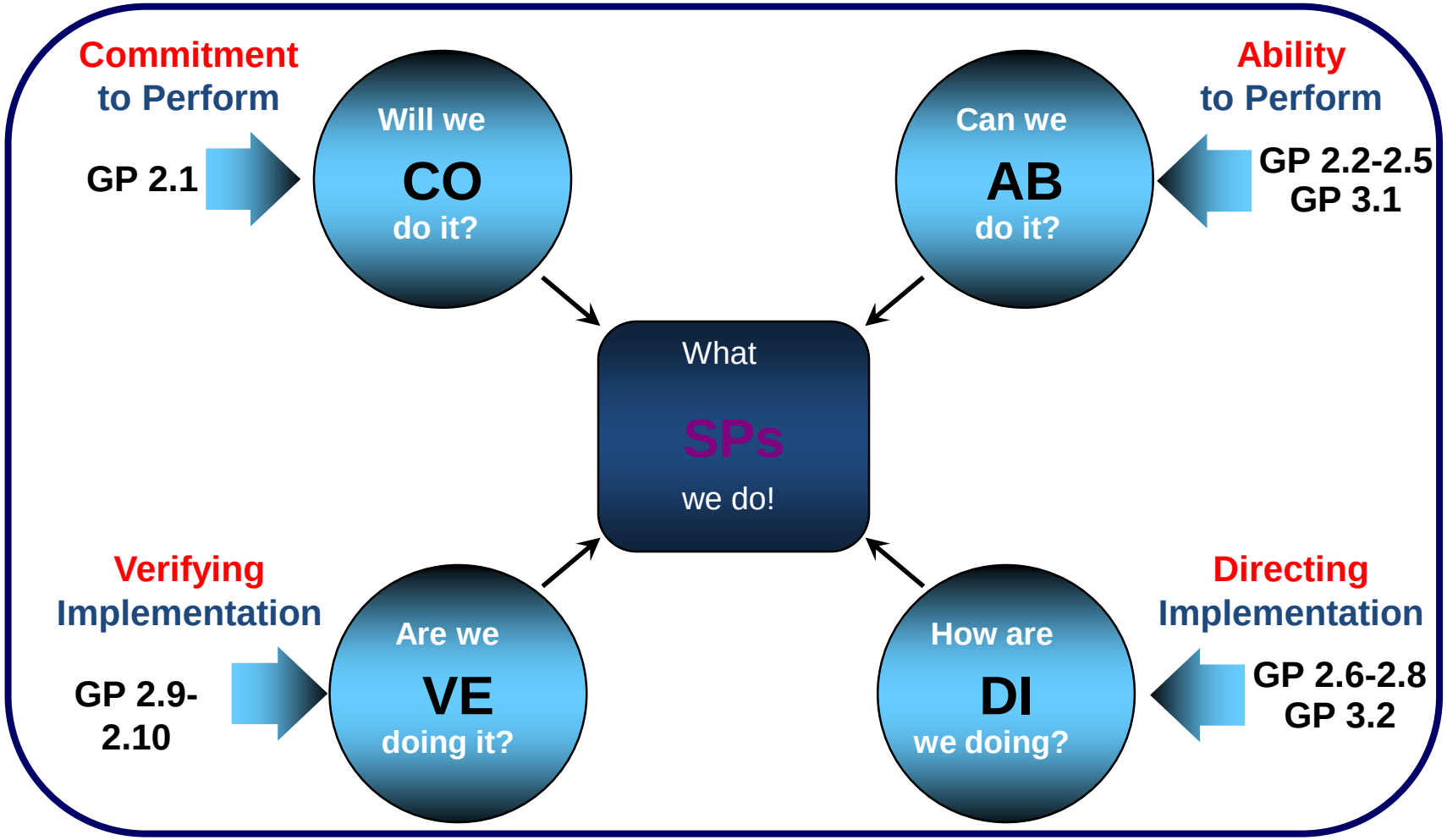
GP 2.1 Establish organizational policy
GP 2.2 Plan the process
GP 2.3 Provide resources
GP 2.4 Assign responsibility
GP 2.5 Train people
GP 2.6 Control Work Products *(manage configuration)*
GP 2.7 Identify and involve relevant stakeholders
GP 2.8 Monitor and control the process
GP 2.9 Objectively evaluate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

