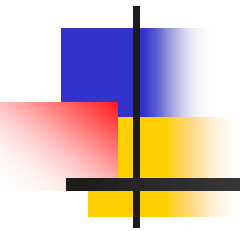
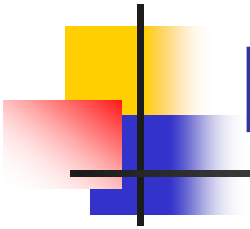


Analyse de besoins et spécifications (LOG410)

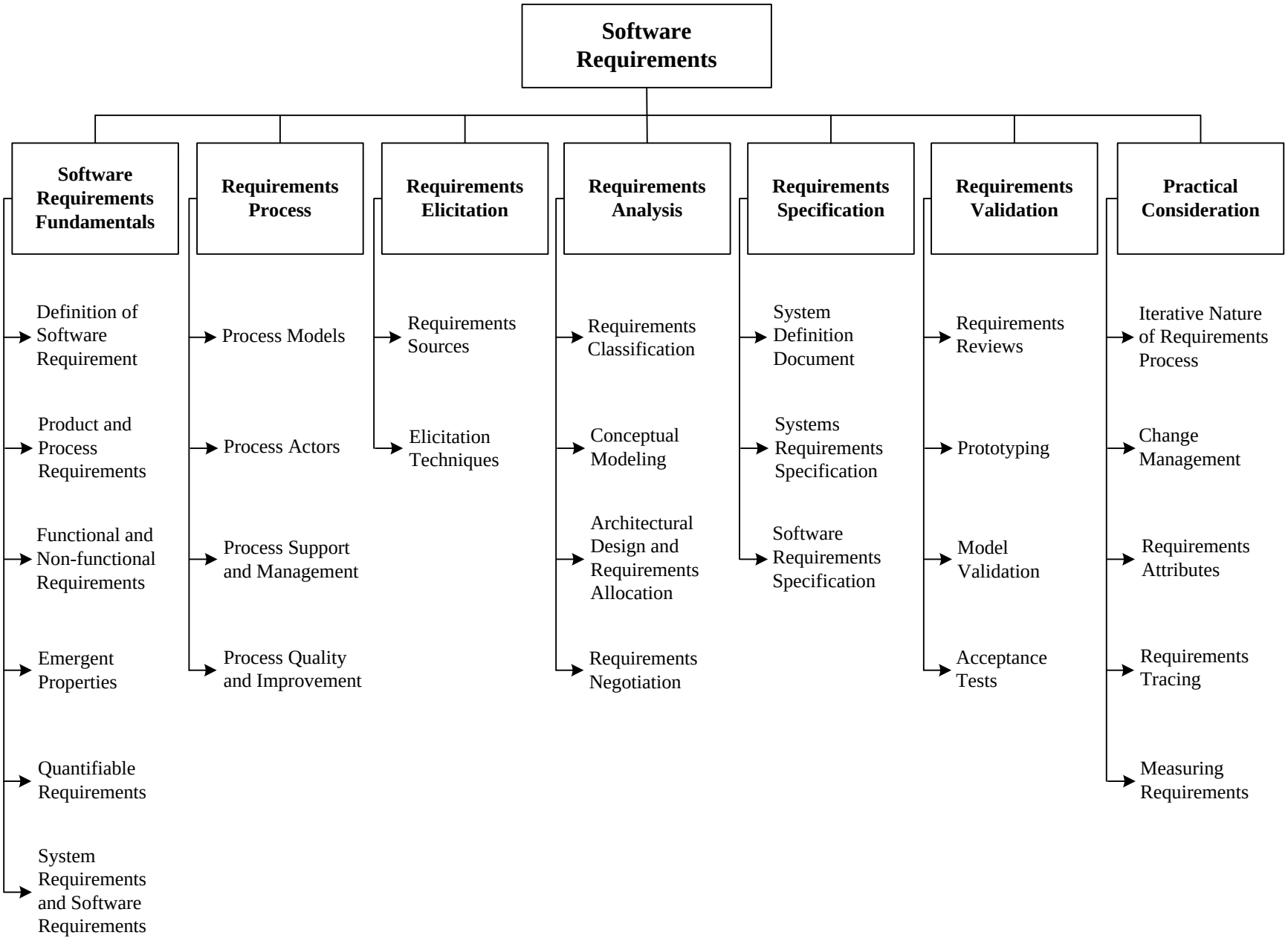
Patrick Tremblay

Séance 13 : 30 Nov 2021



Plan du cours 12

- Gestion des exigences
- Amélioration des processus et exigences logicielles
- Exercice de synthèse et de révision de la matière du cours


