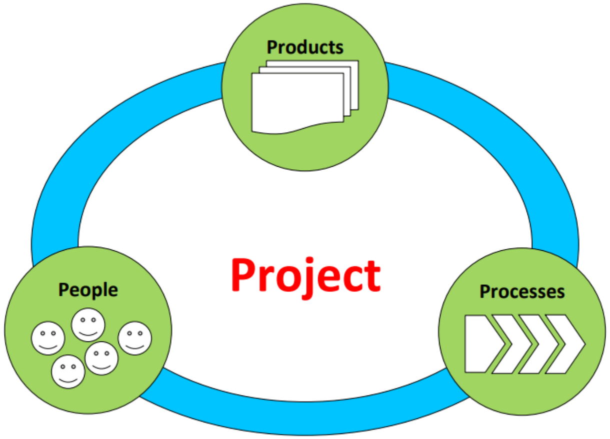
Programų kūrimo procesas

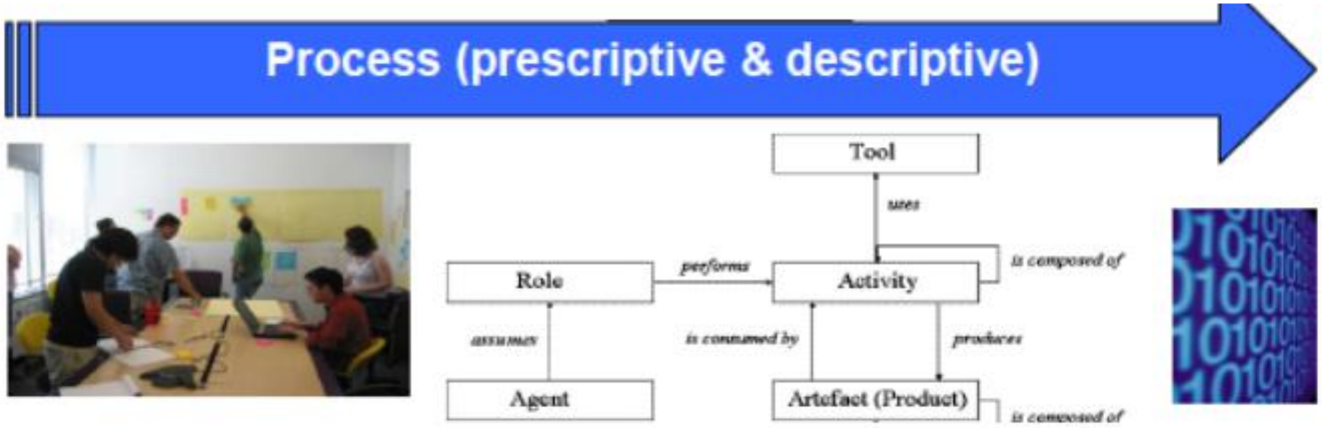
Dr. Asta Slotkienė

The Three Ps in Software Engineering



Model World

People (abstract)



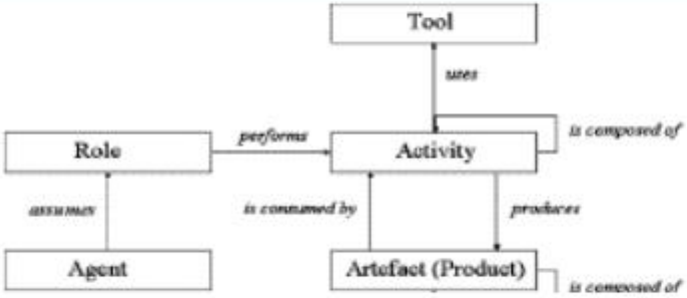
Products (abstract)

The six Ps in Software Engineering

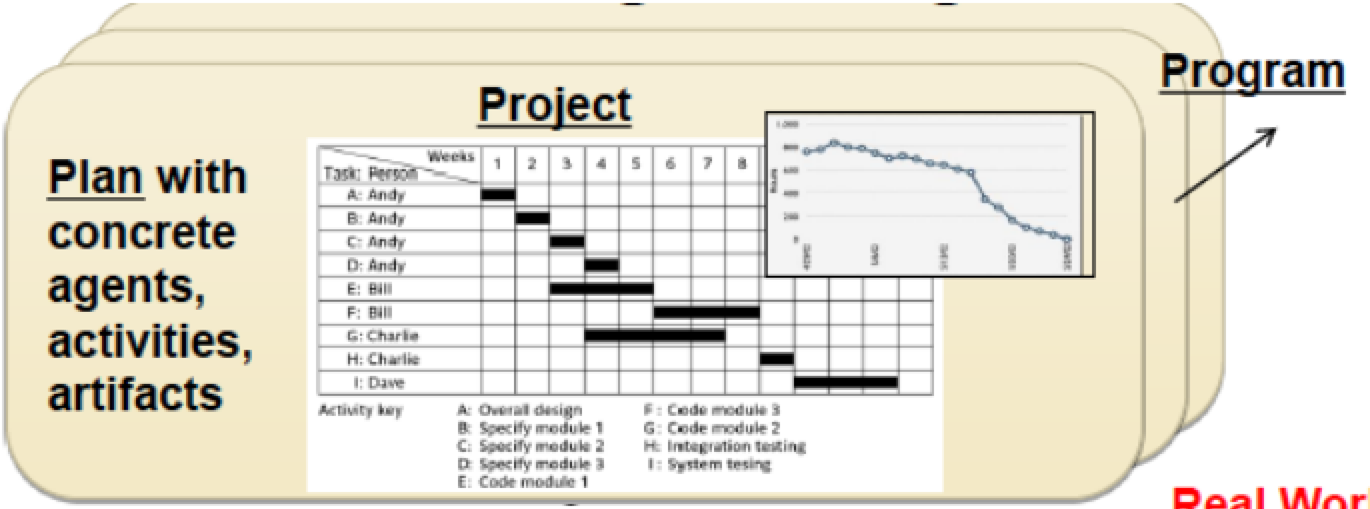
Process (prescriptive & descriptive)

Model World

People (abstract)



Products (abstract)



Project

Plan with concrete agents, activities, artifacts

Task: Person \ Weeks	1	2	3	4	5	6	7	8					
A: Andy	■												
B: Andy		■											
C: Andy			■										
D: Andy				■									
E: Bill			■	■	■								
F: Bill						■	■	■					
G: Charlie				■	■	■	■						
H: Charlie									■				
I: Dave										■	■	■	

Activity key		
	A: Overall design	F : Code module 3
	B: Specify module 1	G : Code module 2
	C: Specify module 2	H: Integration testing
	D: Specify module 3	I : System tesing
	E: Code module 1	

Program

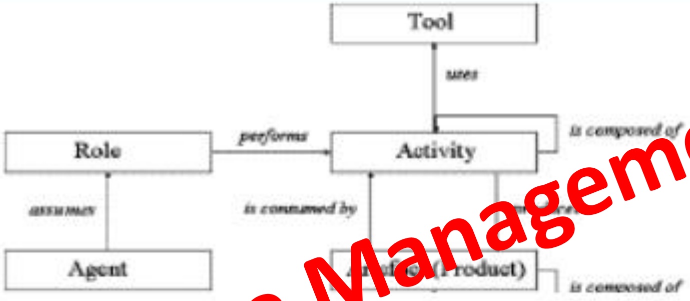
Real World

The six Ps in Software Engineering

Process (prescriptive & descriptive)

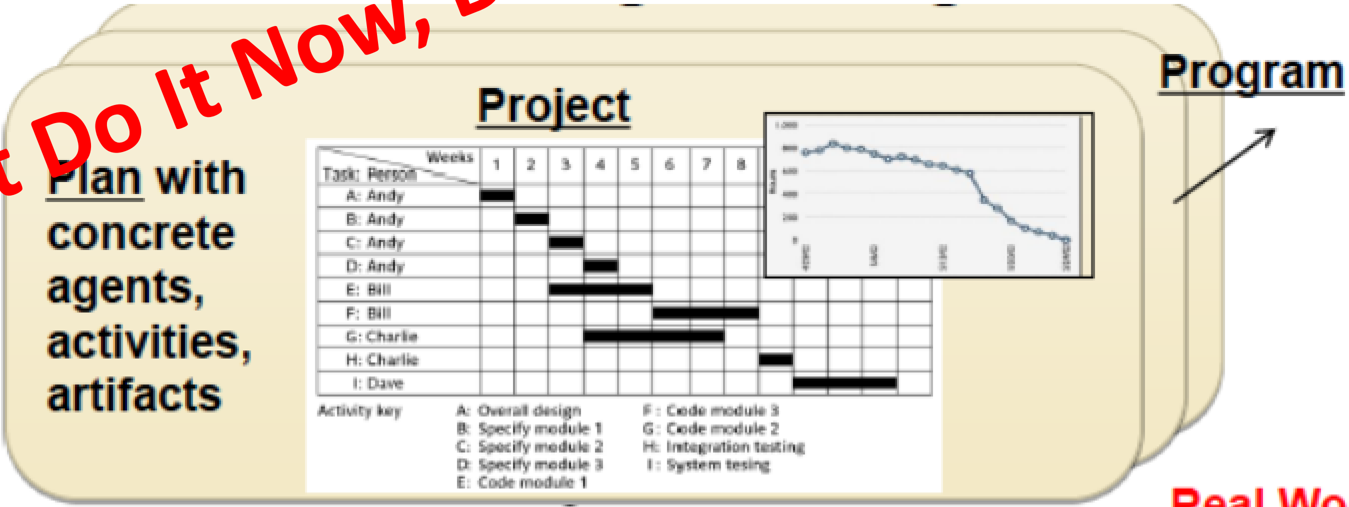
Model World

People (abstract)



Products (abstract)

Just Do It Now, Damage Management

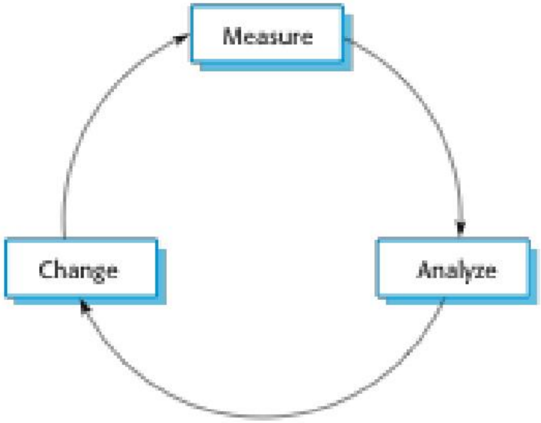


Plan with concrete agents, activities, artifacts

Real World

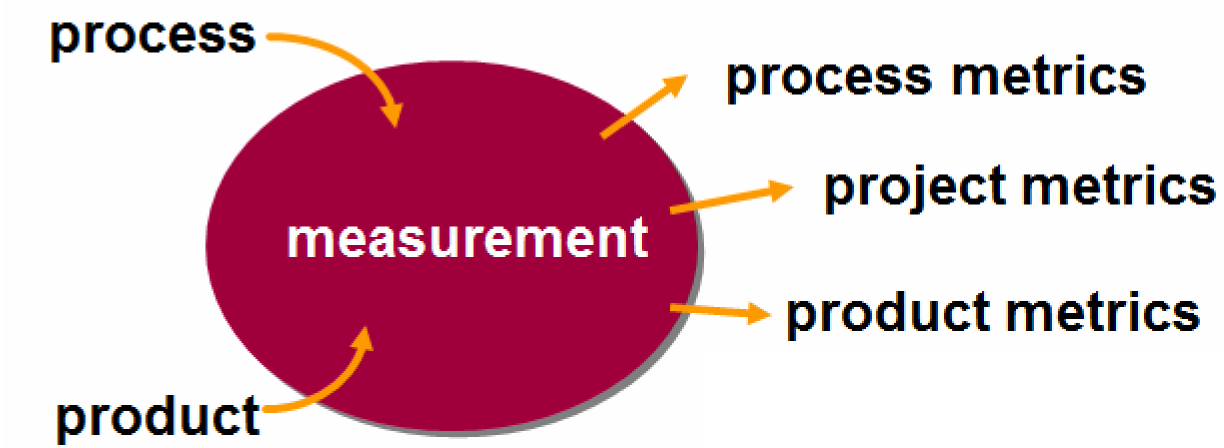
Software Process Improvement

- **Software Process Improvement (SPI)** are actions taken to **change the processes of an organization so that they achieve more effectively their business goals**.
- Software process improvement models emphasize iterative cycles of continuous improvement.
- **A software process improvement cycle typically involves the subprocesses of measuring, analyzing, and changing.** [SWEBOK 3]



Software Engineering Measurement

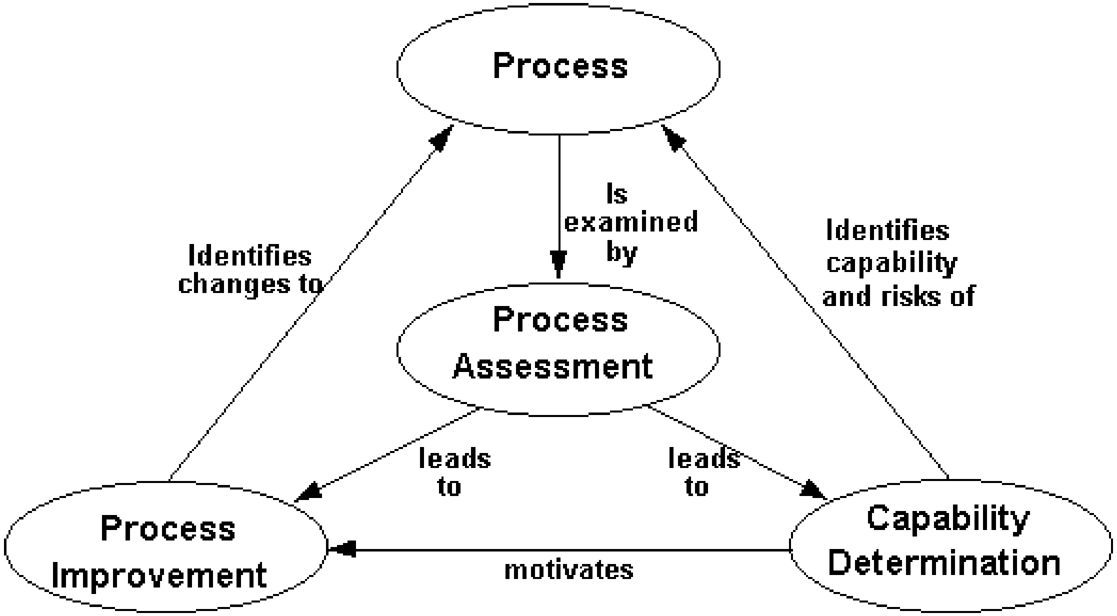
- The importance of measurement and its **role in better management and engineering practices**
- Effective measurement has become one of the cornerstones of organizational maturity.
- Measurement can be applied to **organizations, projects, processes, and work products**



Software Process Improvement

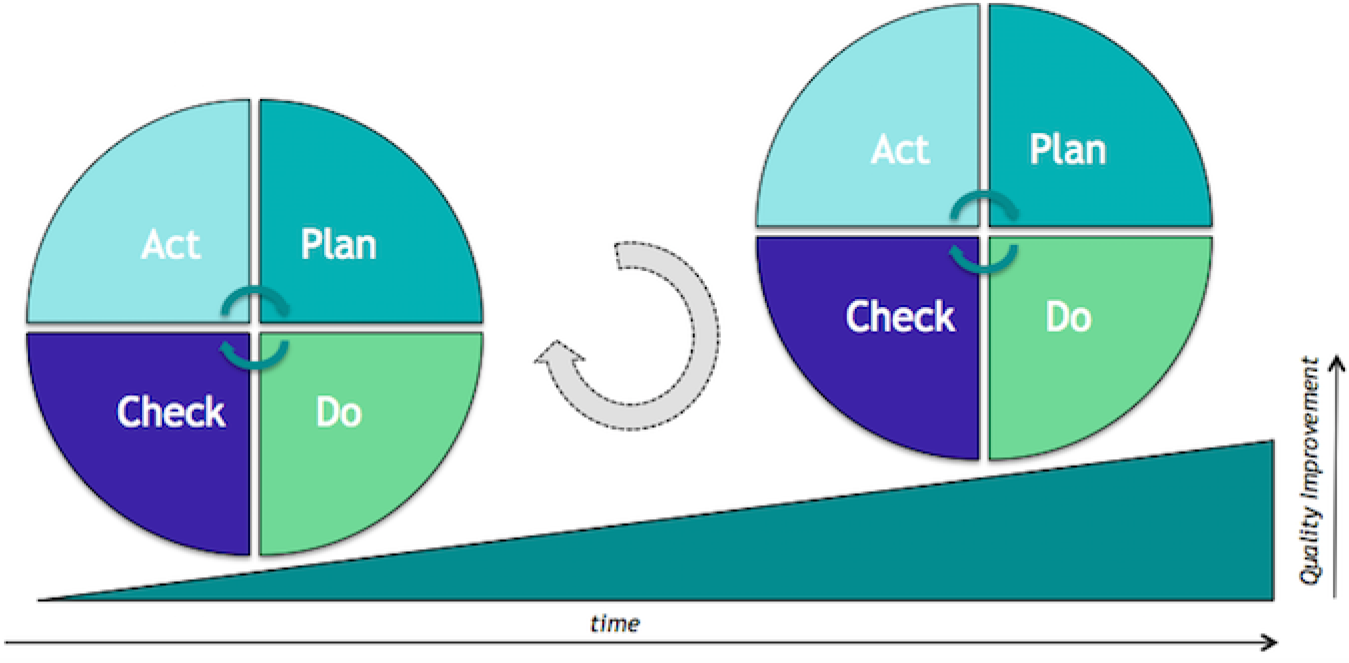
- Typically, software process and assessment are guided by:
  - **a maturity level**
  - **a process capability**

  profile based on capability/maturity model(s).



PDCA
Plan-Do-Check-Act

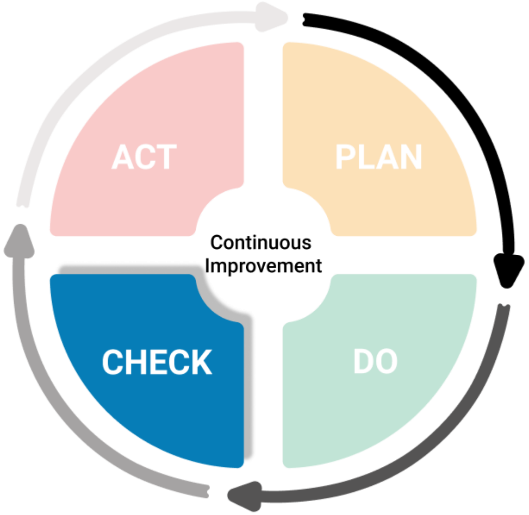
- ISO/IEC 27001



https://resources.sei.cmu.edu/asset_files/whitepaper/2013_019_001_297130.pdf

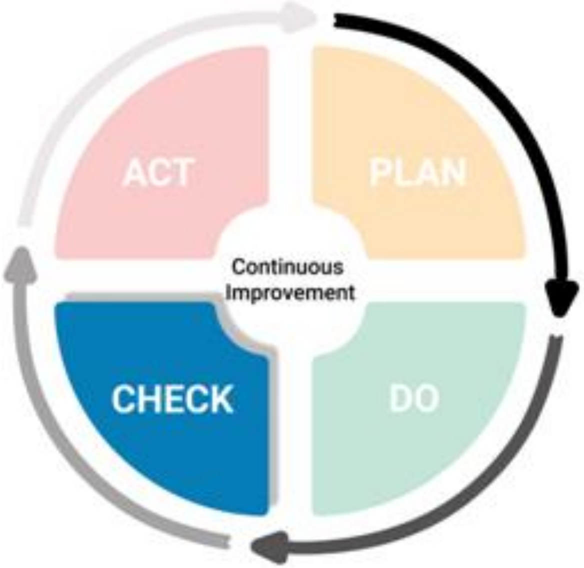
Software process assessments

- Software process assessments are used to evaluate **the form and content of a software process**, which may **be specified by a standardized set of criteria**



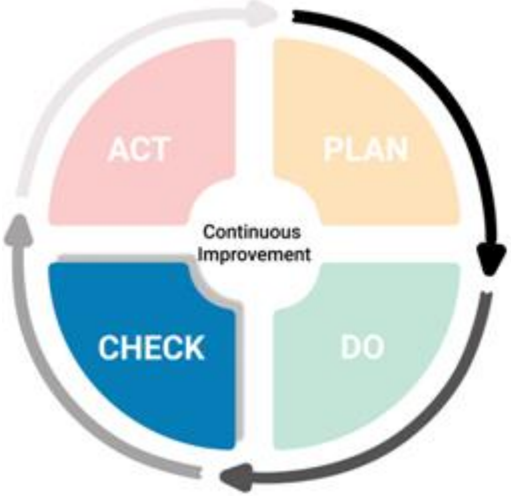
Plan-Do-Check-Act

- Planing:
  - Improvement activities include **identifying and prioritizing desired improvements** ;
- Doing
  - introducing an improvement, including **change management and training**



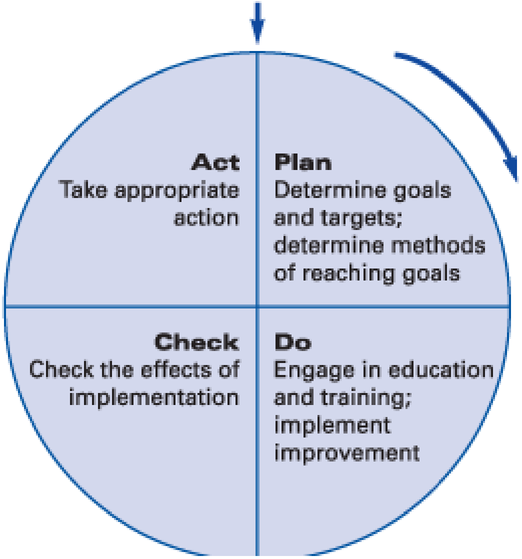
Plan-Do-Check-Act

- Checking
  - **evaluating the improvement** as compared to previous or exemplary process results and costs
- Acting:
  - **making further modifications**



Plan-Do-Check-Act

A Common Version of the PDCA Wheel

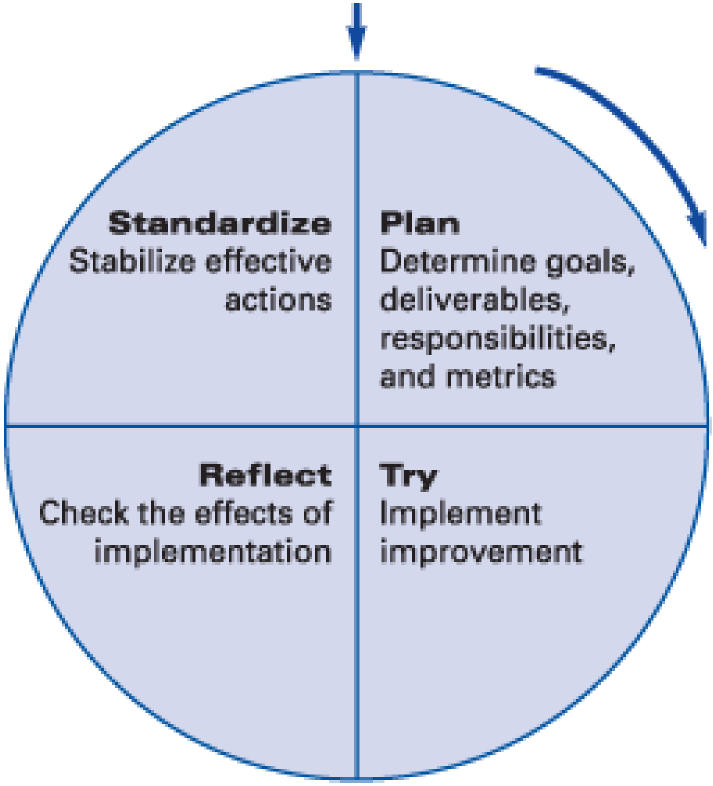


Act
Take appropriate action

Plan
Determine goals and targets; determine methods of reaching goals

Check
Check the effects of implementation

Do
Engage in education and training; implement improvement



Standardize
Stabilize effective actions

Plan
Determine goals, deliverables, responsibilities, and metrics

Reflect
Check the effects of implementation

Try
Implement improvement

https://www.lean.org/lexicon-terms/plan-do-check-act/

Maturity

- Maturity levels apply to your organization’s process **improvement achievement in multiple process areas**.