GG3 (ML3): Institutionalize a Defined Process

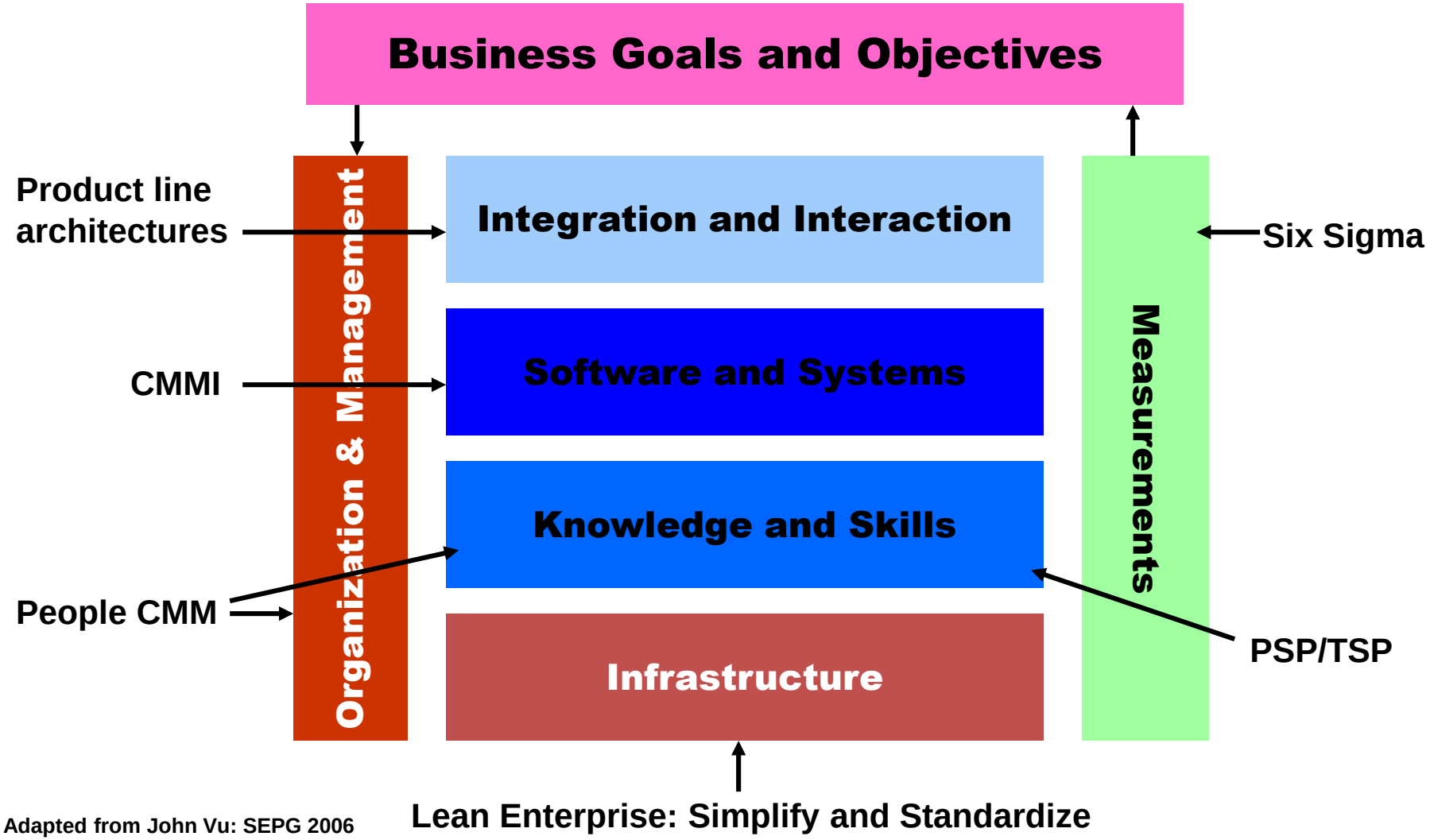
GP 3.1 Establish a defined process

GP 3.2 Collect process related experience *(improvement information)*

Common Features (GPs) - Basis for Institutionalization



Multiple Models/Technologies Architectures



Adapted from John Vu: SEPG 2006

CMMI and other models