- These levels are a means of improving the processes corresponding to a given **set of process areas** (i.e., maturity level).
  - The five maturity levels are numbered 1 through 5.

Branda – organizacijos charakteristika, nusakanti, kiek organizacijos procesas yra apibrėžtas, valdomas, matuojamas, kontroliuojamas ir nuolatos gerinamas

Proceso branda yra gebėjimas sujungti žmogiškuosius išteklius, metodus, procedūras ir įrankius tam, kad būtų patenkinti kliento poreikiai (Poulin, 2006)

Proceso brandos lygiai (CMMI)

- 0. Neegzistuojantis procesas
  - Nėra sistemišką veikimą pagrindžiančių įrodymų
- 1. Pradinis procesas (Initial)
  - Chaotiškas – neprognozuojamos kainos, terminai ir kokybė
- 2. Atkartojamas procesas (Repeatable)
  - Intuityvus – kaina ir kokybė smarkiai varijuoja, pakankamas terminų valdymas, neformalios procedūros
- 3. Apibrėžtas procesas (Defined)
  - Kokybinis – patikimai nustatomos kainos ir terminai, gerėjanti, bet neprognozuojama kokybė
- 4. Kiekybiškai valdomas procesas (Quantitatively Managed)
  - Kiekybinis – pakankama statistinė produktų kokybės kontrolė
- 5. Nuolatos gerinamas procesas (Optimizing)
  - Kiekybinės metrikos naudojamos nuolatiniam proceso automatizavimo plėtimui ir proceso gerinimui

Maturity

- **Maturity levels represent a staged path for an organization's performance and process improvement** efforts based on predefined sets of practice areas.

- **Maturity:**
  - **Processes are documented**
  - **Assigned responsibilities**
  - **Processes are managed**
  - **Processes are structured**
  - **Customer-centric processes**

- Each maturity level builds on the previous maturity levels by adding new functionality or rigor.

Maturity

- **Maturity levels represent a staged path for an organization's performance and process improvement** efforts based on predefined sets of practice areas.

- **Maturity:**
  - **Processes are documented**
  - **Assigned responsibilities**
  - **Processes are managed**
  - **Processes are structured**
  - **Customer-centric processes**

- Each maturity level builds on the previous maturity levels by adding new functionality or rigor.

Capability

- Capability levels apply to your organization’s process **improvement achievement in individual process areas.**
  - The four capability levels are numbered 0 through 3.

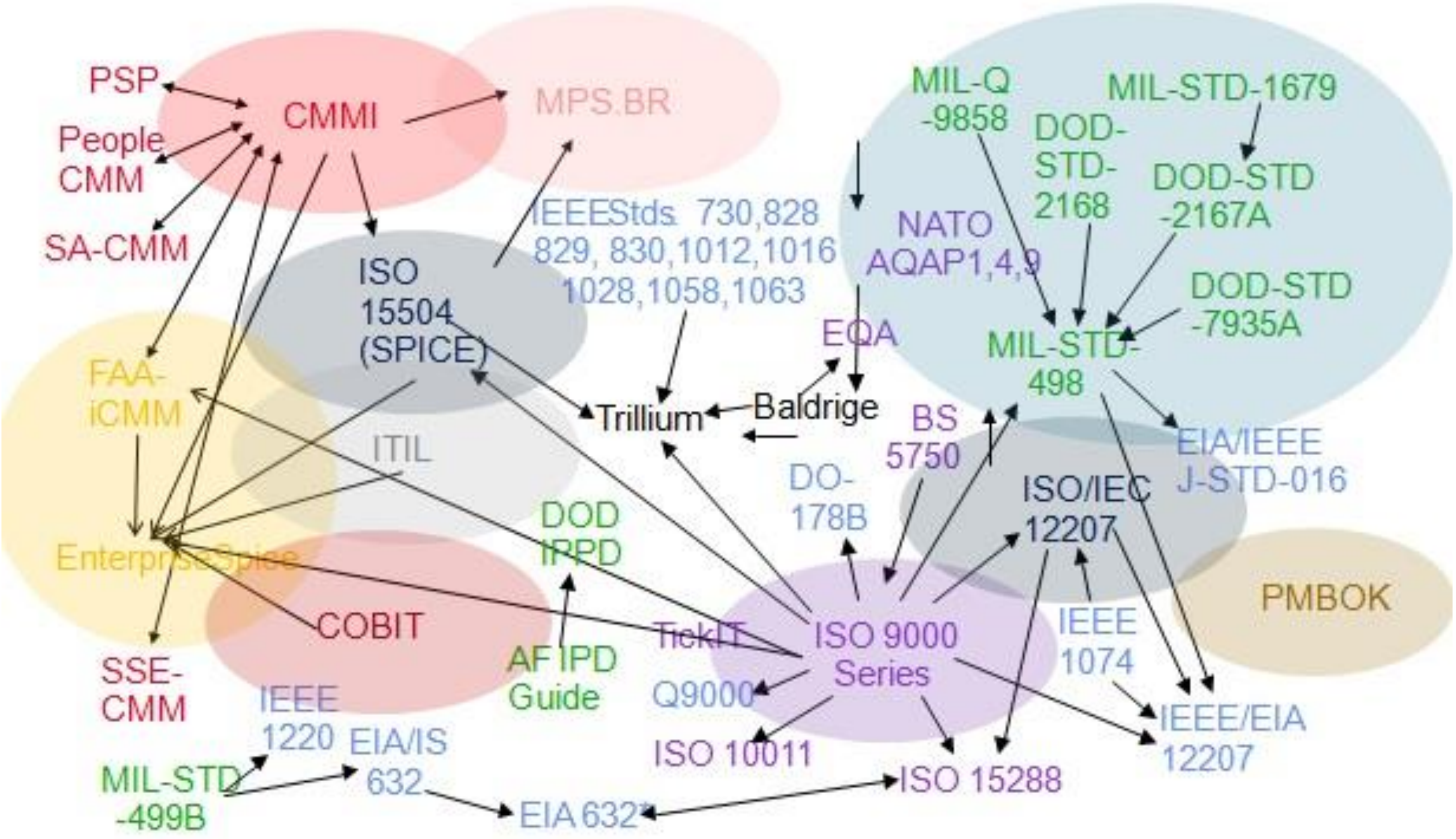
  Proceso gebėjimas yra **skirtumo tarp užsibrėžtų tikslų ir realių rezultatų kontrolė,** gebėjimas nustatyti, ar bus pasiekti užsibrėžti tikslai ir užtikrintas efektyvumas (Lockamy, McCormack, 2004)

  Gebėjimas – proceso charakteristika, nusakanti laukiamų rezultatų, kuriuos galima gauti taikant tą procesą, pasiskirstymą

Capability

- **Capability**:
  - **Team have the necessary competencies**
  - **Responsibilities are assigned to the process**
  - **The results of the process are controlled**
  - **The activities of the organization are effective**

Capability/Maturity models

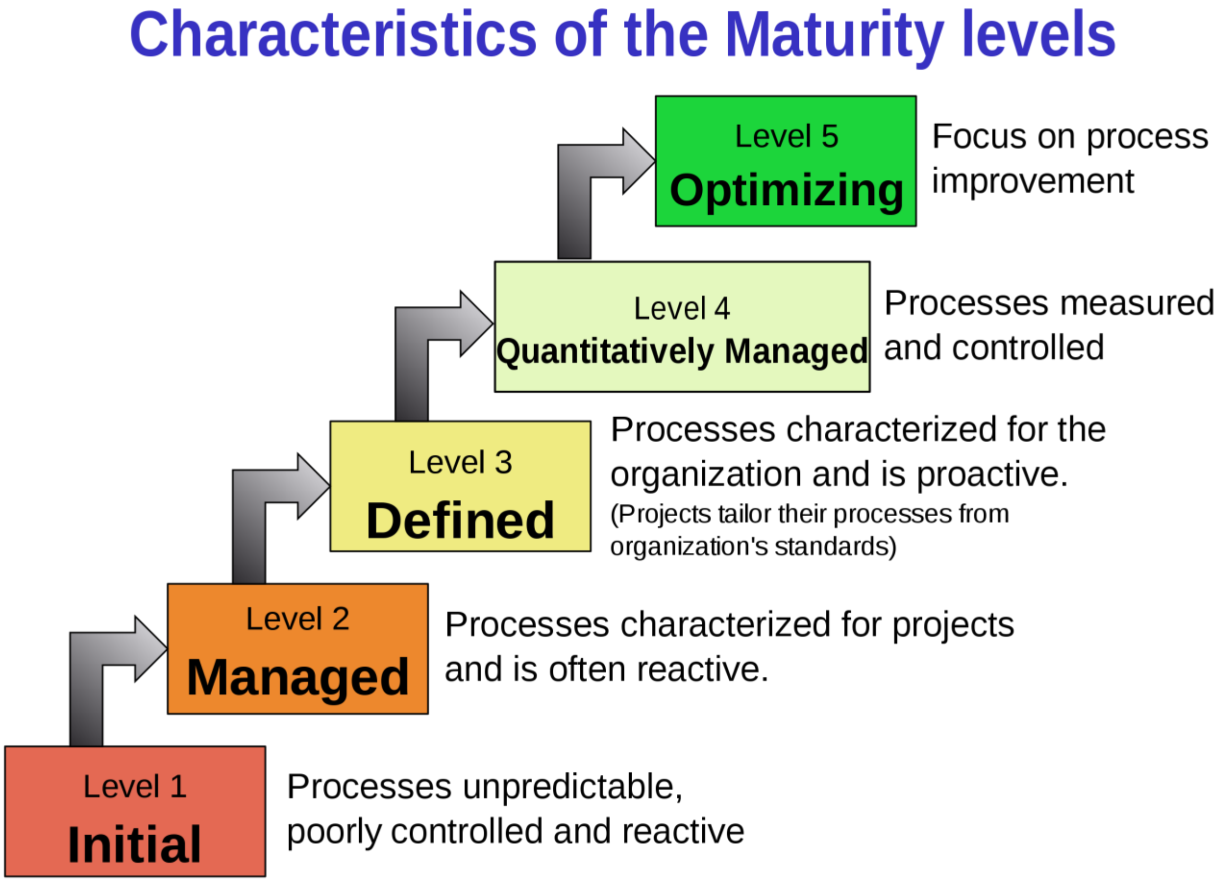


Adapted from [S. A. Sheard, Evolution of the Frameworks Quagmire. IEEE Computer, July 2001]

http://www.gqs.ufsc.br/software-process-improvement/

Maturity models

- CMM - **Capability Maturity Model**



Characteristics of the Maturity levels

Level 5
Optimizing
Focus on process improvement

Level 4
Quantitatively Managed
Processes measured and controlled

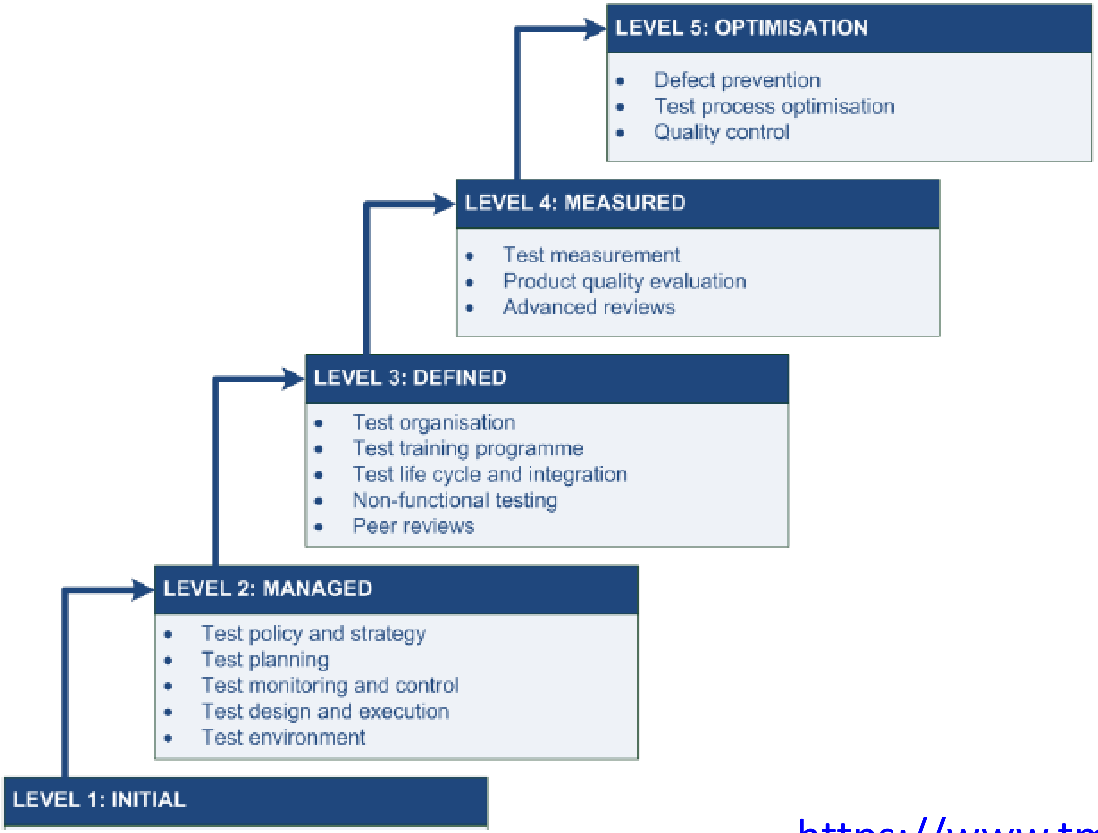
Level 3
Defined
Processes characterized for the organization and is proactive.
(Projects tailor their processes from organization's standards)

Level 2
Managed
Processes characterized for projects and is often reactive.

Level 1
Initial
Processes unpredictable, poorly controlled and reactive

Maturity models

- TMM **Test Maturity Model**



LEVEL 5: OPTIMISATION

- Defect prevention
- Test process optimisation
- Quality control

LEVEL 4: MEASURED

- Test measurement
- Product quality evaluation
- Advanced reviews

LEVEL 3: DEFINED

- Test organisation
- Test training programme
- Test life cycle and integration
- Non-functional testing
- Peer reviews

LEVEL 2: MANAGED

- Test policy and strategy
- Test planning
- Test monitoring and control
- Test design and execution
- Test environment

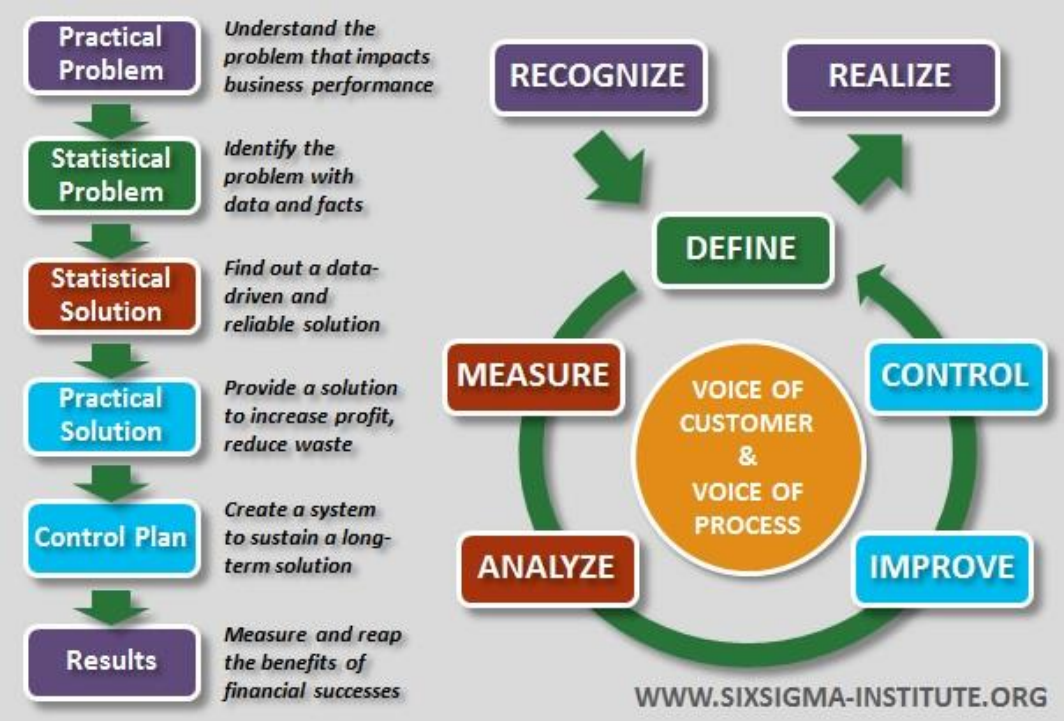
LEVEL 1: INITIAL

Figure 1 - TMMi model.

https://www.tmmi.org/tmmi-model/

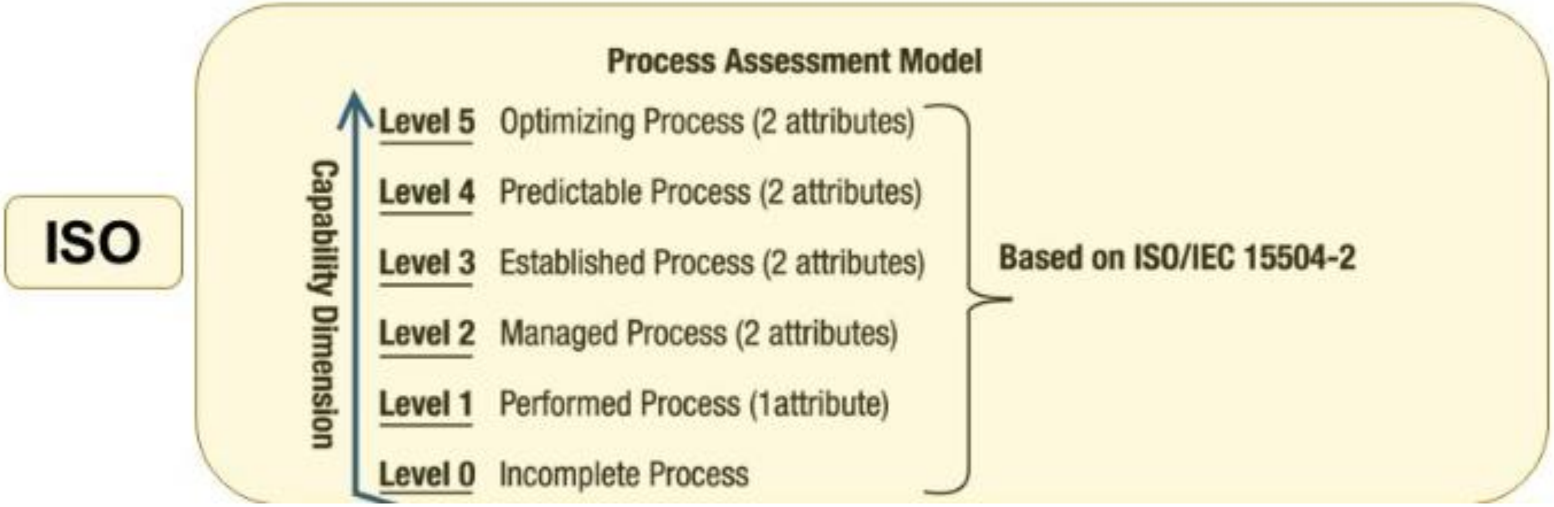
Maturity models

- SixSigma
  - *Six Sigma is a* ***quality program*** *that, when all is said and done,* ***improves your customer's experience, lowers your costs, and builds better leaders****.* — Jack Welch


Practical Problem
Understand the problem that impacts business performance
Statistical Problem
Identify the problem with data and facts
Statistical Solution
Find out a data-driven and reliable solution
Practical Solution
Provide a solution to increase profit, reduce waste
Control Plan
Create a system to sustain a long-term solution
Results
Measure and reap the benefits of financial successes
RECOGNIZE
REALIZE
DEFINE
MEASURE
CONTROL
VOICE OF CUSTOMER & VOICE OF PROCESS
ANALYZE
IMPROVE
WWW.SIXSIGMA-INSTITUTE.ORG

Maturity models

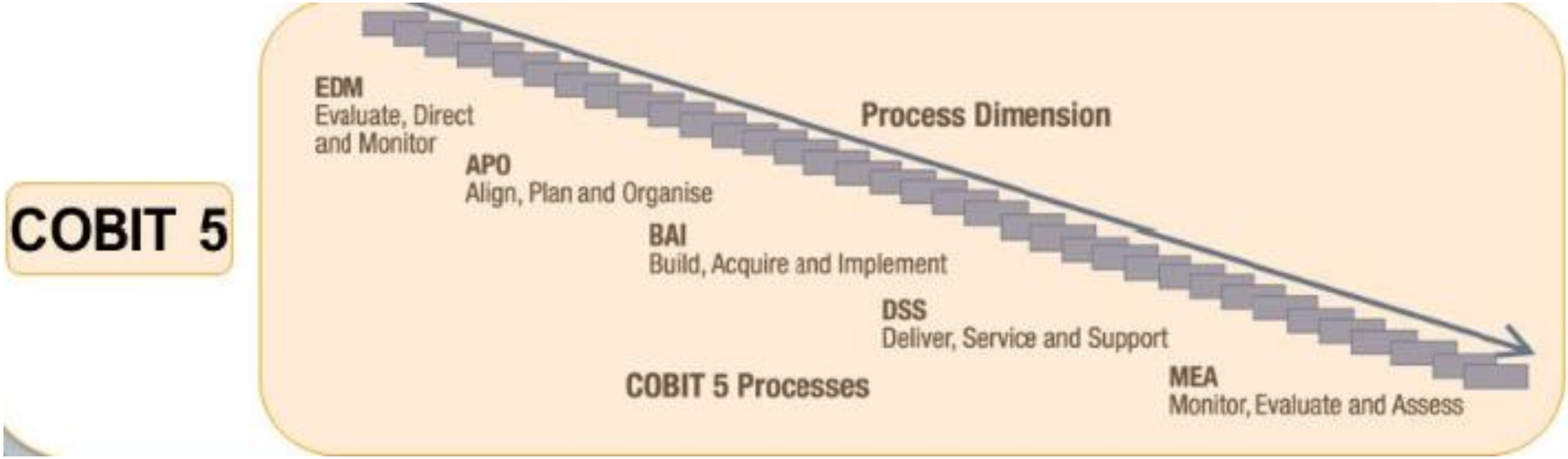
- COBIT (**Control Objectives for Information and Related Technologies**) is a framework created by ISACA for information technology (IT) management and IT governance.



ISO
Process Assessment Model
Capability Dimension
Level 5 Optimizing Process (2 attributes)
Level 4 Predictable Process (2 attributes)
Level 3 Established Process (2 attributes)
Level 2 Managed Process (2 attributes)
Level 1 Performed Process (1attribute)
Level 0 Incomplete Process
Based on ISO/IEC 15504-2

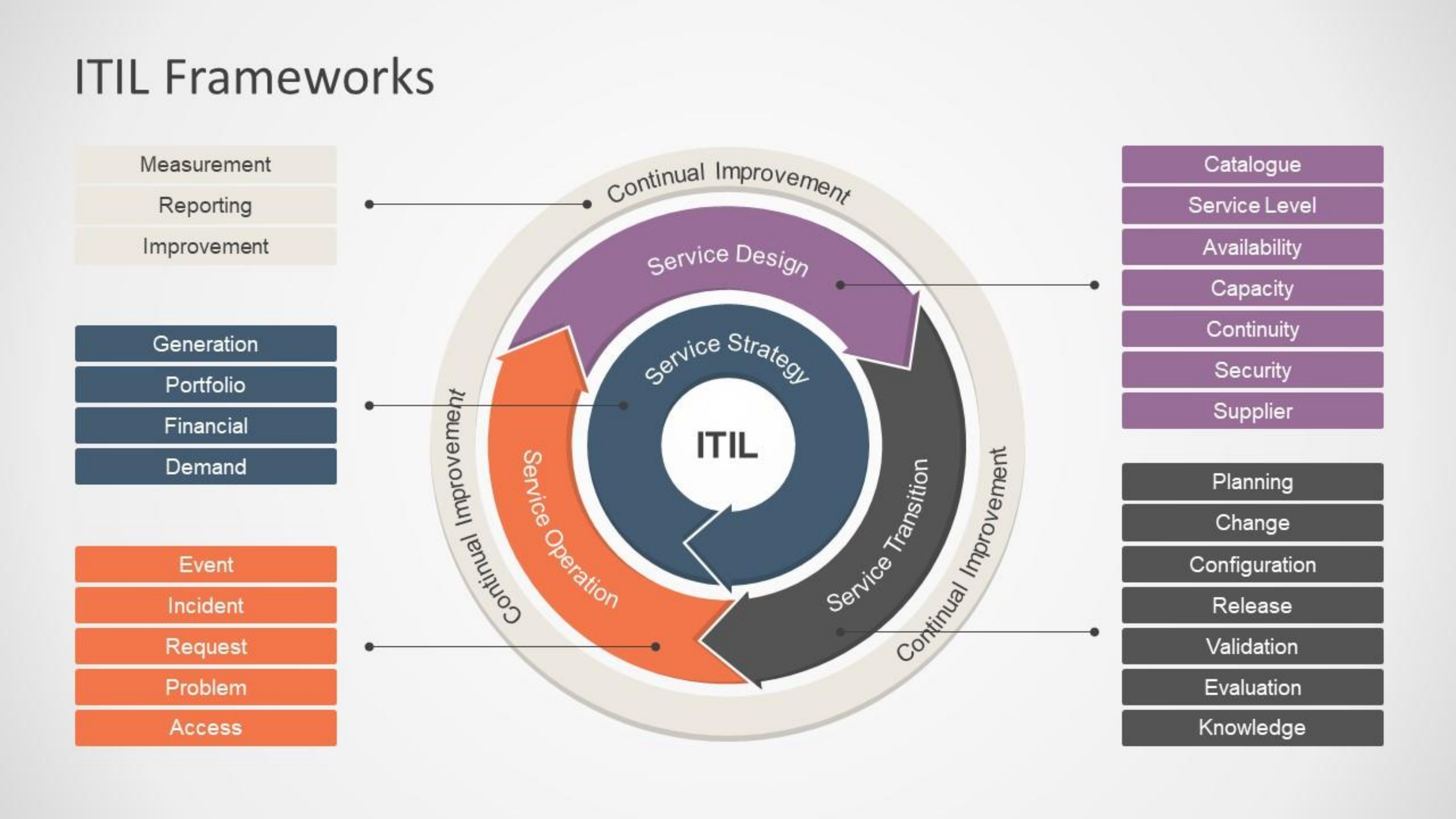
Maturity models

- The framework defines a set of generic processes **for the management of IT,** with each process defined together with ***process inputs and outputs, key process-activities, process objectives, performance measures and an elementary maturity model***.


COBIT 5
EDM
Evaluate, Direct and Monitor
APO
Align, Plan and Organise
BAI
Build, Acquire and Implement
DSS
Deliver, Service and Support
MEA
Monitor, Evaluate and Assess
Process Dimension
COBIT 5 Processes

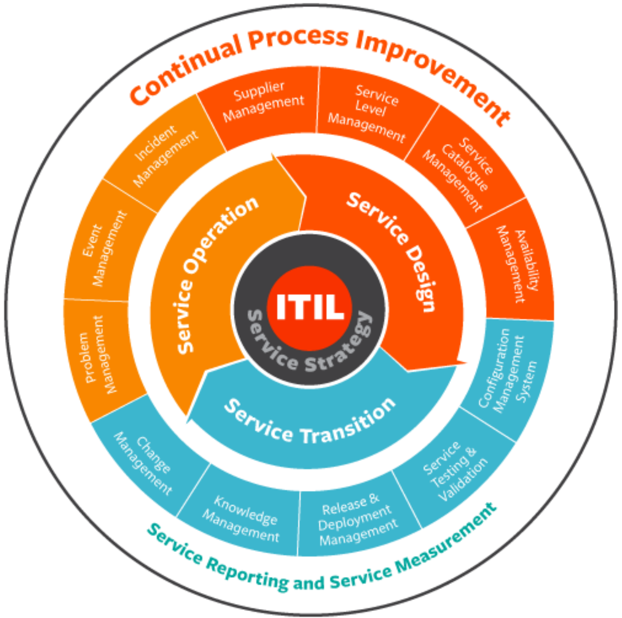
Maturity models

- ITIL **Information Technology Infrastructure Library -** Maturity model and self-assessment service has been developed to help organizations **improve their IT service management** within the ITIL framework.



Maturity models

- ITIL describes processes, procedures, tasks, and checklists which are not organization-specific nor technology-specific, but can be applied by an organization toward strategy, delivering value, and maintaining a minimum level of competency.
- It allows the organization to establish a baseline from which it **can plan, implement, and measure.**



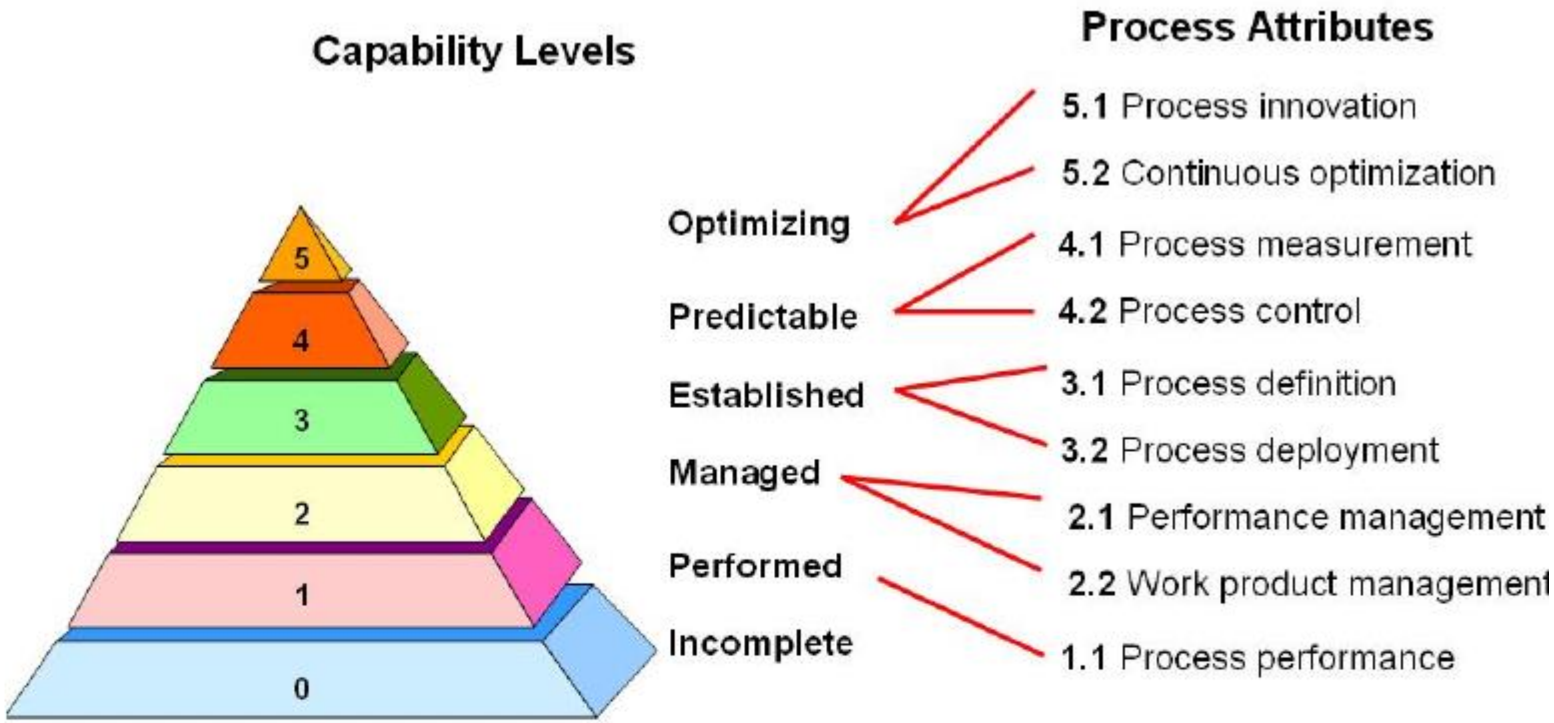
Maturity models

- **ISO/IEC 15504** *Information technology – Process assessment*, also termed **Software Process Improvement and Capability Determination** (**SPICE**),
- Is a set of technical standards documents for the **computer software development process and related business management functions.**

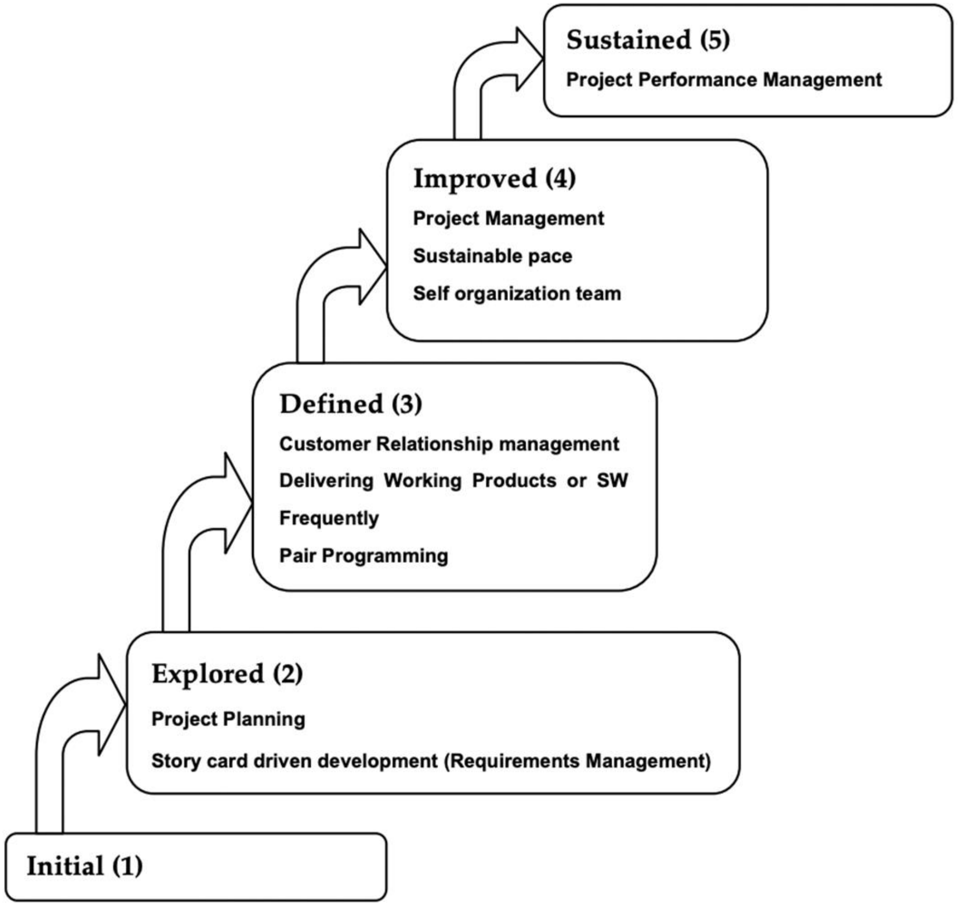
- *It is one of the joint International Organization for Standardization (ISO) and International Electrotechnical Commission (IEC) standards, which was developed by the ISO and IEC joint subcommittee, ISO/IEC JTC 1/SC 7.*

Maturity models

- **ISO/IEC 15504**



Agile Maturity Model (Pathel and Ramachandran, 2009)



Sustained (5)

Project Performance Management

Improved (4)

Project Management

Sustainable pace

Self organization team

Defined (3)

Customer Relationship management

Delivering Working Products or SW Frequently

Pair Programming

Explored (2)

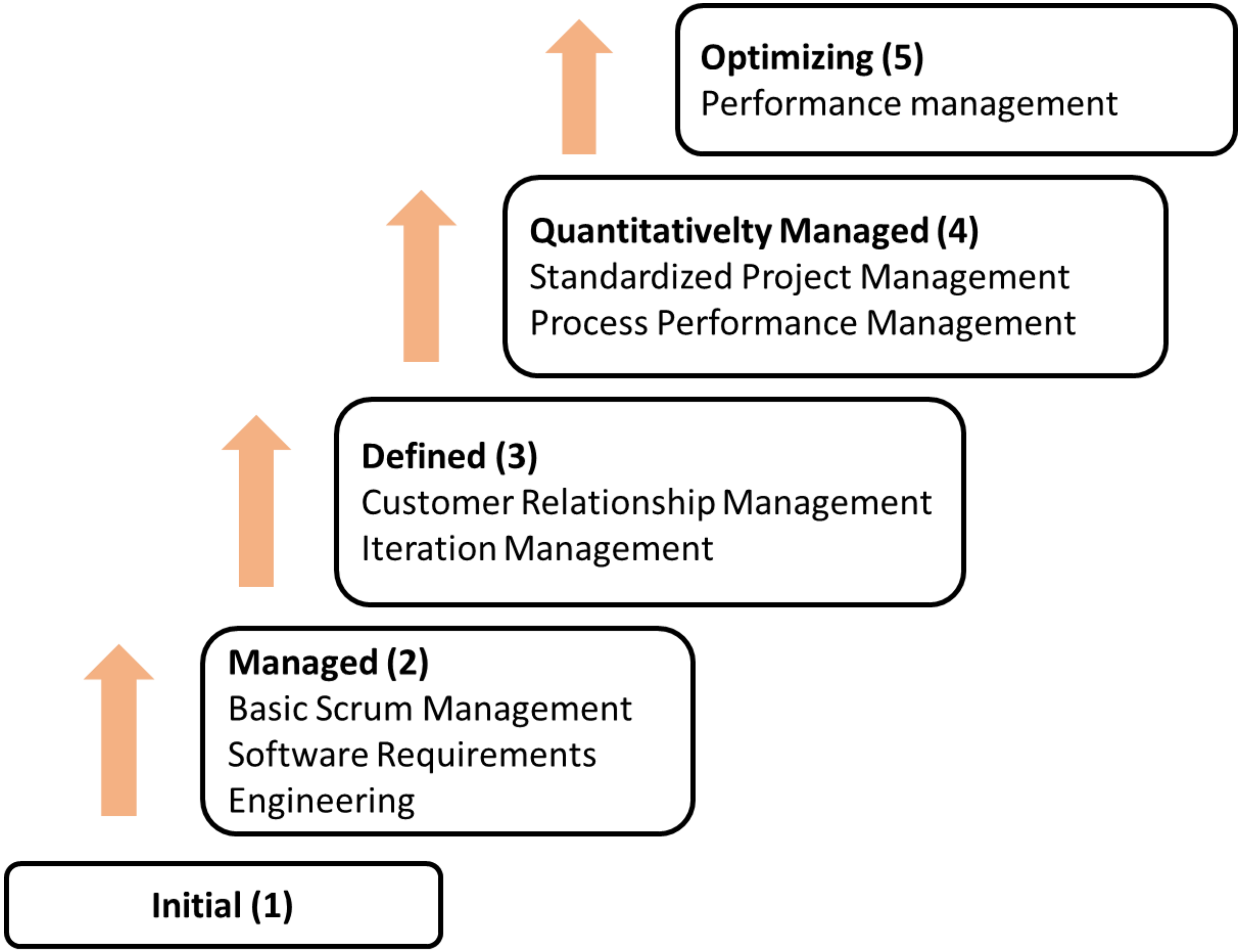
Project Planning

Story card driven development (Requirements Management)

Initial (1)

http://ijse.org.eg/papers/agile-maturity-model-amm-a-software-process-improvement-framework-foragile-software-development-practices/

Scrum Maturity Model (Yin, 2011)



Optimizing (5)
Performance management

Quantitativelty Managed (4)
Standardized Project Management
Process Performance Management

Defined (3)
Customer Relationship Management
Iteration Management

Managed (2)
Basic Scrum Management
Software Requirements Engineering

Initial (1)

Capability Maturity Model - CMM

- CMMI (angl. Capability Maturity Model Integration).

- Department of Defense decided in the 80s to do something about the **many problems in its expensive software projects**

- CMMI models have expanded beyond software engineering to **help any organization in any industry build, improve, and measure their capabilities and improve performance.**

- CMMI is a collection of BEST PRACTICES.

The CMMI describes WHAT to do,
not HOW to do it.

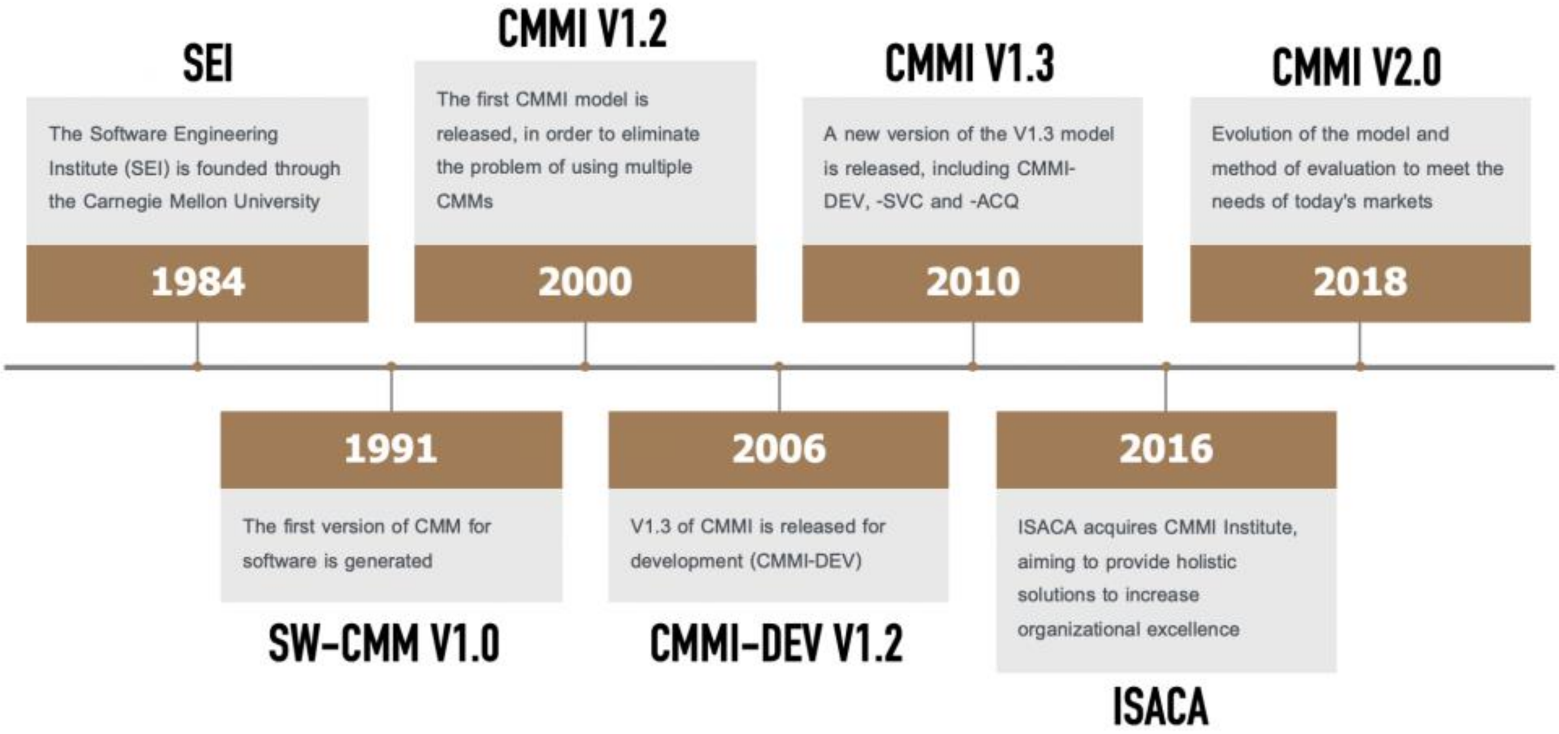
What is a CMM?

- Capability Maturity Model:
  - A reference model of **mature practices** in a specified discipline, used to assess a **group's capability to perform that discipline**
- NOT:
  - It is not a ready-made scheme or template for describing processes
  - It contains no methods for the processes
- *The CMMI model is very complex, with more than 1,000 pages of description.*

The CMM Explosion

- The first CMM (CMM v1.0) was developed for software and released in August 1991
- **Based on this success and the demand from other interests CMMs were developed for other disciplines and functions:**
    - **Systems Engineering**
    - **People**
    - **Integrated Product Development**
    - **Software Acquisition**
    - **Software Quality Assurance**
    - **Measurement**
    - **Others…….**

History of CMMI



SEI

1984

The Software Engineering Institute (SEI) is founded through the Carnegie Mellon University

SW-CMM V1.0

1991

The first version of CMM for software is generated

CMMI V1.2

2000

The first CMMI model is released, in order to eliminate the problem of using multiple CMMs

CMMI-DEV V1.2

2006

V1.3 of CMMI is released for development (CMMI-DEV)

CMMI V1.3

2010

A new version of the V1.3 model is released, including CMMI-DEV, -SVC and -ACQ

ISACA

2016

ISACA acquires CMMI Institute, aiming to provide holistic solutions to increase organizational excellence

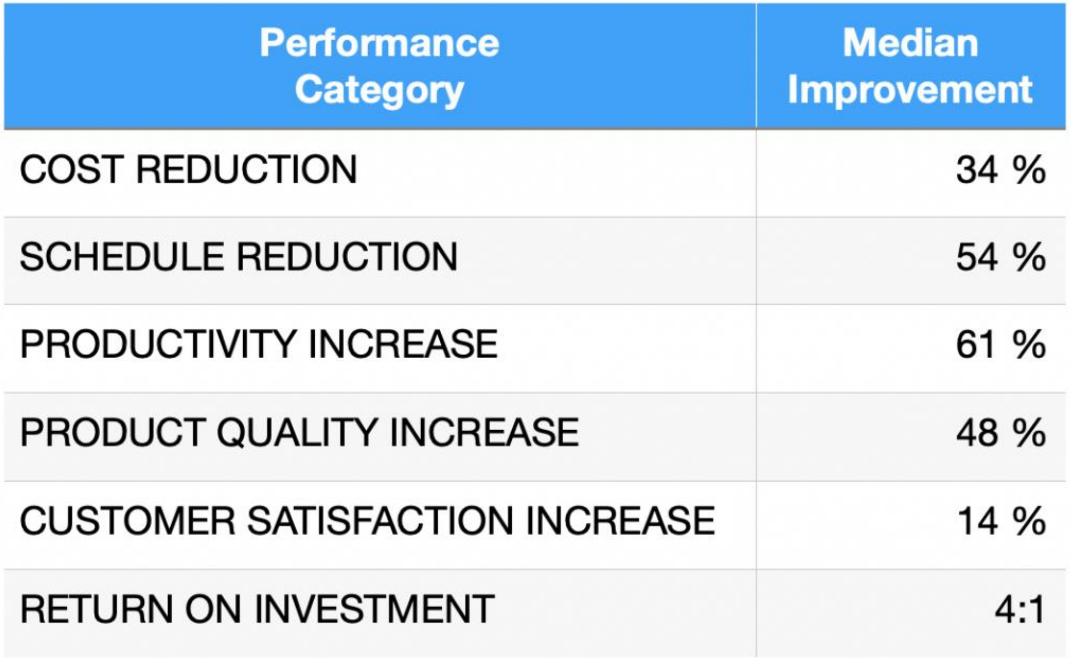
CMMI V2.0

2018

Evolution of the model and method of evaluation to meet the needs of today's markets

Software Process Improvement : CMMI

- Software Engineering Institute collects quantitative measures of CMMI performance improvement:



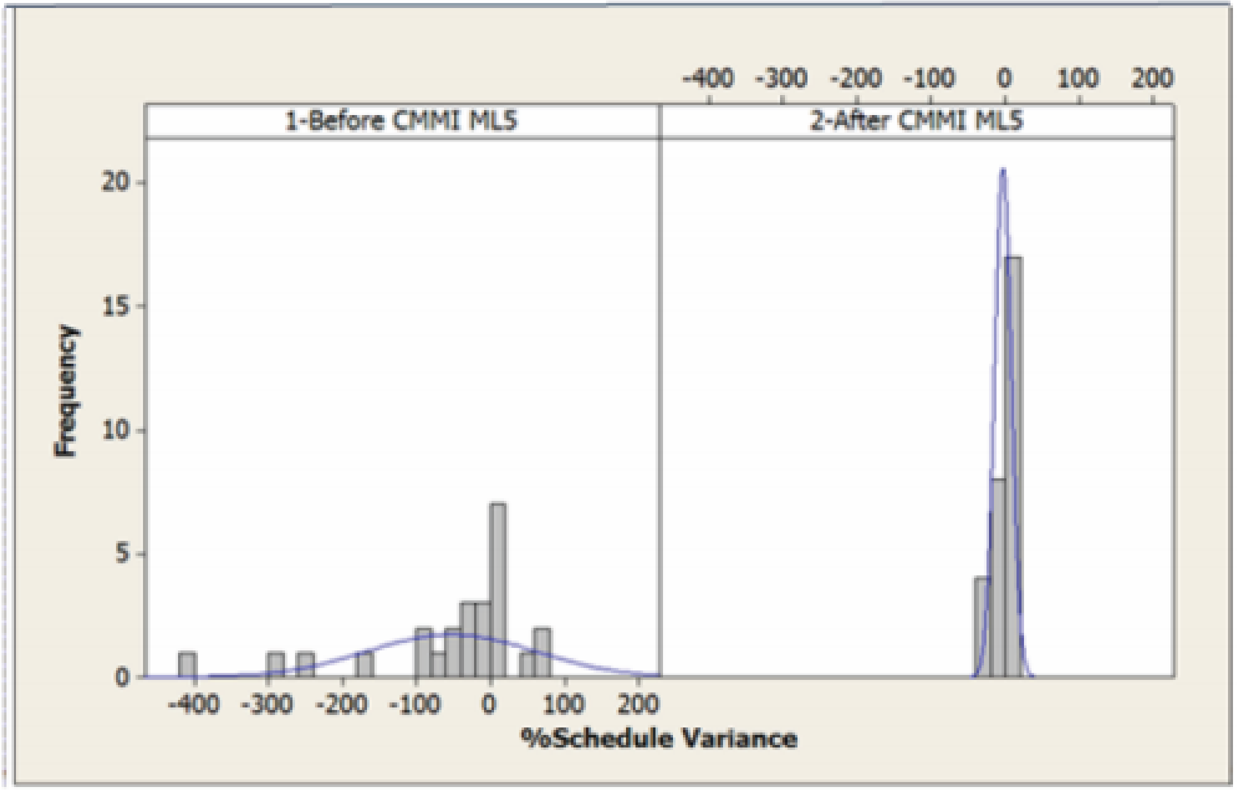
Performance Category	Median Improvement
COST REDUCTION	34 %
SCHEDULE REDUCTION	54 %
PRODUCTIVITY INCREASE	61 %
PRODUCT QUALITY INCREASE	48 %
CUSTOMER SATISFACTION INCREASE	14 %
RETURN ON INVESTMENT	4:1

https://www.sei.cmu.edu/

Proven results of CMMI (John Maher, 2016)

Benefits from Using CMMI

- Reduce schedule variance



https://www.sei.cmu.edu/

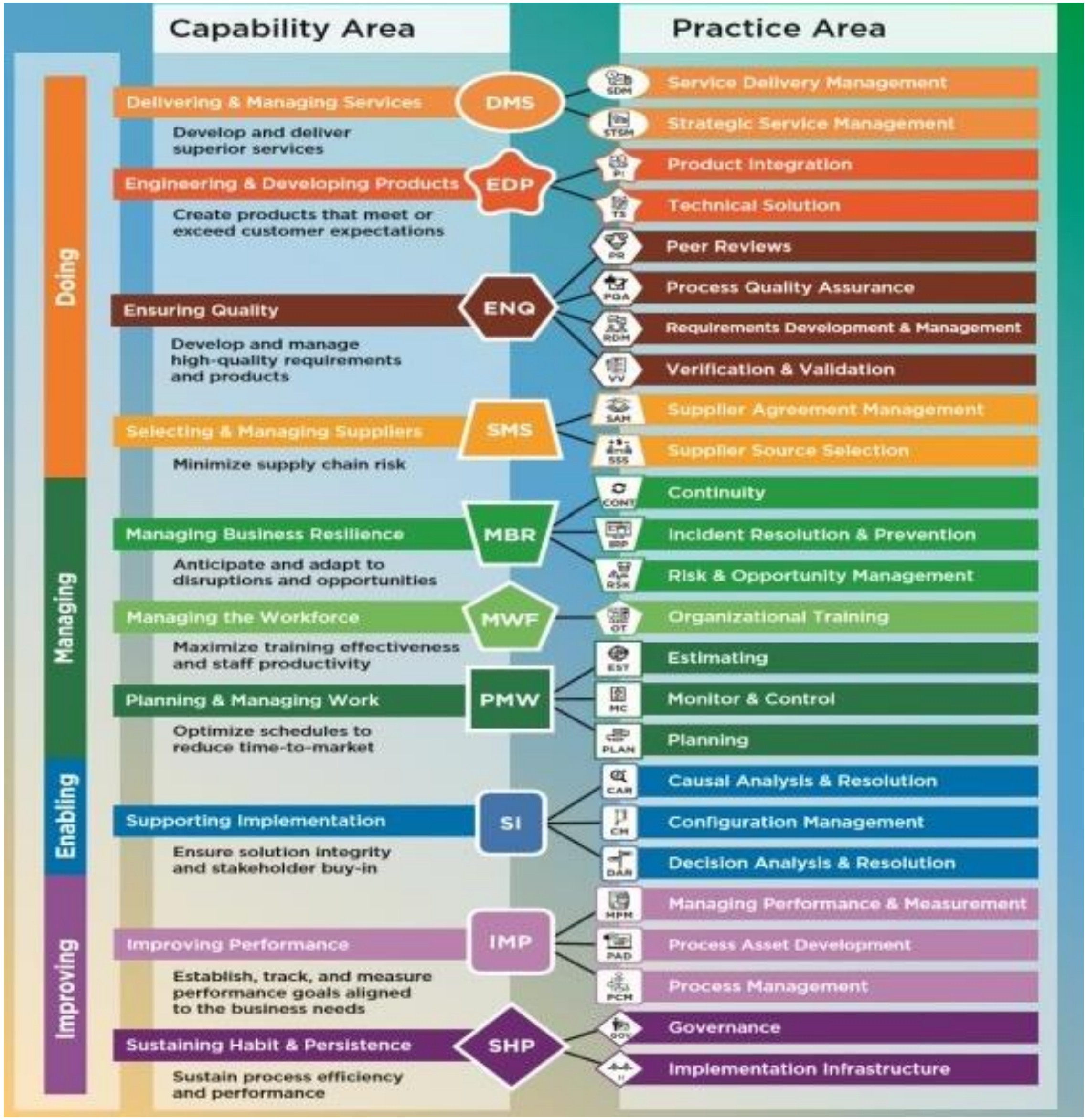
Benefits from Using CMMI

1. Better customer satisfaction
2. Increased quality
3. More accurate schedules
4. Lower development costs
5. Substantial return on investment
6. Improved employee morale and reduced turnover

Benefits from Using CMMI

- Organization's activities are explicitly **linked to its business objectives.**
- **Visibility into the organization's** activities is increased to help to ensure that the product or service **meets the customer's expectations.**
- The teams learn from **new areas of best practice** (e.g., measurement, risk)

- *CMMI is being adopted worldwide, including North America, Europe, Asia, Australia, South America, and Africa.*



Capability Area

Practice Area

Doing

Delivering & Managing Services — DMS

Develop and deliver superior services

- Service Delivery Management
- Strategic Service Management

Engineering & Developing Products — EDP

Create products that meet or exceed customer expectations

- Product Integration
- Technical Solution

Ensuring Quality — ENQ

Develop and manage high-quality requirements and products

- Peer Reviews
- Process Quality Assurance
- Requirements Development & Management
- Verification & Validation

Selecting & Managing Suppliers — SMS

Minimize supply chain risk

- Supplier Agreement Management
- Supplier Source Selection

Managing

Managing Business Resilience — MBR

Anticipate and adapt to disruptions and opportunities

- Continuity
- Incident Resolution & Prevention
- Risk & Opportunity Management

Managing the Workforce — MWF

Maximize training effectiveness and staff productivity

- Organizational Training

Planning & Managing Work — PMW

Optimize schedules to reduce time-to-market

- Estimating
- Monitor & Control
- Planning

Enabling

Supporting Implementation — SI

Ensure solution integrity and stakeholder buy-in

- Causal Analysis & Resolution
- Configuration Management
- Decision Analysis & Resolution

Improving

Improving Performance — IMP

Establish, track, and measure performance goals aligned to the business needs

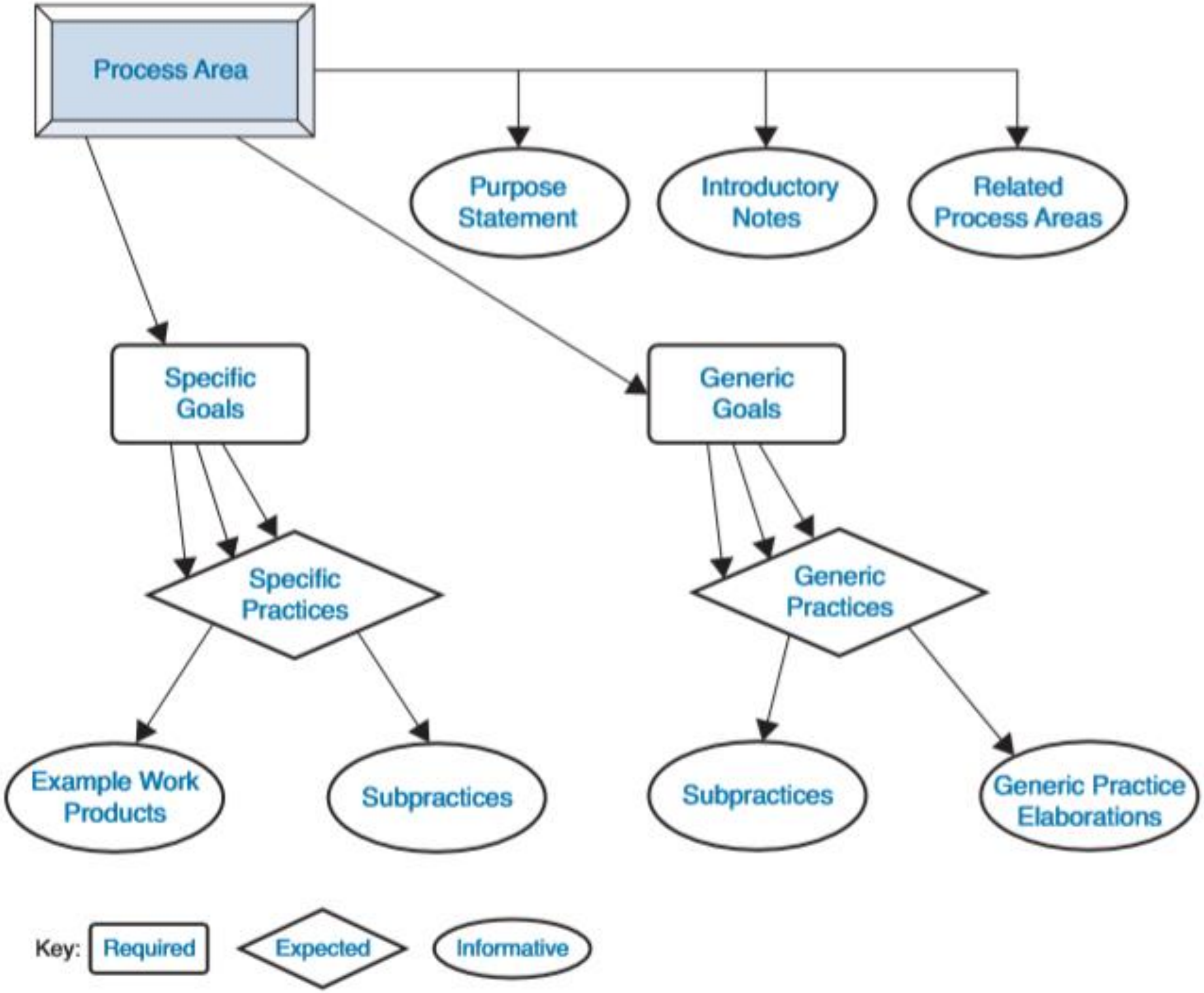
- Managing Performance & Measurement
- Process Asset Development
- Process Management

Sustaining Habit & Persistence — SHP

Sustain process efficiency and performance

- Governance
- Implementation Infrastructure

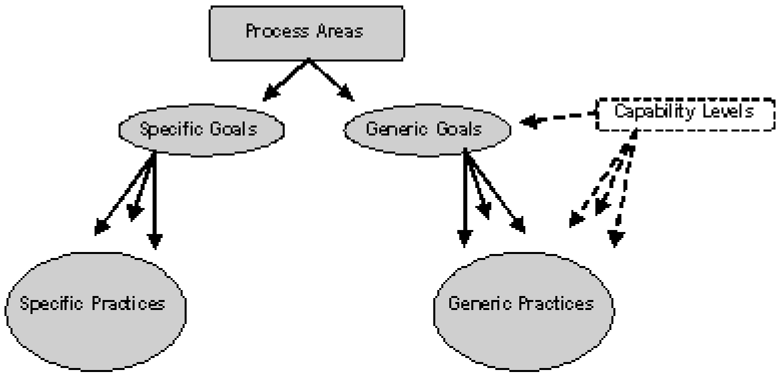
CMMI Model Components



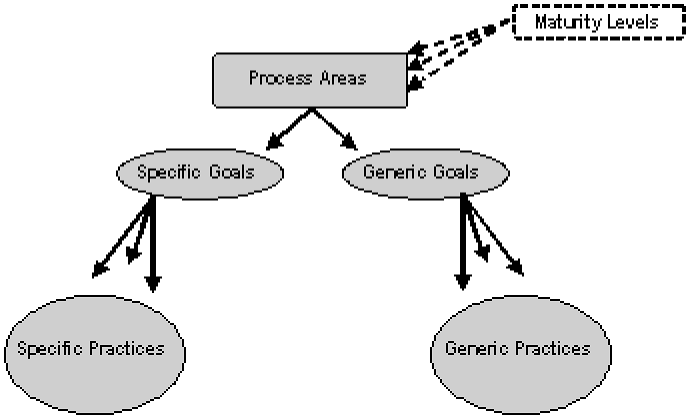
CMMI - DEV

- Continues architecture Tolydinė architektūra
- Staged architecture Pakopinė architektūra

Continuous Representation

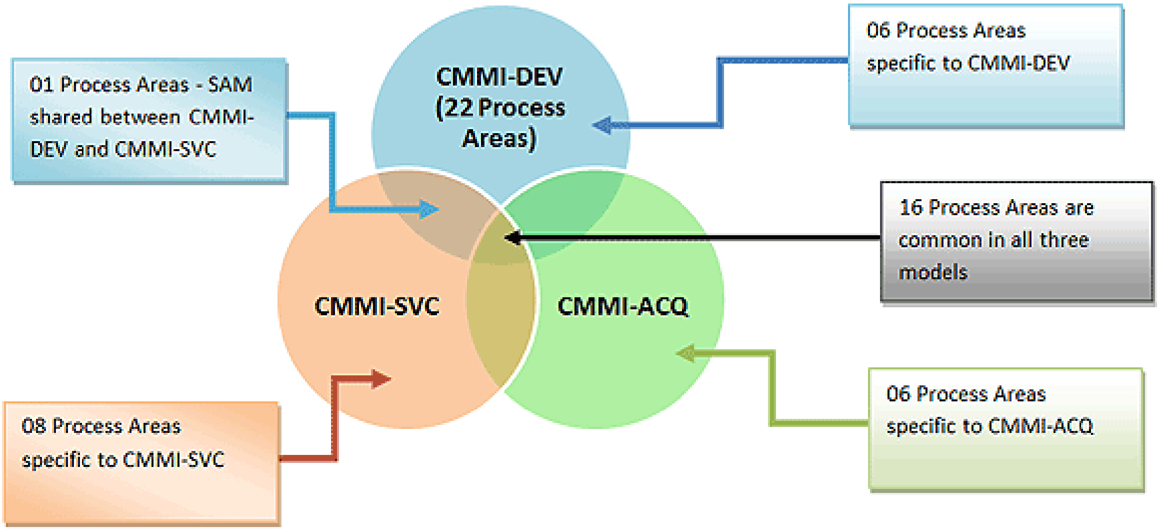


Staged Representation



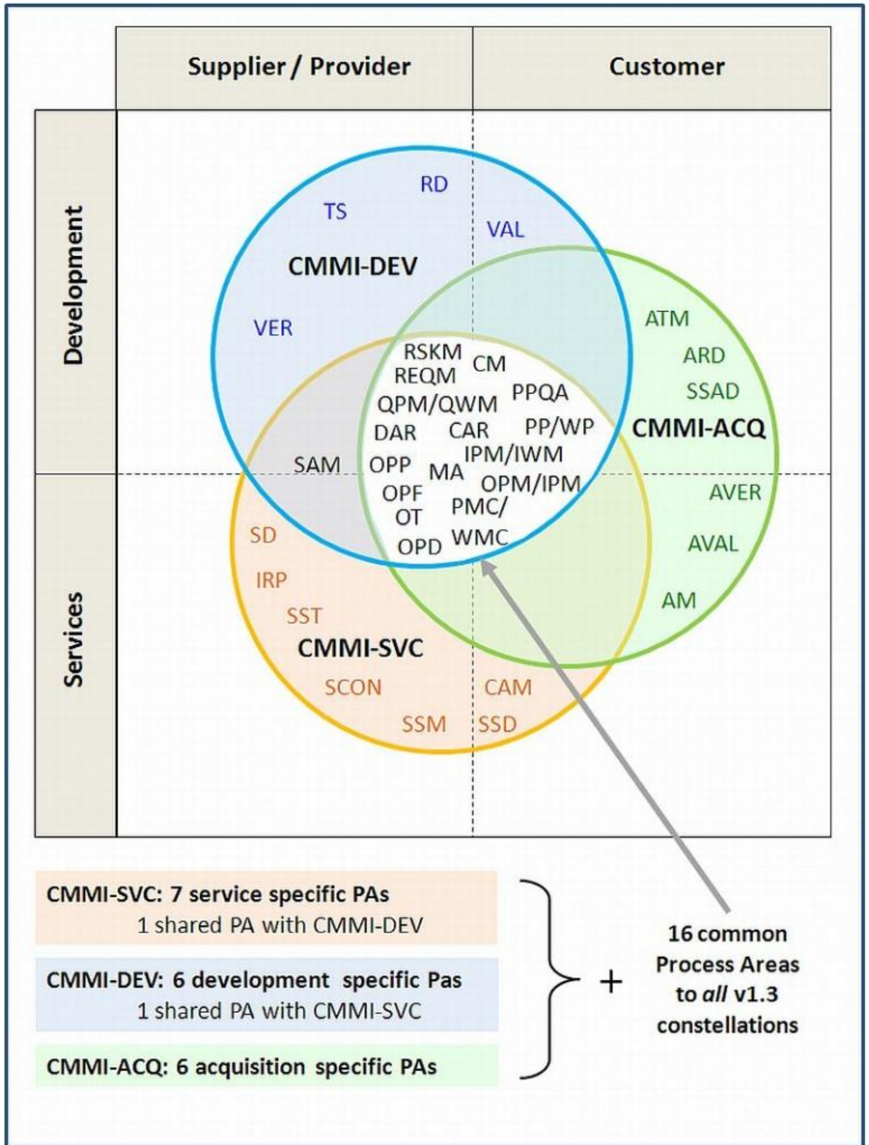
CMMI – DEV Staged architecture

- CMMI for Development (CMMI-DEV
  - It addresses product and service development processes.
- CMMI for Acquisition (CMMI-ACQ),
  - It addresses supply chain management, acquisition, and outsourcing processes in government and industry.
- CMMI for Services (CMMI-SVC),
  - It addresses guidance for delivering services within an organization and to external customers.



01 Process Areas - SAM shared between CMMI-DEV and CMMI-SVC
CMMI-DEV (22 Process Areas)
06 Process Areas specific to CMMI-DEV
16 Process Areas are common in all three models
CMMI-SVC
CMMI-ACQ
08 Process Areas specific to CMMI-SVC
06 Process Areas specific to CMMI-ACQ

CMMI – DEV Staged architecture



CMMI-SVC: 7 service specific PAs
1 shared PA with CMMI-DEV

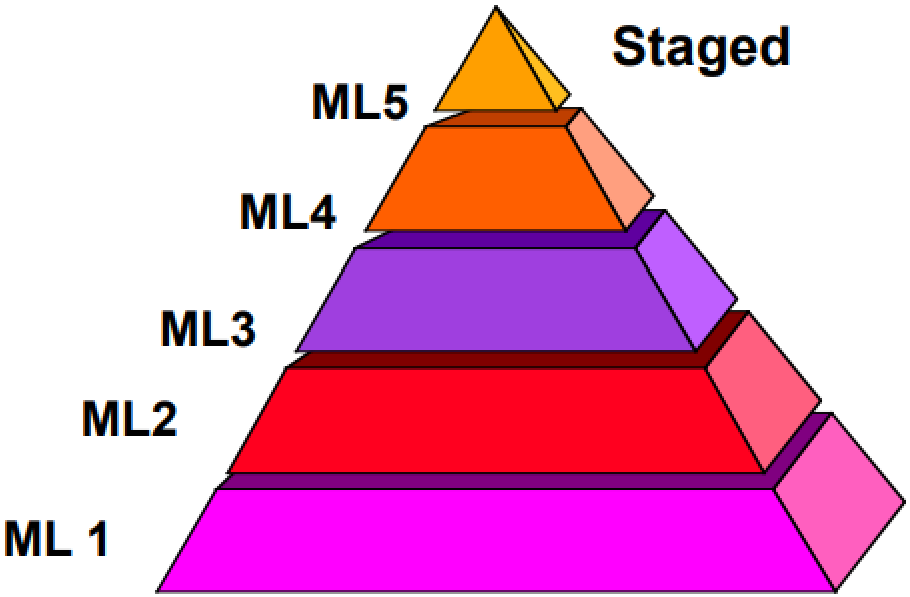
CMMI-DEV: 6 development specific Pas
1 shared PA with CMMI-SVC

CMMI-ACQ: 6 acquisition specific PAs

16 common Process Areas to *all* v1.3 constellations

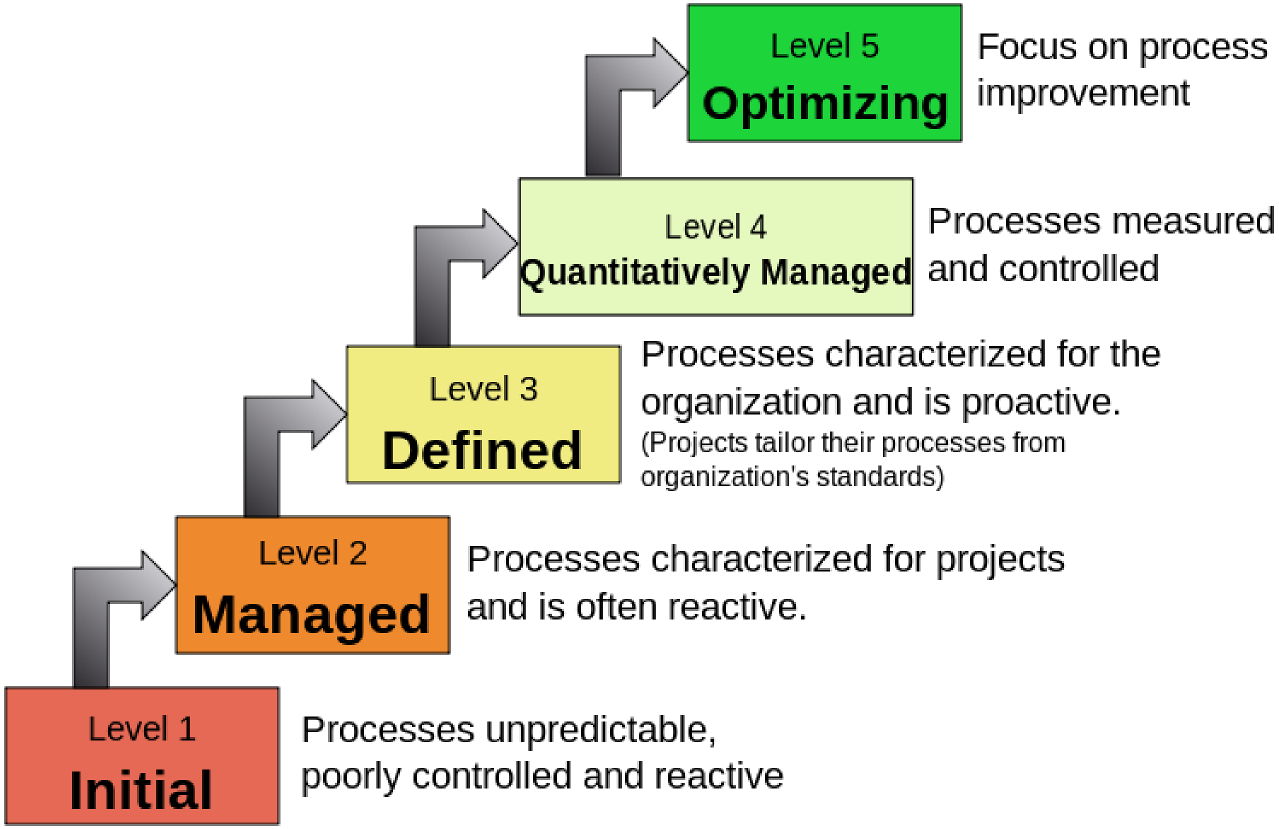
CMMI – DEV Staged architecture

- Defines 5 maturity levels (MLs)
- In order to achieve a maturity level all process areas associated to this level,
- Plus all process areas associated with levels below must have a certain minimal capability.



Capability Maturity Model Integration CMMI

Characteristics of the Maturity levels



Level 5 **Optimizing** Focus on process improvement

Level 4 **Quantitatively Managed** Processes measured and controlled

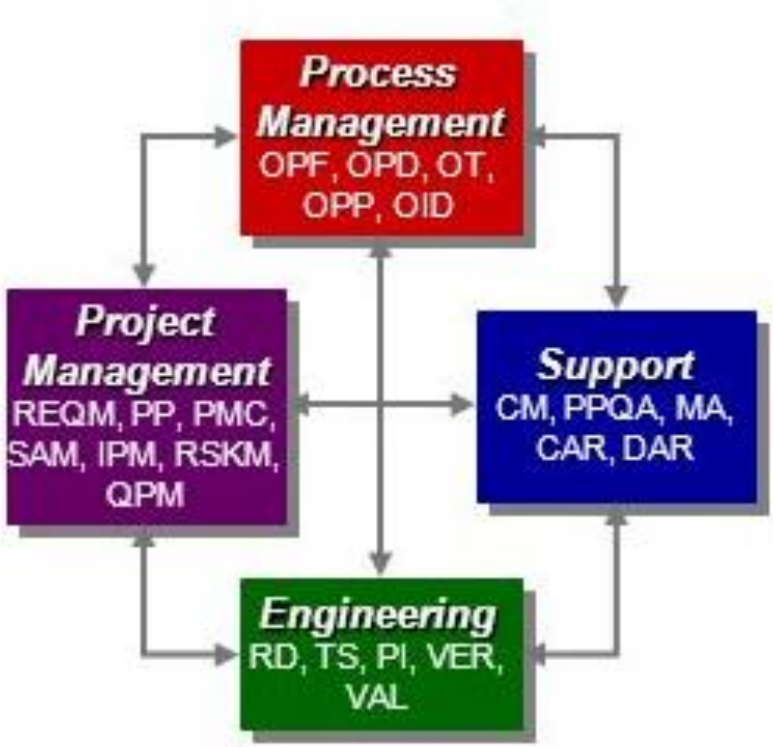
Level 3 **Defined** Processes characterized for the organization and is proactive. (Projects tailor their processes from organization's standards)

Level 2 **Managed** Processes characterized for projects and is often reactive.

Level 1 **Initial** Processes unpredictable, poorly controlled and reactive

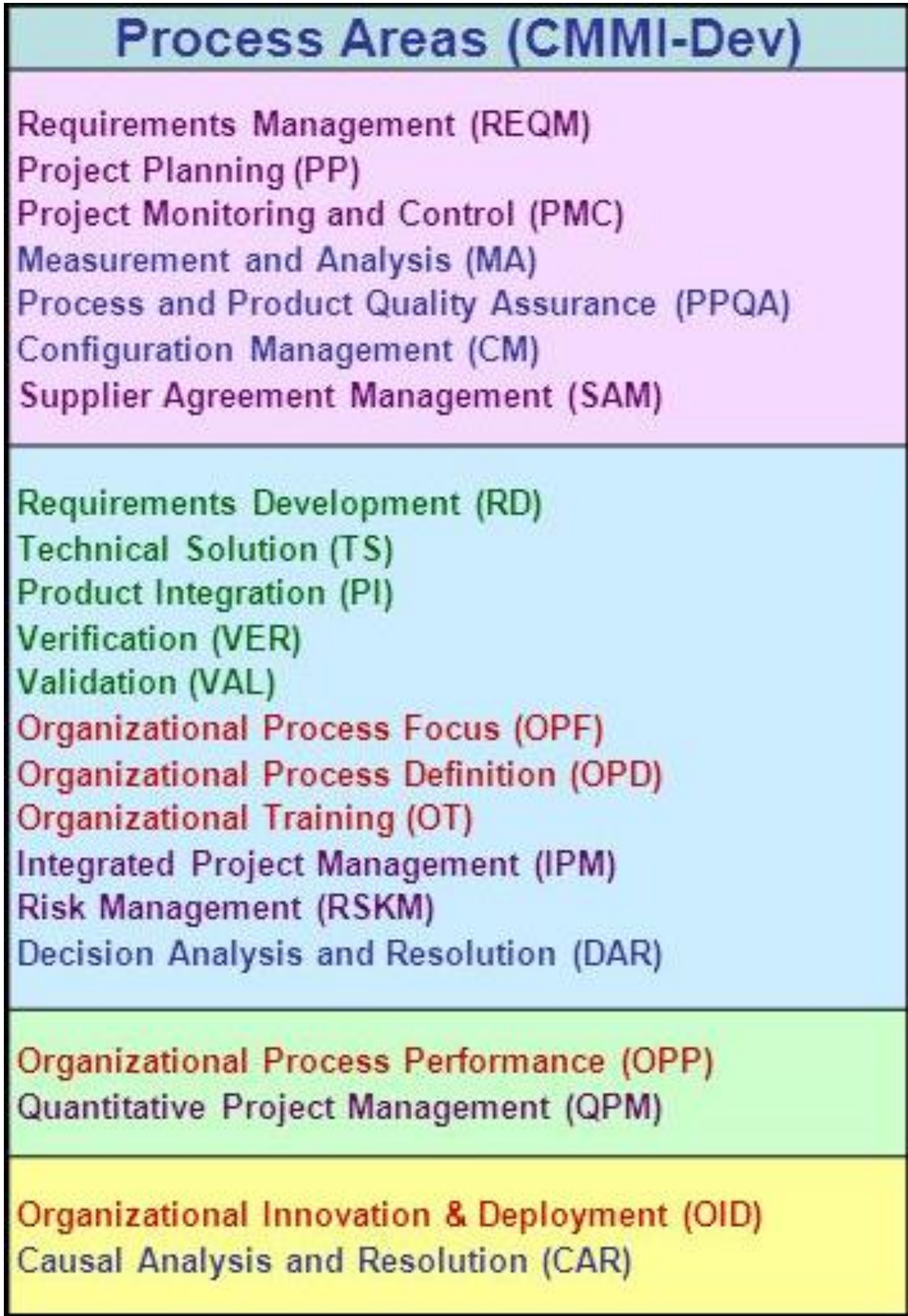
CMMI – DEV Staged architecture

- The 22 Process Areas (PAs) are grouped into four categories:
  1. Engineering
  2. Process Management
  3. Project Management
  4. Support



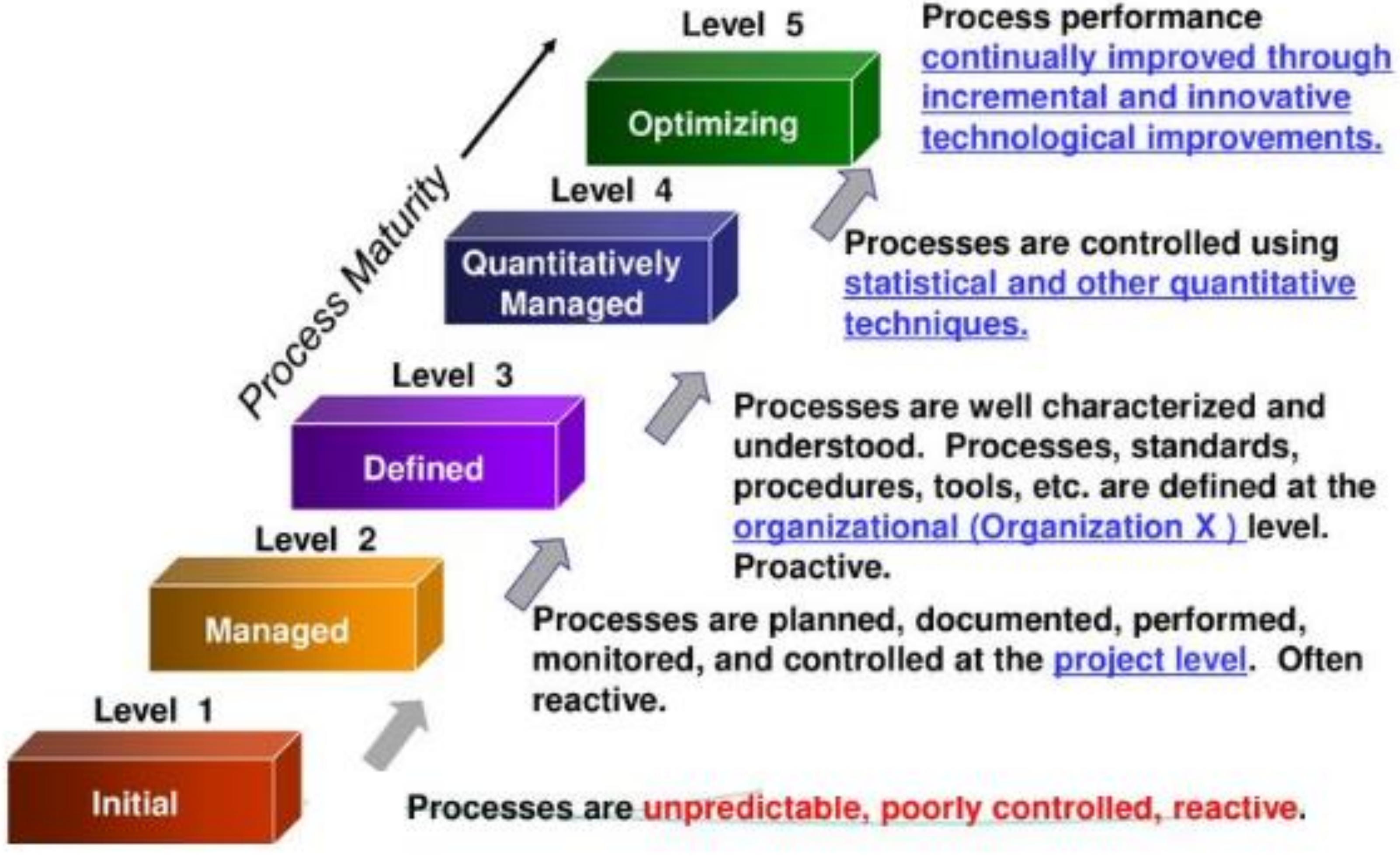
CMMI – DEV Staged architecture

- The 22 Process Areas



Process Areas (CMMI-Dev)
Requirements Management (REQM) Project Planning (PP) Project Monitoring and Control (PMC) Measurement and Analysis (MA) Process and Product Quality Assurance (PPQA) Configuration Management (CM) Supplier Agreement Management (SAM)
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)
Organizational Process Performance (OPP) Quantitative Project Management (QPM)
Organizational Innovation & Deployment (OID) Causal Analysis and Resolution (CAR)

CMMI – DEV Staged architecture



Process Maturity

Level 5

Optimizing

Process performance continually improved through incremental and innovative technological improvements.

Level 4

Quantitatively Managed

Processes are controlled using statistical and other quantitative techniques.

Level 3

Defined

Processes are well characterized and understood. Processes, standards, procedures, tools, etc. are defined at the organizational (Organization X) level. Proactive.

Level 2

Managed

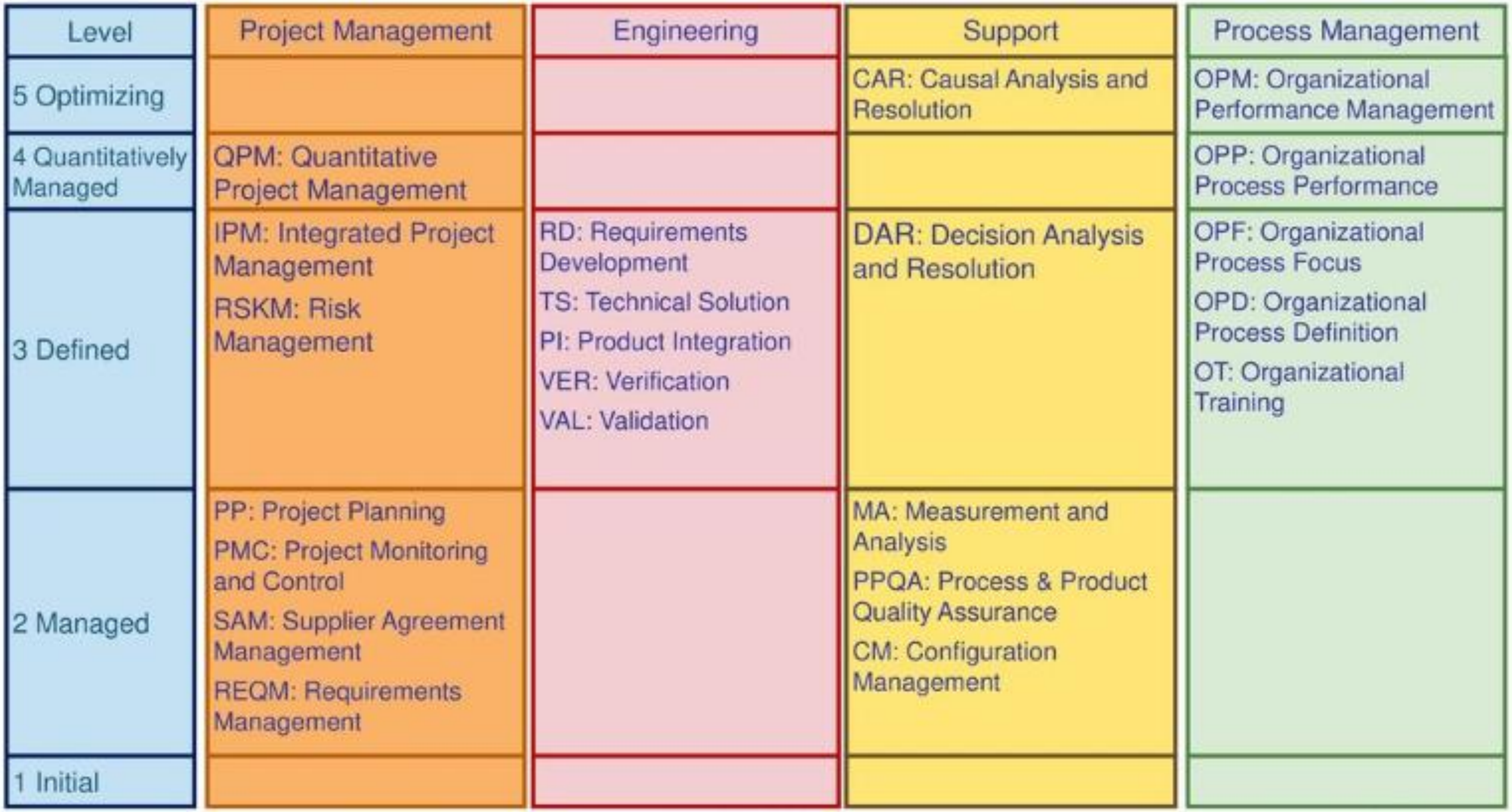
Processes are planned, documented, performed, monitored, and controlled at the project level. Often reactive.

Level 1

Initial

Processes are unpredictable, poorly controlled, reactive.

CMMI – DEV Staged architecture



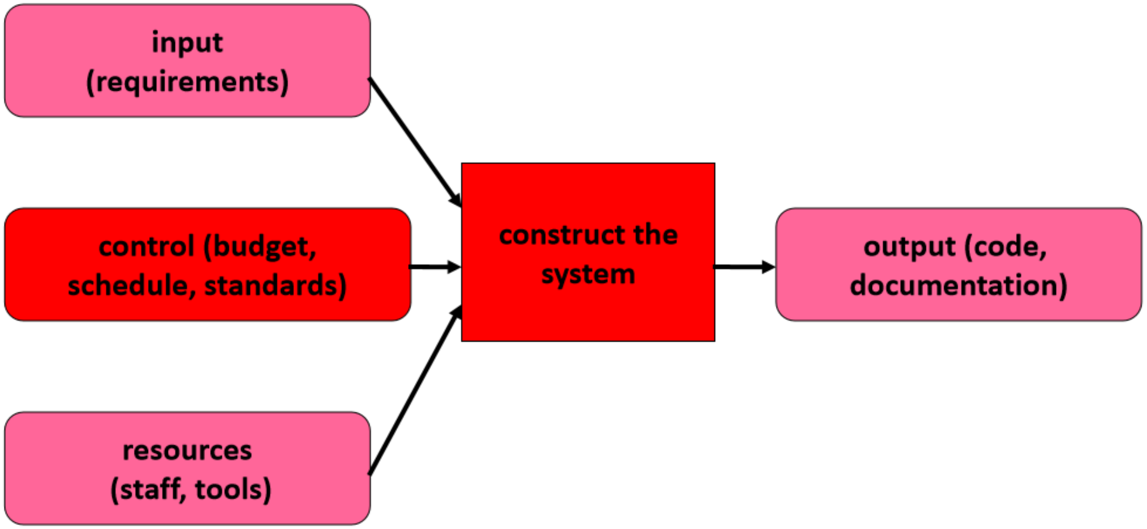
Level	Project Management	Engineering	Support	Process Management
5 Optimizing			CAR: Causal Analysis and Resolution	OPM: Organizational Performance Management
4 Quantitatively Managed	QPM: Quantitative Project Management			OPP: Organizational Process Performance
3 Defined	IPM: Integrated Project Management RSKM: Risk Management	RD: Requirements Development TS: Technical Solution PI: Product Integration VER: Verification VAL: Validation	DAR: Decision Analysis and Resolution	OPF: Organizational Process Focus OPD: Organizational Process Definition OT: Organizational Training
2 Managed	PP: Project Planning PMC: Project Monitoring and Control SAM: Supplier Agreement Management REQM: Requirements Management		MA: Measurement and Analysis PPQA: Process & Product Quality Assurance CM: Configuration Management	
1 Initial				

Maturity Levels: Initial

- **Processes are usually ad hoc and chaotic.**
- The organization usually does **not provide a stable environment.**
- Success in these organizations **depends on the competence and heroics of the people** in the organization and not on the use of proven processes.
- Maturity level INITAL organizations often produce products and services that work;
  - however, they **frequently exceed the budget and schedule of their projects.**

Maturity Levels

- **Managed or Repeatable**
- The projects of the organization have **ensured that requirements are managed and that processes are planned, performed, measured, and controlled.**
- Maturity level 2 helps to ensure that existing practices are **retained during times of stress**


input (requirements)
control (budget, schedule, standards)
resources (staff, tools)
construct the system
output (code, documentation)

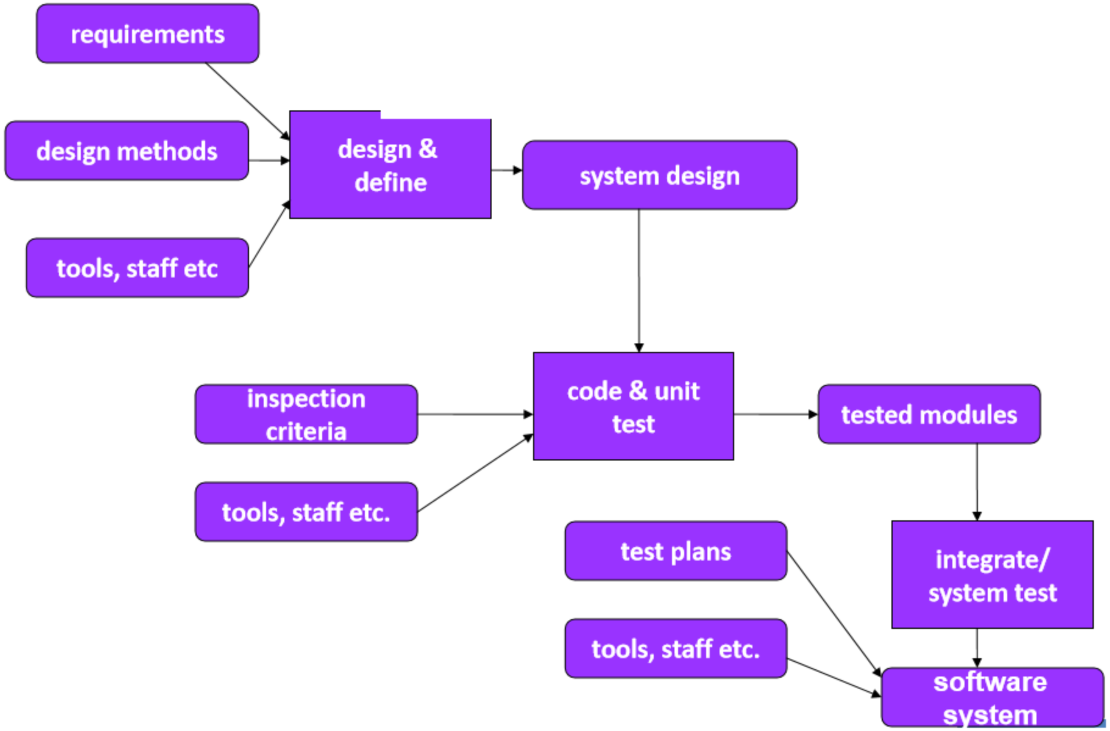
Maturity Levels

- Managed

- To move to ML 2 (Managed) focus is on process areas:
  - Requirements management
  - Project planning
  - Project monitoring & control
  - Sub-contract management
  - Measurement and analysis
  - Quality assurance
  - Configuration management

Maturity Levels: Defined

- Processes are well characterized and **understood, and are described in standards, procedures, tools, and methods.**
- A critical distinction between maturity level 2 and maturity level 3 **is the scope of standards, process descriptions, and procedures.**



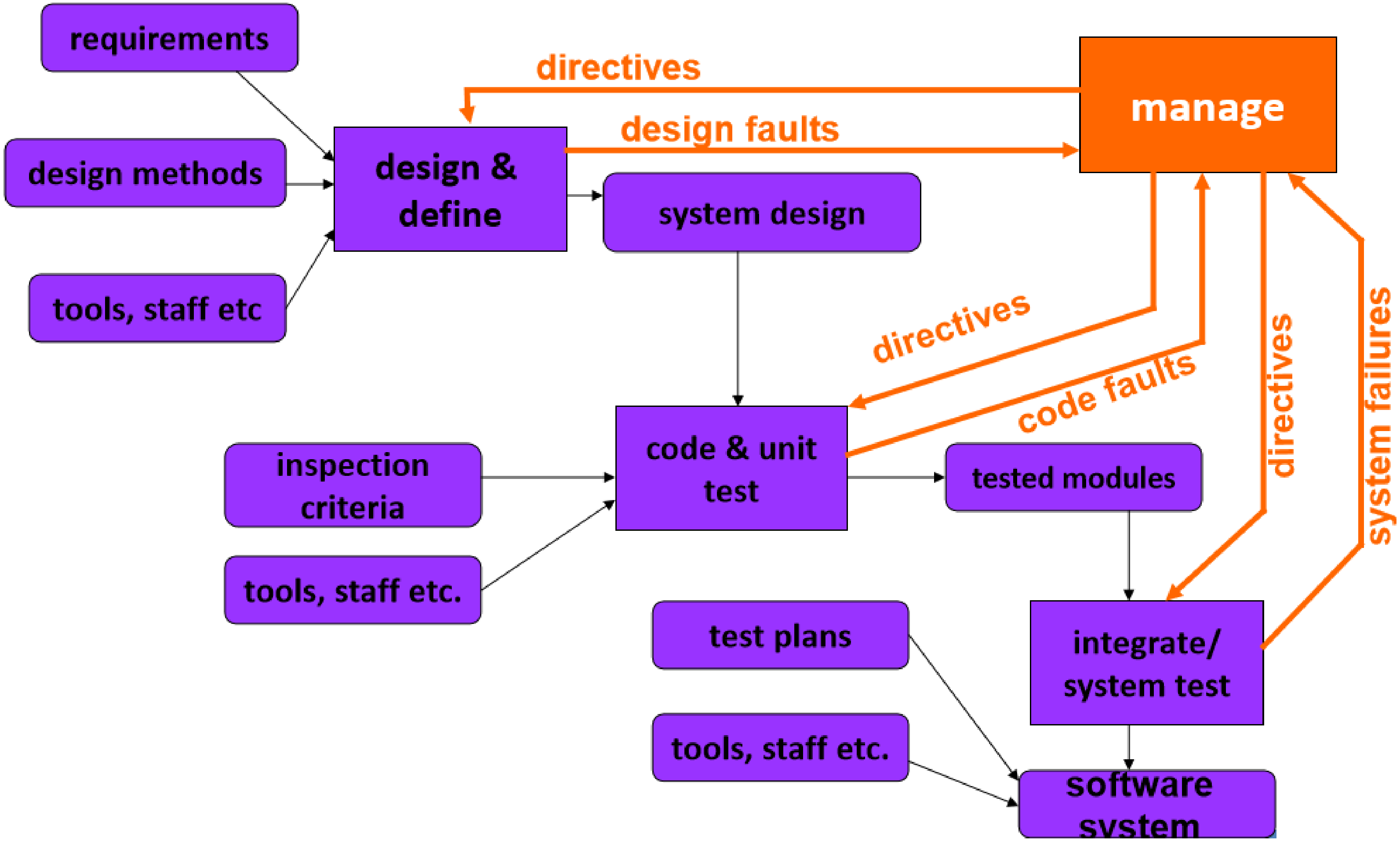
Maturity Levels: Defined

- To move to ML 3 (Defined) focus on process areas:
  - Requirements development and technical solution
  - Product integration
  - Verification and validation
  - Organizational process definition
  - Organizational process focus
  - Organizational training
  - Risk management
  - Integrated project management
  - Decision analysis and resolution

Maturity Levels: Quantitavely Managed

- **Subprocesses are selected that significantly contribute to overall process performance**.
  - These selected subprocesses are controlled using statistical and other quantitative techniques.
- **Quantitative objectives for quality and process performance are established** and used as criteria in managing processes.
- Quantitative objectives are **based on the needs of the customer, end users, organization, and process implementers**.

Maturity Levels: Quantitatively Managed



Maturity Levels: Quantitavely Managed

- Maturity level 5 focuses on **continually improving process performance** *through both incremental and innovative technological improvements*

- Processes are **continually improved based on a quantitative understanding of the common causes of variation inherent in processes.**

- Quantitative process-improvement objectives for the organization are established, **continually revised to reflect changing business objectives**, and used as criteria in managing process improvement.

Maturity Levels: Optimizing

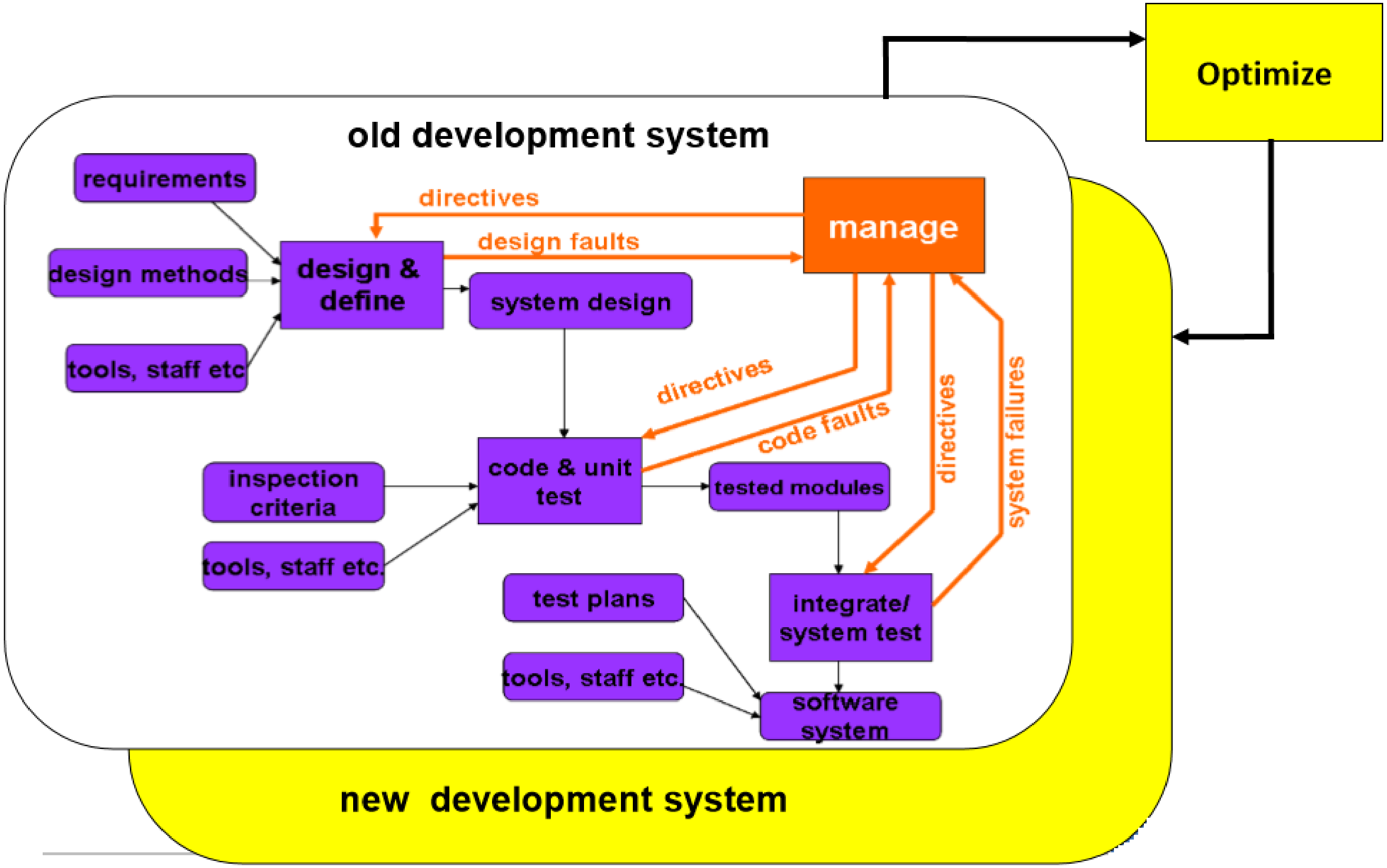
- To move to ML 5 (Optimizing) focus on process areas:
  - Causal analysis and resolution
  - Organizational performance management

Maturity Levels: Optimizing

- To move to ML 5 (Optimizing) focus on process areas:
  - Causal analysis and resolution
  - Organizational performance management

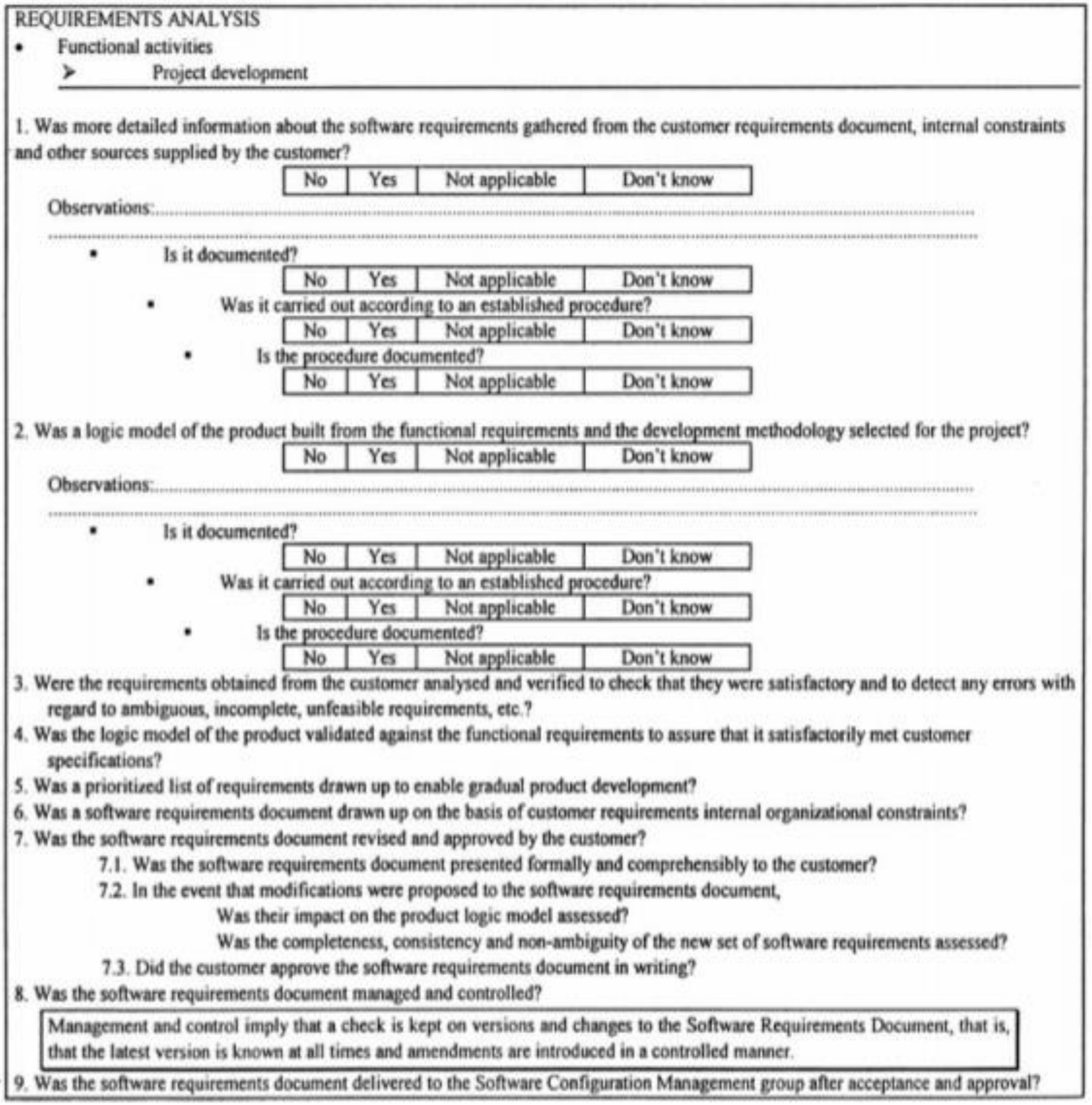
Maturity Levels: Optimizing

- Optimizing



CMMI Evaluation – Questionnaire Example

- Related to Process Areas **Requirements Development**



REQUIREMENTS ANALYSIS

- Functional activities
  - ➤ Project development

1. Was more detailed information about the software requirements gathered from the customer requirements document, internal constraints and other sources supplied by the customer?

No	Yes	Not applicable	Don't know

Observations:...

....

- Is it documented?

No	Yes	Not applicable	Don't know

- Was it carried out according to an established procedure?

No	Yes	Not applicable	Don't know

- Is the procedure documented?

No	Yes	Not applicable	Don't know

2. Was a logic model of the product built from the functional requirements and the development methodology selected for the project?

No	Yes	Not applicable	Don't know

Observations:...

....

- Is it documented?

No	Yes	Not applicable	Don't know

- Was it carried out according to an established procedure?

No	Yes	Not applicable	Don't know

- Is the procedure documented?

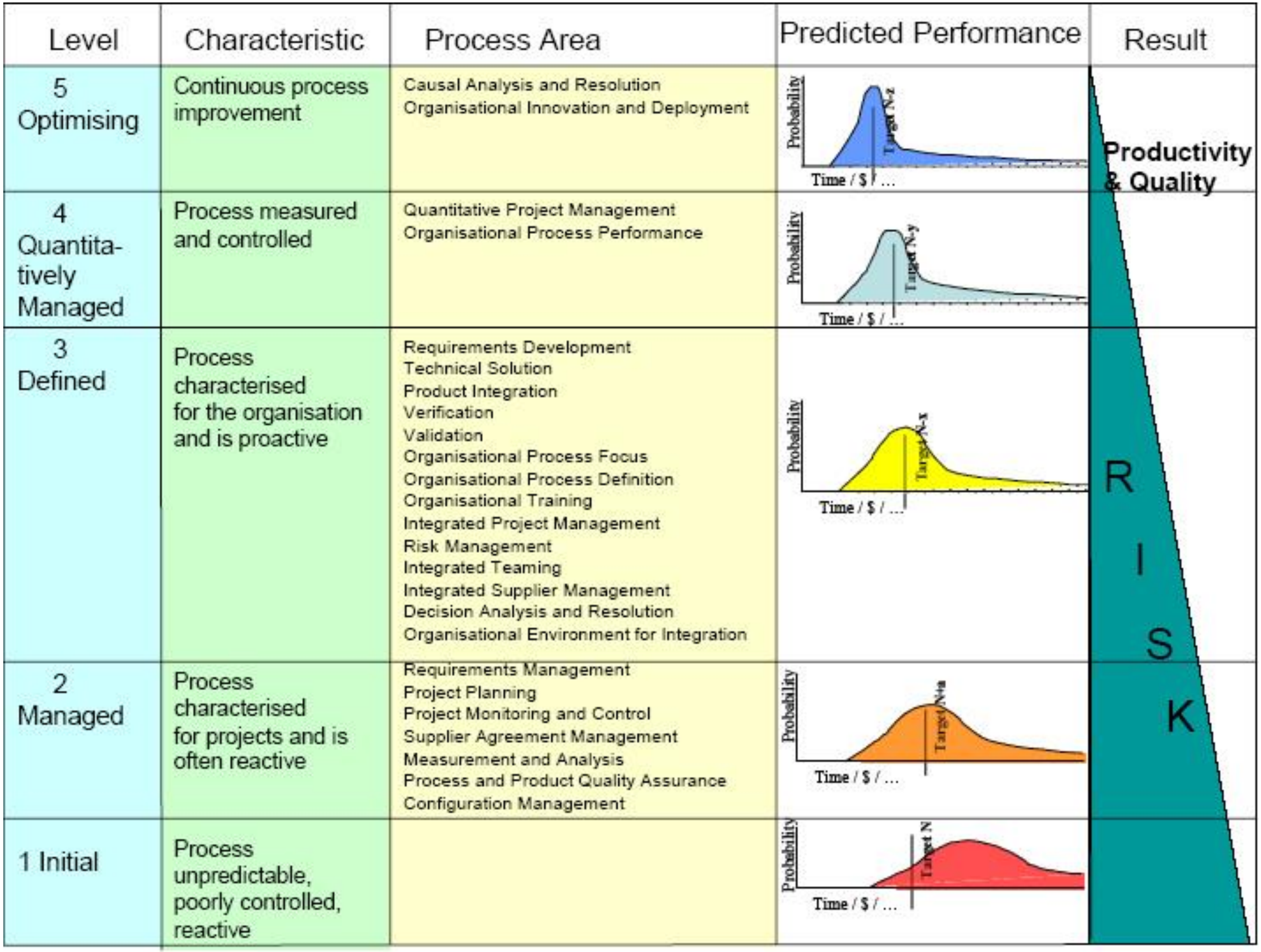
No	Yes	Not applicable	Don't know

3. Were the requirements obtained from the customer analysed and verified to check that they were satisfactory and to detect any errors with regard to ambiguous, incomplete, unfeasible requirements, etc.?
4. Was the logic model of the product validated against the functional requirements to assure that it satisfactorily met customer specifications?
5. Was a prioritized list of requirements drawn up to enable gradual product development?
6. Was a software requirements document drawn up on the basis of customer requirements internal organizational constraints?
7. Was the software requirements document revised and approved by the customer?
   7.1. Was the software requirements document presented formally and comprehensibly to the customer?
   7.2. In the event that modifications were proposed to the software requirements document,
   Was their impact on the product logic model assessed?
   Was the completeness, consistency and non-ambiguity of the new set of software requirements assessed?
   7.3. Did the customer approve the software requirements document in writing?
8. Was the software requirements document managed and controlled?

Management and control imply that a check is kept on versions and changes to the Software Requirements Document, that is, that the latest version is known at all times and amendments are introduced in a controlled manner.

9. Was the software requirements document delivered to the Software Configuration Management group after acceptance and approval?

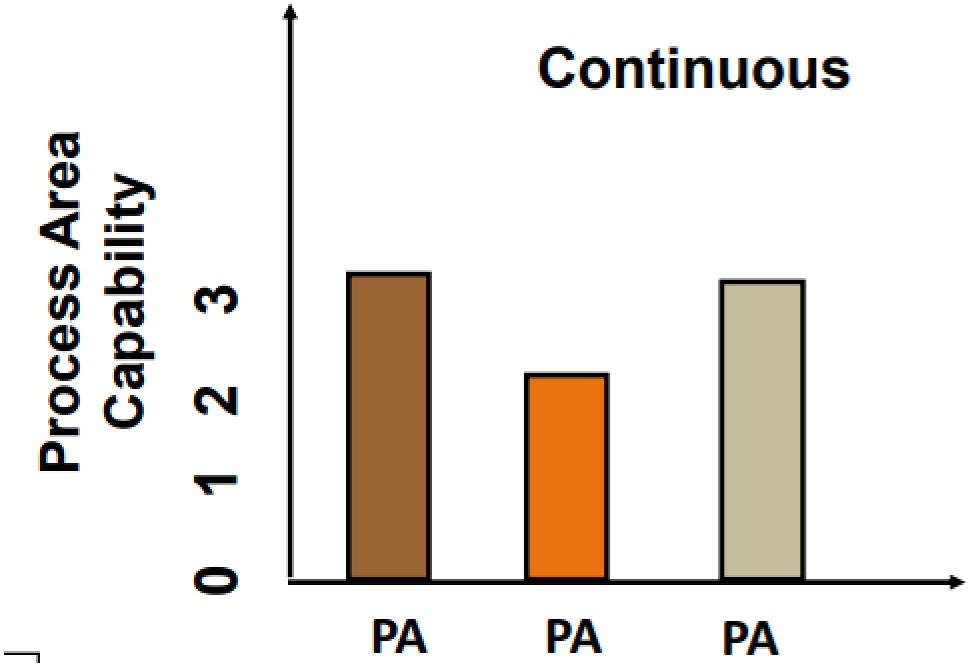
CMMI – DEV Staged architecture



Level	Characteristic	Process Area	Predicted Performance	Result
5 Optimising	Continuous process improvement	Causal Analysis and Resolution Organisational Innovation and Deployment	Probability Target N-z Time / $ / …	Productivity & Quality
4 Quantitatively Managed	Process measured and controlled	Quantitative Project Management Organisational Process Performance	Probability Target N-y Time / $ / …	
3 Defined	Process characterised for the organisation and is proactive	Requirements Development Technical Solution Product Integration Verification Validation Organisational Process Focus Organisational Process Definition Organisational Training Integrated Project Management Risk Management Integrated Teaming Integrated Supplier Management Decision Analysis and Resolution Organisational Environment for Integration	Probability Target N-x Time / $ / …	R I S
2 Managed	Process characterised for projects and is often reactive	Requirements Management Project Planning Project Monitoring and Control Supplier Agreement Management Measurement and Analysis Process and Product Quality Assurance Configuration Management	Probability Target N+a Time / $ / …	K
1 Initial	Process unpredictable, poorly controlled, reactive		Probability Target N Time / $ / …	

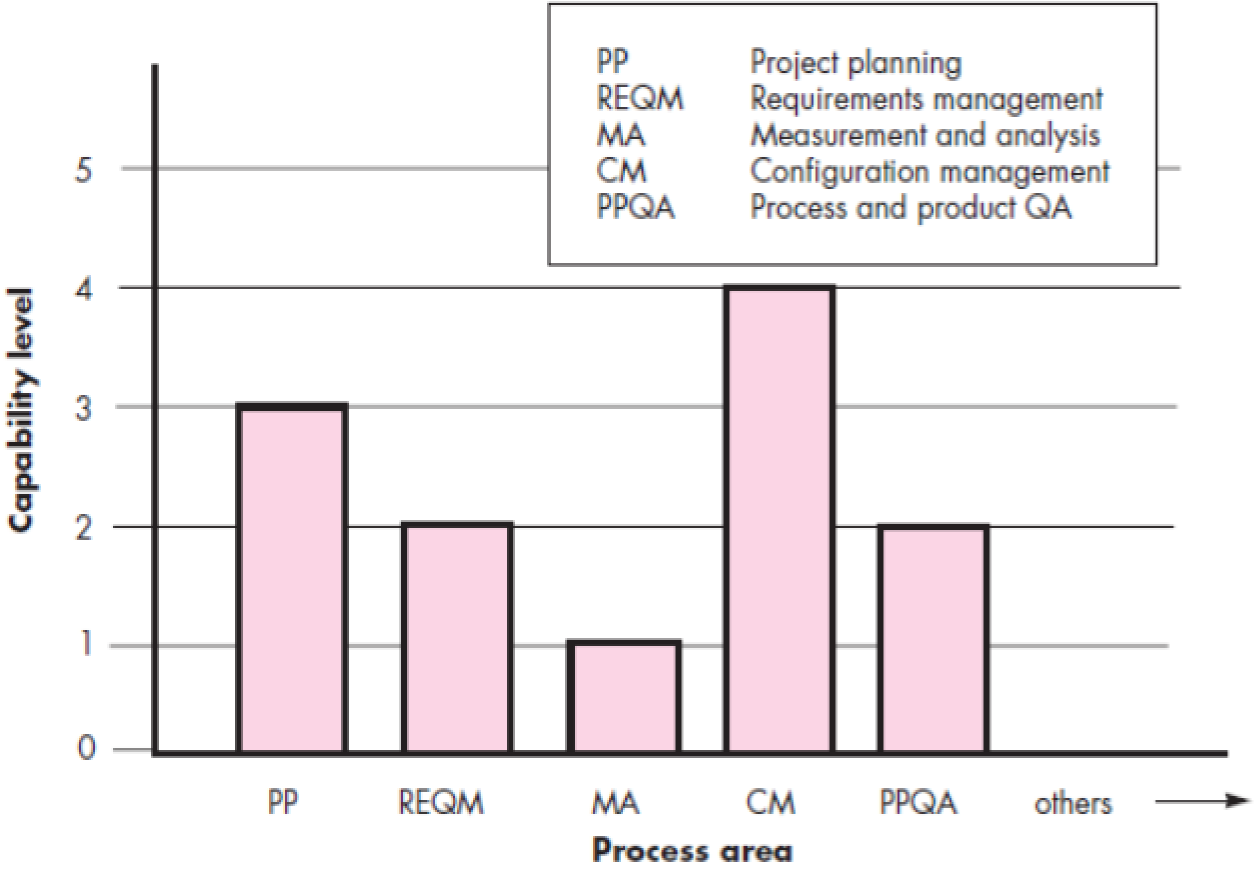
CMMI – DEV Continues architecture

- A maturity profile is established based on the **capabilities of individual process areas**



CMMI – DEV Continues architecture

- A maturity profile is established based on the capabilities of individual process areas



Capability levels

- **Capability Level 0: Incomplete**
  - An incomplete process is a process that either is **not performed** or is **partially performed**.

Capability Levels

- **Capability Level 1: Performed**
  - A capability level 1 process is characterized as a **performed process**.
  - A performed process is a process that:
    - **accomplishes the needed work to produce work products**
    - **without relying entirely on process documentation or plan**

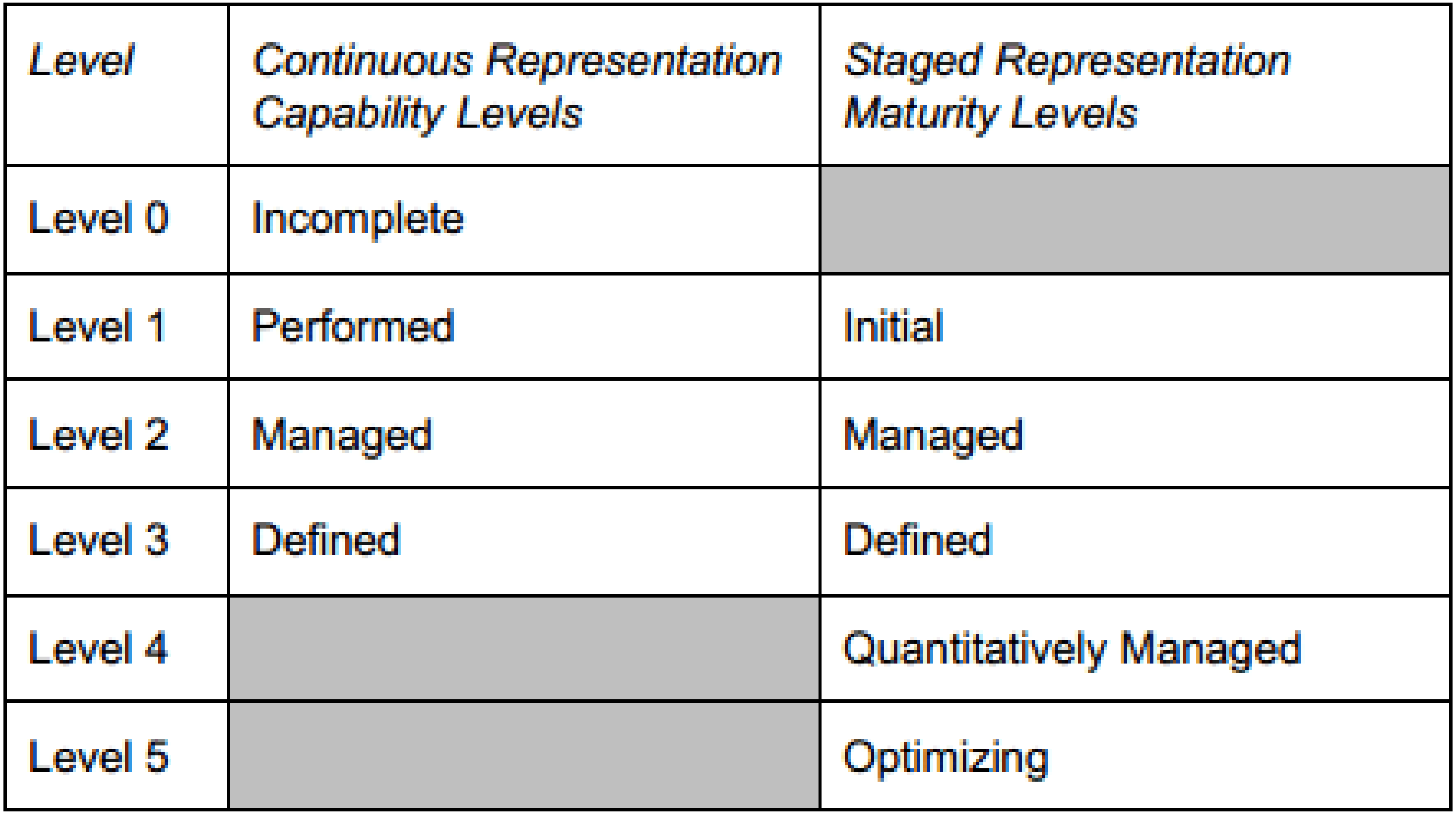
Capability Levels

- **Capability Level 2: Managed**
  - A capability level 2 process is characterized 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;**
    - is **evaluated for adherence to its process description**.

Capability Levels

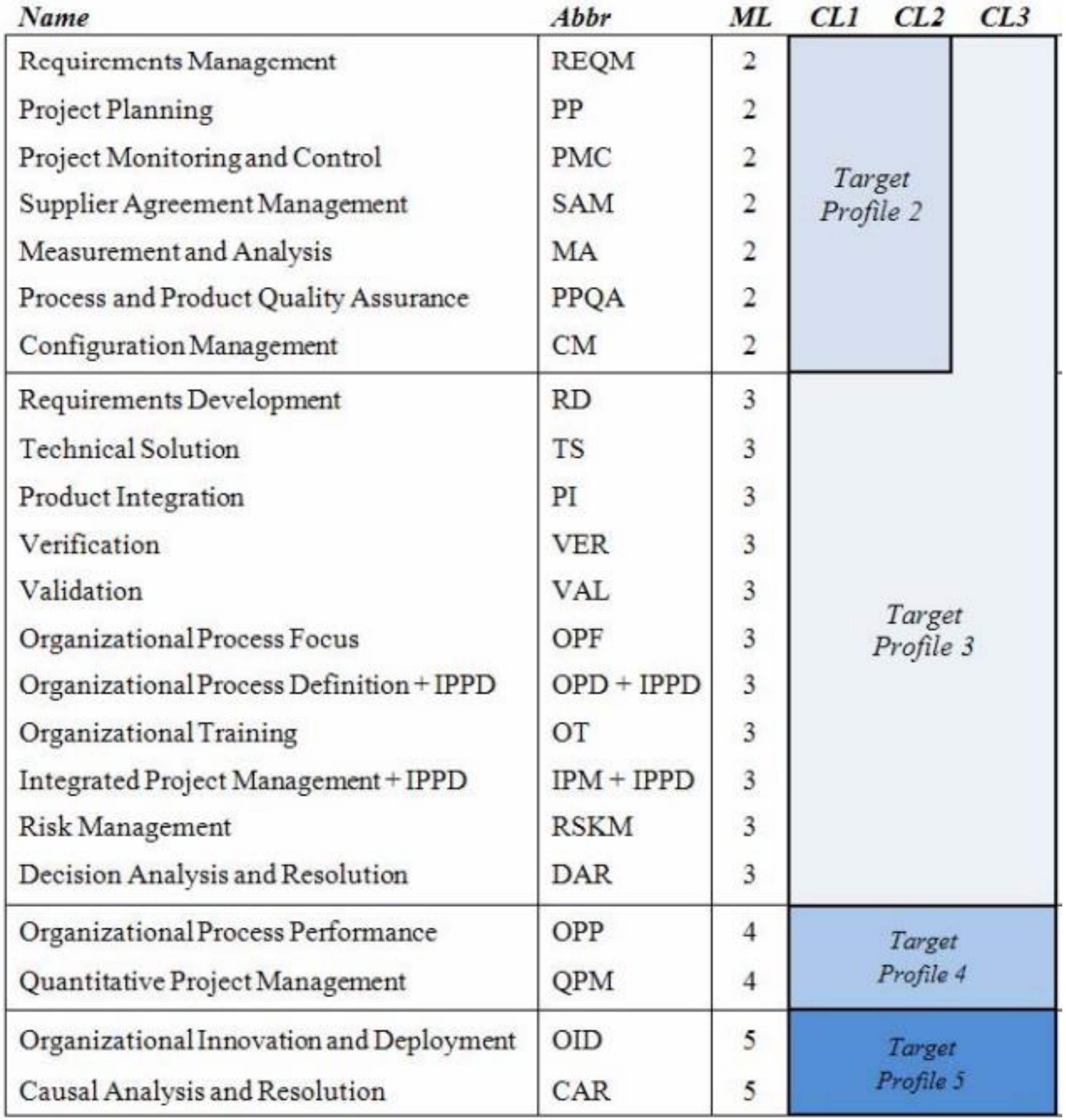
- **Capability Level 3: Defined**
  - A defined process is a managed process that is tailored **from the organization’s set of standard processes according to the organization’s tailoring guidelines;**
  - has a **maintained process description**;
  - and **contributes process related experiences to the organizational process assets.**

Continuous Representation: Capability Levels



Level	*Continuous Representation Capability Levels*	*Staged Representation Maturity Levels*
Level 0	Incomplete	
Level 1	Performed	Initial
Level 2	Managed	Managed
Level 3	Defined	Defined
Level 4		Quantitatively Managed
Level 5		Optimizing

Target Profiles and Equivalent Staging



Name	*Abbr*	*ML*	*CL1*	*CL2*	*CL3*
Requirements Management	REQM	2	*Target Profile 2*		
Project Planning	PP	2			
Project Monitoring and Control	PMC	2			
Supplier Agreement Management	SAM	2			
Measurement and Analysis	MA	2			
Process and Product Quality Assurance	PPQA	2			
Configuration Management	CM	2			
Requirements Development	RD	3	*Target Profile 3*		
Technical Solution	TS	3			
Product Integration	PI	3			
Verification	VER	3			
Validation	VAL	3			
Organizational Process Focus	OPF	3			
Organizational Process Definition + IPPD	OPD + IPPD	3			
Organizational Training	OT	3			
Integrated Project Management + IPPD	IPM + IPPD	3			
Risk Management	RSKM	3			
Decision Analysis and Resolution	DAR	3			
Organizational Process Performance	OPP	4	*Target Profile 4*		
Quantitative Project Management	QPM	4			
Organizational Innovation and Deployment	OID	5	*Target Profile 5*		
Causal Analysis and Resolution	CAR	5			

ISO 15504 Maturity Models

- SPICE (ISO 15504)
  - Different structure of processes than in CMMI (roughly following ISO 12207)
  - 6 Maturity levels (beginning at Level 0)

- https://cmmiinstitute.com/

ISO/IEC 15504 IT process assessment

- To provide **guidance on the assessment of software development processes**

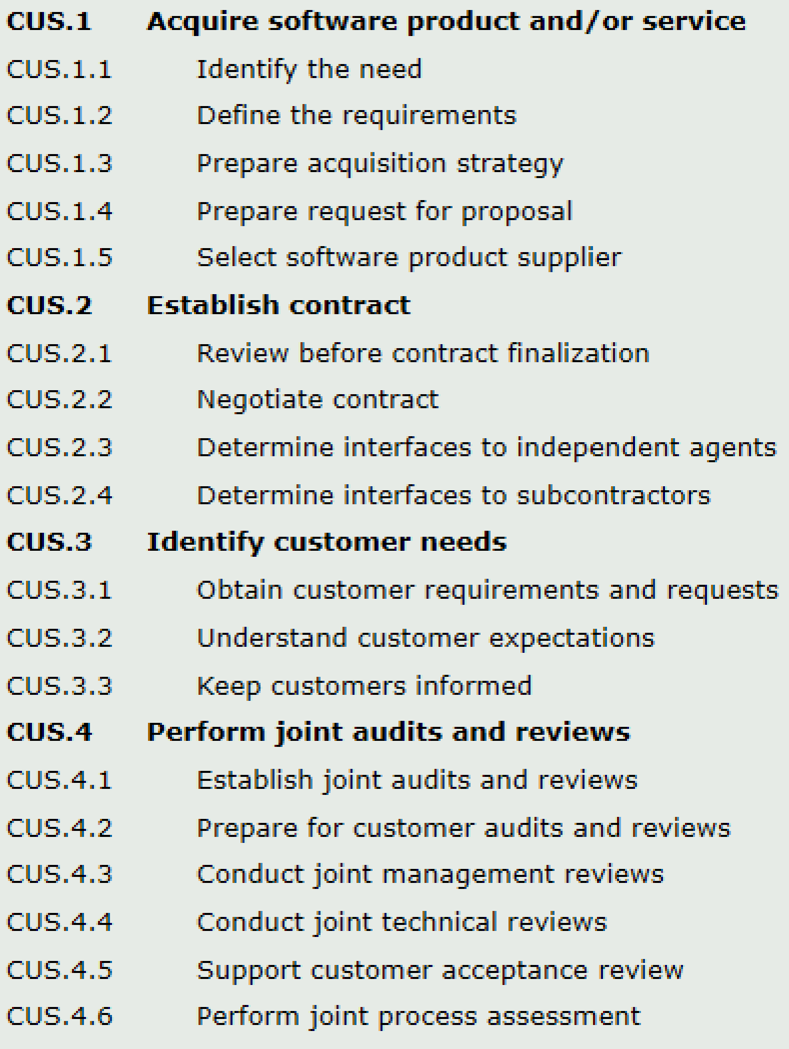
- Process Reference Model:
  - Needs a defined set of processes that represent good practice to be the benchmark
  - **ISO 12207 is the default process reference model**
  - Could use others in specific environments

Process Categories ISO 15504

- Customer-supplier (CUS)
- Engineering (ENG)
- Project (PRO)
- Support (SUP)
- Organizing (ORG)

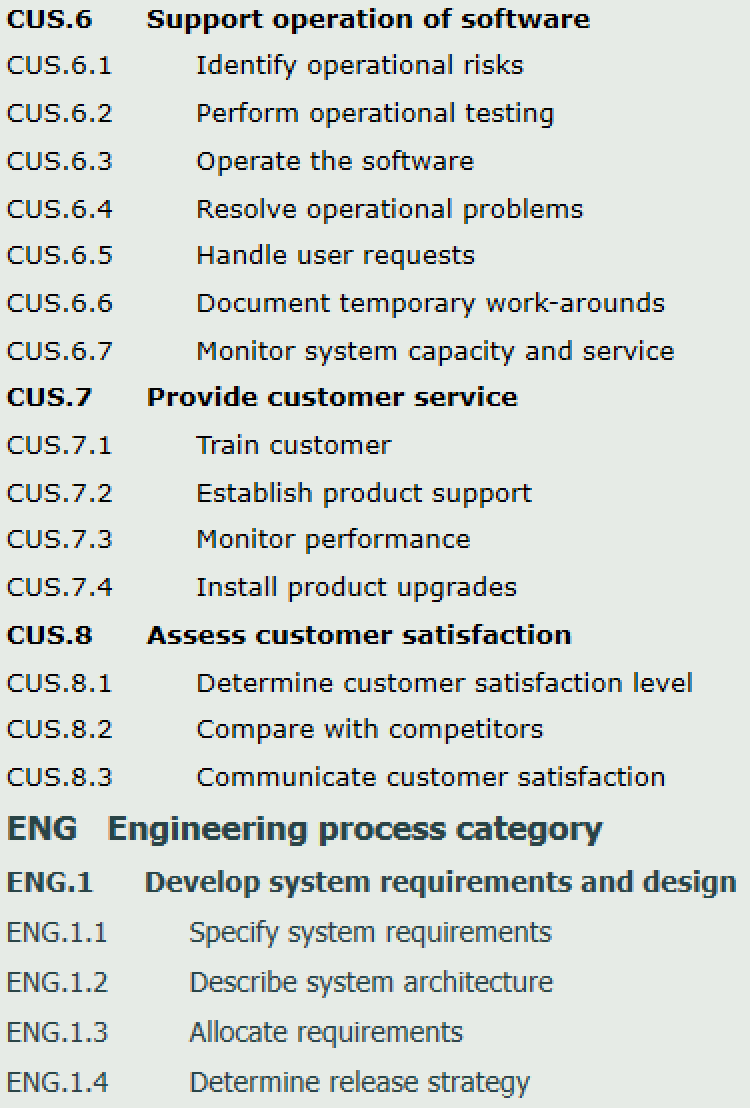
http://www.rad.fr/spice1.htm

Summary list of base practices



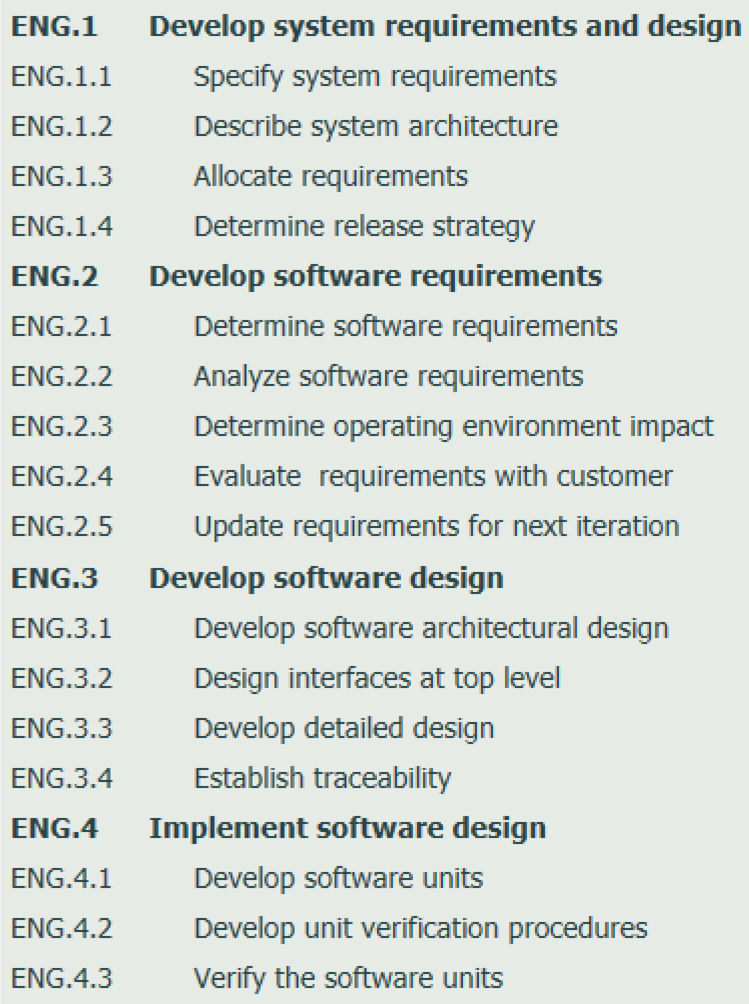
CUS.1	**Acquire software product and/or service**
CUS.1.1	Identify the need
CUS.1.2	Define the requirements
CUS.1.3	Prepare acquisition strategy
CUS.1.4	Prepare request for proposal
CUS.1.5	Select software product supplier
CUS.2	**Establish contract**
CUS.2.1	Review before contract finalization
CUS.2.2	Negotiate contract
CUS.2.3	Determine interfaces to independent agents
CUS.2.4	Determine interfaces to subcontractors
CUS.3	**Identify customer needs**
CUS.3.1	Obtain customer requirements and requests
CUS.3.2	Understand customer expectations
CUS.3.3	Keep customers informed
CUS.4	**Perform joint audits and reviews**
CUS.4.1	Establish joint audits and reviews
CUS.4.2	Prepare for customer audits and reviews
CUS.4.3	Conduct joint management reviews
CUS.4.4	Conduct joint technical reviews
CUS.4.5	Support customer acceptance review
CUS.4.6	Perform joint process assessment

http://www.rad.fr/spice1.htm

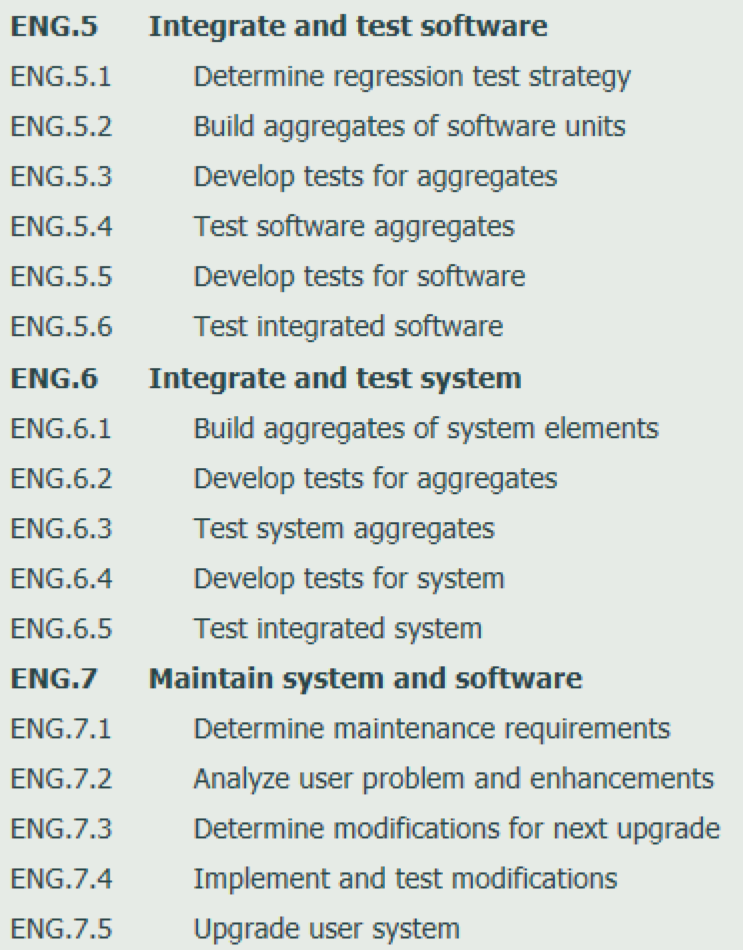


CUS.6	**Support operation of software**
CUS.6.1	Identify operational risks
CUS.6.2	Perform operational testing
CUS.6.3	Operate the software
CUS.6.4	Resolve operational problems
CUS.6.5	Handle user requests
CUS.6.6	Document temporary work-arounds
CUS.6.7	Monitor system capacity and service
CUS.7	**Provide customer service**
CUS.7.1	Train customer
CUS.7.2	Establish product support
CUS.7.3	Monitor performance
CUS.7.4	Install product upgrades
CUS.8	**Assess customer satisfaction**
CUS.8.1	Determine customer satisfaction level
CUS.8.2	Compare with competitors
CUS.8.3	Communicate customer satisfaction
ENG	**Engineering process category**
ENG.1	**Develop system requirements and design**
ENG.1.1	Specify system requirements
ENG.1.2	Describe system architecture
ENG.1.3	Allocate requirements
ENG.1.4	Determine release strategy

Summary list of base practices



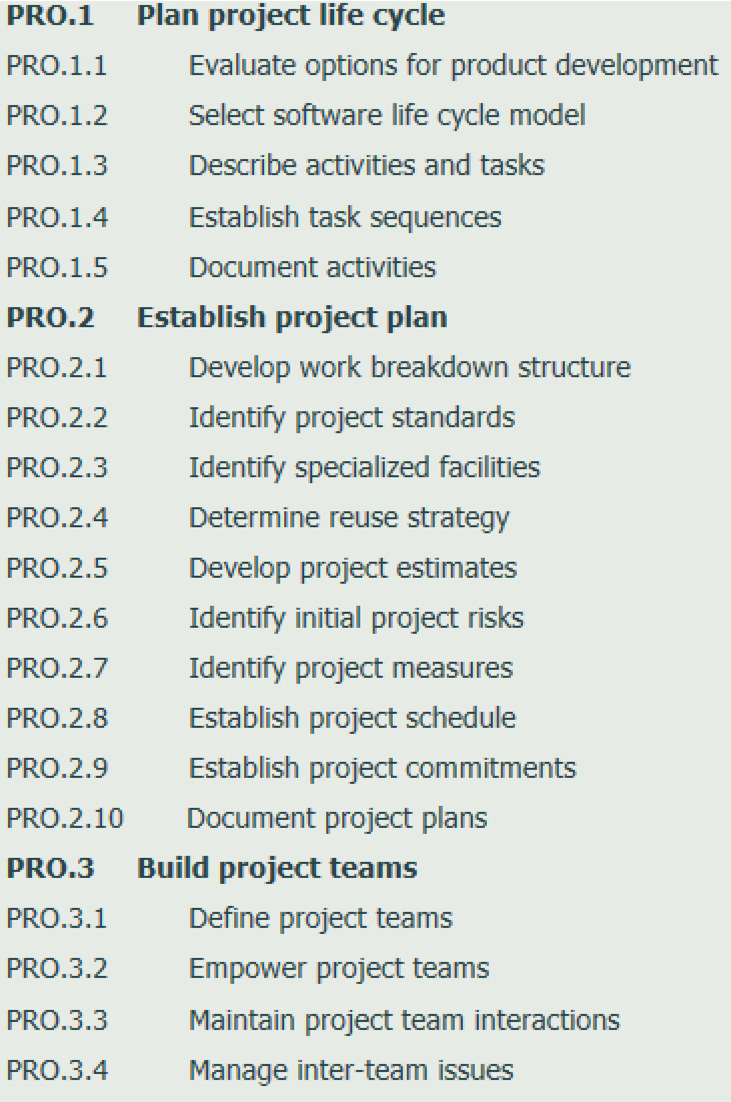
ENG.1	**Develop system requirements and design**
ENG.1.1	Specify system requirements
ENG.1.2	Describe system architecture
ENG.1.3	Allocate requirements
ENG.1.4	Determine release strategy
ENG.2	**Develop software requirements**
ENG.2.1	Determine software requirements
ENG.2.2	Analyze software requirements
ENG.2.3	Determine operating environment impact
ENG.2.4	Evaluate requirements with customer
ENG.2.5	Update requirements for next iteration
ENG.3	**Develop software design**
ENG.3.1	Develop software architectural design
ENG.3.2	Design interfaces at top level
ENG.3.3	Develop detailed design
ENG.3.4	Establish traceability
ENG.4	**Implement software design**
ENG.4.1	Develop software units
ENG.4.2	Develop unit verification procedures
ENG.4.3	Verify the software units



ENG.5	**Integrate and test software**
ENG.5.1	Determine regression test strategy
ENG.5.2	Build aggregates of software units
ENG.5.3	Develop tests for aggregates
ENG.5.4	Test software aggregates
ENG.5.5	Develop tests for software
ENG.5.6	Test integrated software
ENG.6	**Integrate and test system**
ENG.6.1	Build aggregates of system elements
ENG.6.2	Develop tests for aggregates
ENG.6.3	Test system aggregates
ENG.6.4	Develop tests for system
ENG.6.5	Test integrated system
ENG.7	**Maintain system and software**
ENG.7.1	Determine maintenance requirements
ENG.7.2	Analyze user problem and enhancements
ENG.7.3	Determine modifications for next upgrade
ENG.7.4	Implement and test modifications
ENG.7.5	Upgrade user system

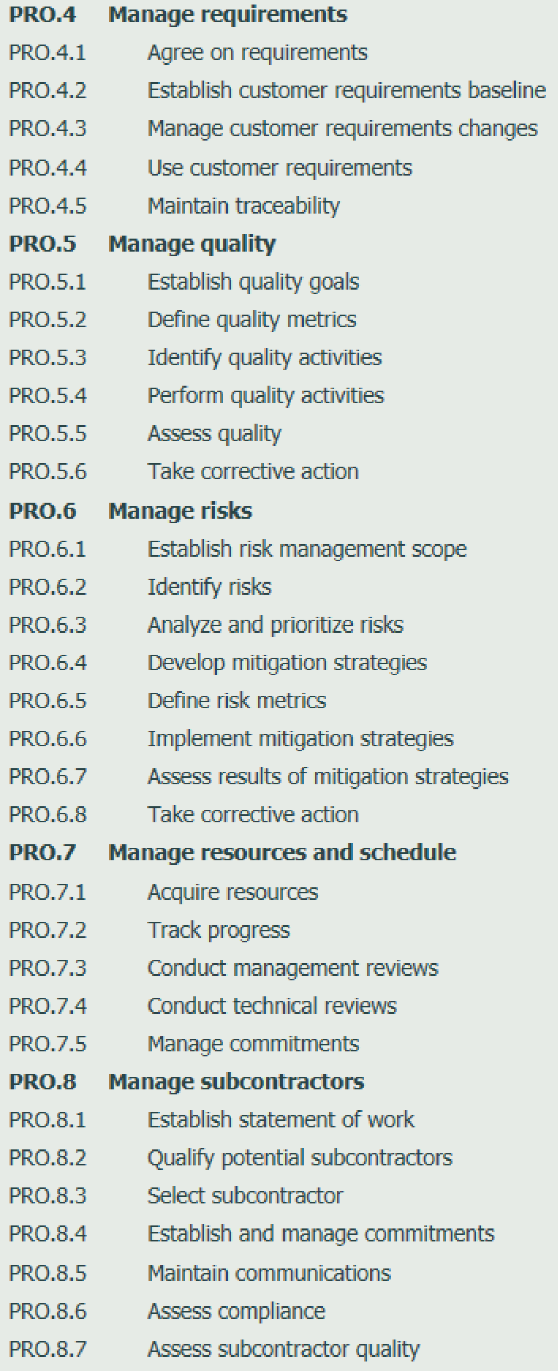
http://www.rad.fr/spice1.htm

Summary list of base practices



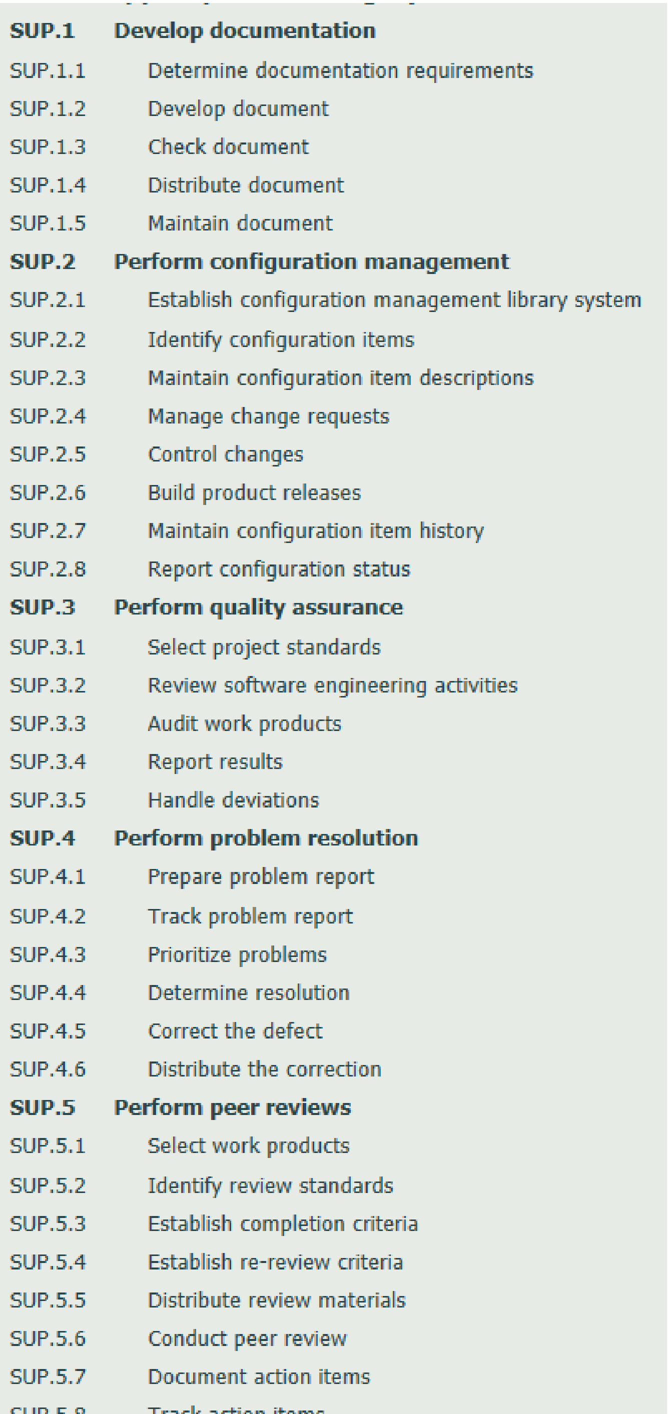
PRO.1	**Plan project life cycle**
PRO.1.1	Evaluate options for product development
PRO.1.2	Select software life cycle model
PRO.1.3	Describe activities and tasks
PRO.1.4	Establish task sequences
PRO.1.5	Document activities
PRO.2	**Establish project plan**
PRO.2.1	Develop work breakdown structure
PRO.2.2	Identify project standards
PRO.2.3	Identify specialized facilities
PRO.2.4	Determine reuse strategy
PRO.2.5	Develop project estimates
PRO.2.6	Identify initial project risks
PRO.2.7	Identify project measures
PRO.2.8	Establish project schedule
PRO.2.9	Establish project commitments
PRO.2.10	Document project plans
PRO.3	**Build project teams**
PRO.3.1	Define project teams
PRO.3.2	Empower project teams
PRO.3.3	Maintain project team interactions
PRO.3.4	Manage inter-team issues

http://www.rad.fr/spice1.htm



PRO.4	**Manage requirements**
PRO.4.1	Agree on requirements
PRO.4.2	Establish customer requirements baseline
PRO.4.3	Manage customer requirements changes
PRO.4.4	Use customer requirements
PRO.4.5	Maintain traceability
PRO.5	**Manage quality**
PRO.5.1	Establish quality goals
PRO.5.2	Define quality metrics
PRO.5.3	Identify quality activities
PRO.5.4	Perform quality activities
PRO.5.5	Assess quality
PRO.5.6	Take corrective action
PRO.6	**Manage risks**
PRO.6.1	Establish risk management scope
PRO.6.2	Identify risks
PRO.6.3	Analyze and prioritize risks
PRO.6.4	Develop mitigation strategies
PRO.6.5	Define risk metrics
PRO.6.6	Implement mitigation strategies
PRO.6.7	Assess results of mitigation strategies
PRO.6.8	Take corrective action
PRO.7	**Manage resources and schedule**
PRO.7.1	Acquire resources
PRO.7.2	Track progress
PRO.7.3	Conduct management reviews
PRO.7.4	Conduct technical reviews
PRO.7.5	Manage commitments
PRO.8	**Manage subcontractors**
PRO.8.1	Establish statement of work
PRO.8.2	Qualify potential subcontractors
PRO.8.3	Select subcontractor
PRO.8.4	Establish and manage commitments
PRO.8.5	Maintain communications
PRO.8.6	Assess compliance
PRO.8.7	Assess subcontractor quality

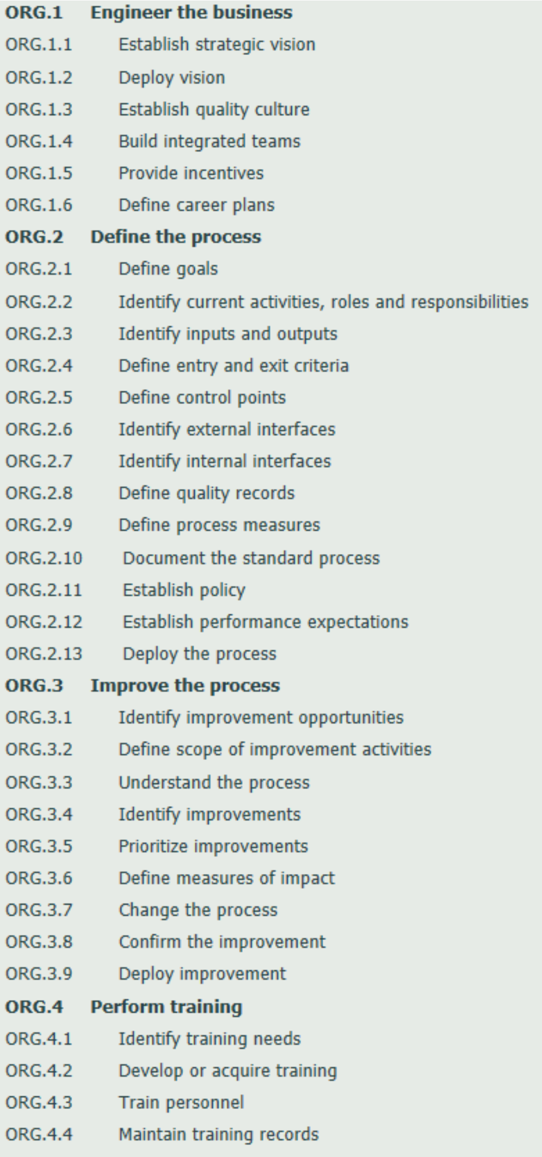
Summary list of base practices



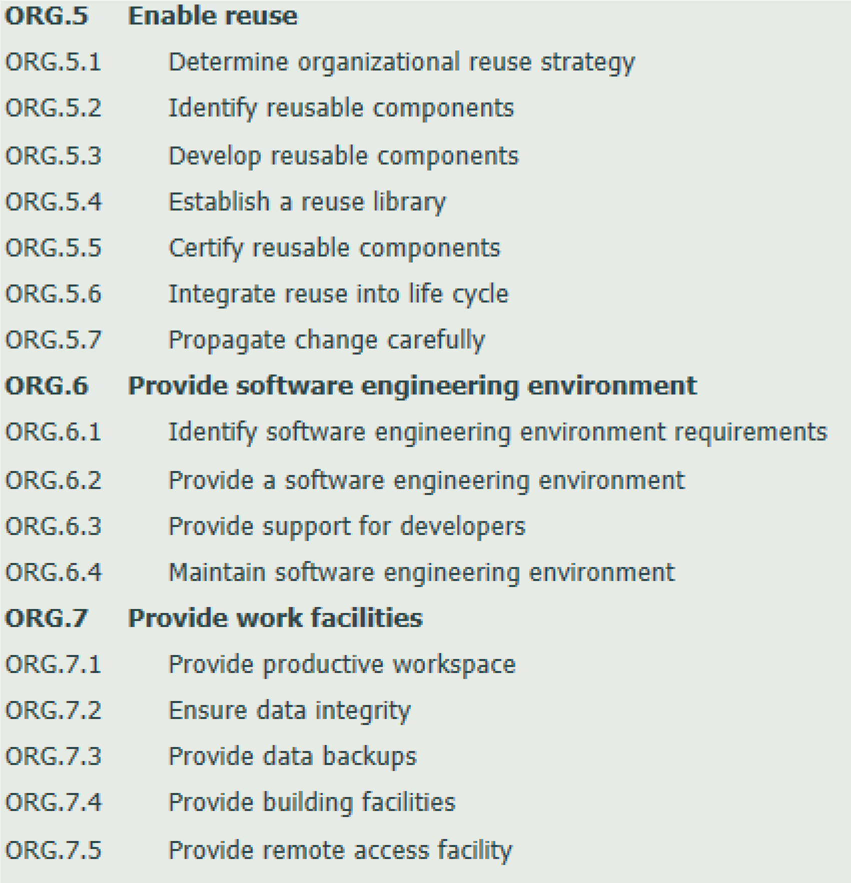
SUP.1	**Develop documentation**
SUP.1.1	Determine documentation requirements
SUP.1.2	Develop document
SUP.1.3	Check document
SUP.1.4	Distribute document
SUP.1.5	Maintain document
SUP.2	**Perform configuration management**
SUP.2.1	Establish configuration management library system
SUP.2.2	Identify configuration items
SUP.2.3	Maintain configuration item descriptions
SUP.2.4	Manage change requests
SUP.2.5	Control changes
SUP.2.6	Build product releases
SUP.2.7	Maintain configuration item history
SUP.2.8	Report configuration status
SUP.3	**Perform quality assurance**
SUP.3.1	Select project standards
SUP.3.2	Review software engineering activities
SUP.3.3	Audit work products
SUP.3.4	Report results
SUP.3.5	Handle deviations
SUP.4	**Perform problem resolution**
SUP.4.1	Prepare problem report
SUP.4.2	Track problem report
SUP.4.3	Prioritize problems
SUP.4.4	Determine resolution
SUP.4.5	Correct the defect
SUP.4.6	Distribute the correction
SUP.5	**Perform peer reviews**
SUP.5.1	Select work products
SUP.5.2	Identify review standards
SUP.5.3	Establish completion criteria
SUP.5.4	Establish re-review criteria
SUP.5.5	Distribute review materials
SUP.5.6	Conduct peer review
SUP.5.7	Document action items
SUP.5.8	Track action items

http://www.rad.fr/spice1.htm

Summary list of base practices



ORG.1	**Engineer the business**
ORG.1.1	Establish strategic vision
ORG.1.2	Deploy vision
ORG.1.3	Establish quality culture
ORG.1.4	Build integrated teams
ORG.1.5	Provide incentives
ORG.1.6	Define career plans
ORG.2	**Define the process**
ORG.2.1	Define goals
ORG.2.2	Identify current activities, roles and responsibilities
ORG.2.3	Identify inputs and outputs
ORG.2.4	Define entry and exit criteria
ORG.2.5	Define control points
ORG.2.6	Identify external interfaces
ORG.2.7	Identify internal interfaces
ORG.2.8	Define quality records
ORG.2.9	Define process measures
ORG.2.10	Document the standard process
ORG.2.11	Establish policy
ORG.2.12	Establish performance expectations
ORG.2.13	Deploy the process
ORG.3	**Improve the process**
ORG.3.1	Identify improvement opportunities
ORG.3.2	Define scope of improvement activities
ORG.3.3	Understand the process
ORG.3.4	Identify improvements
ORG.3.5	Prioritize improvements
ORG.3.6	Define measures of impact
ORG.3.7	Change the process
ORG.3.8	Confirm the improvement
ORG.3.9	Deploy improvement
ORG.4	**Perform training**
ORG.4.1	Identify training needs
ORG.4.2	Develop or acquire training
ORG.4.3	Train personnel
ORG.4.4	Maintain training records



ORG.5	**Enable reuse**
ORG.5.1	Determine organizational reuse strategy
ORG.5.2	Identify reusable components
ORG.5.3	Develop reusable components
ORG.5.4	Establish a reuse library
ORG.5.5	Certify reusable components
ORG.5.6	Integrate reuse into life cycle
ORG.5.7	Propagate change carefully
ORG.6	**Provide software engineering environment**
ORG.6.1	Identify software engineering environment requirements
ORG.6.2	Provide a software engineering environment
ORG.6.3	Provide support for developers
ORG.6.4	Maintain software engineering environment
ORG.7	**Provide work facilities**
ORG.7.1	Provide productive workspace
ORG.7.2	Ensure data integrity
ORG.7.3	Provide data backups
ORG.7.4	Provide building facilities
ORG.7.5	Provide remote access facility

http://www.rad.fr/spice1.htm

IS0 15504 Process Assessment

- For each process in the relevant Process Reference Model:
  - For each set of attribute level criteria
  - Assess whether:
    - N: not achieved 0-15%
    - P: partially achieved >15%-50%
    - L: largely achieved >50%-85%
    - F: fully achieved >85%