Software Requirements
Software Requirements Fundamentals
Definition of Software Requirement
Product and Process Requirements
Functional and Non-functional Requirements
Emergent Properties
Quantifiable Requirements
System Requirements and Software Requirements
Requirements Process
Process Models
Process Actors
Process Support and Management
Process Quality and Improvement
Requirements Elicitation
Requirements Sources
Elicitation Techniques
Requirements Analysis
Requirements Classification
Conceptual Modeling
Architectural Design and Requirements Allocation
Requirements Negotiation
Requirements Specification
System Definition Document
Systems Requirements Specification
Software Requirements Specification
Requirements Validation
Requirements Reviews
Prototyping
Model Validation
Acceptance Tests
Practical Consideration
Iterative Nature of Requirements Process
Change Management
Requirements Attributes
Requirements Tracing
Measuring Requirements

Exercice de synthèse et de révision de la matière vue dans le cours

- Compléter par équipe de deux la grille de correspondance entre la matière vue dans le cours **(acétates présentées en classe, chapitres du livre, autres lectures, normes, laboratoires**) avec la taxonomie (structure) proposée dans le Guide SWEBOK pour Software Requirements
- Discussion en plénière

Conclusion de l'exercice

- "The hardest single part of building a software system is deciding precisely what to build. No other part of the conceptual work is as difficult as establishing the detailed technical requirements, including all the interfaces to people, to machines, and to other software systems. No other part of the work so cripples the resulting system if done wrong. No other part is more difficult to rectify later.

- Therefore, the most important function that the software builder performs for the client is the iterative extraction and refinement of the product requirements."

- Source: Brooks, F.P.: No Silver Bullet : Essence and Accidents of Software Engineering"; IEEE Computer, vol. 20, no. 4, pp. 10-19, April 1987

Amélioration de processus et exigences logicielles

Enseignant: Patrick Tremblay

Notes élaborées par le Professeur Pierre Bourque
Révisées par F.Coallier

École de technologie supérieure
Département de génie électrique

Carnegie Mellon
Software Engineering Institute

Pittsburgh, PA 15213-3890

Source de ces acétates

CMMI V1.1 Tutorial

E-SEPG

April 9, 2002

Mike Phillips, CMMI Program Manager

http://www.sei.cmu.edu/cmmi/presentations/euro-sepg-tutorial/

Sponsored by the U.S. Department of Defense

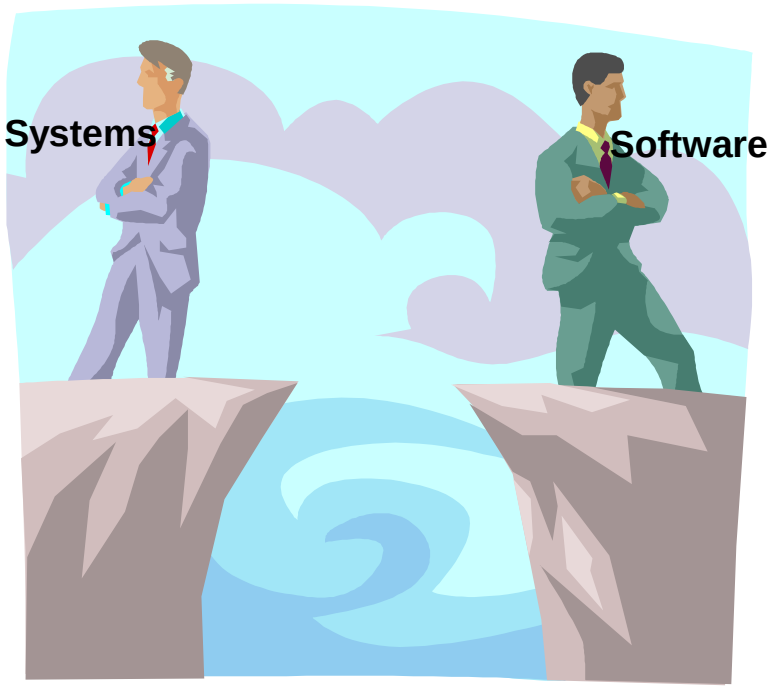
École de technologie supérieure
Département de génie électrique

Commonly Used CMMs

Software CMM	staged	software development
System Engineering CMM	continuous	system engineering
System Engineering Capability Model	continuous	system engineering
Software Acquisition CMM	staged	software acquisition
System Security Engineering CMM	continuous	security engineering
Personal Software Process	staged	individual software development
FAA-iCMM	continuous	software engineering, systems engineering, and acquisition
IPD-CMM	hybrid	integrated product development
People CMM	staged	workforce
SPICE Model	continuous	software development

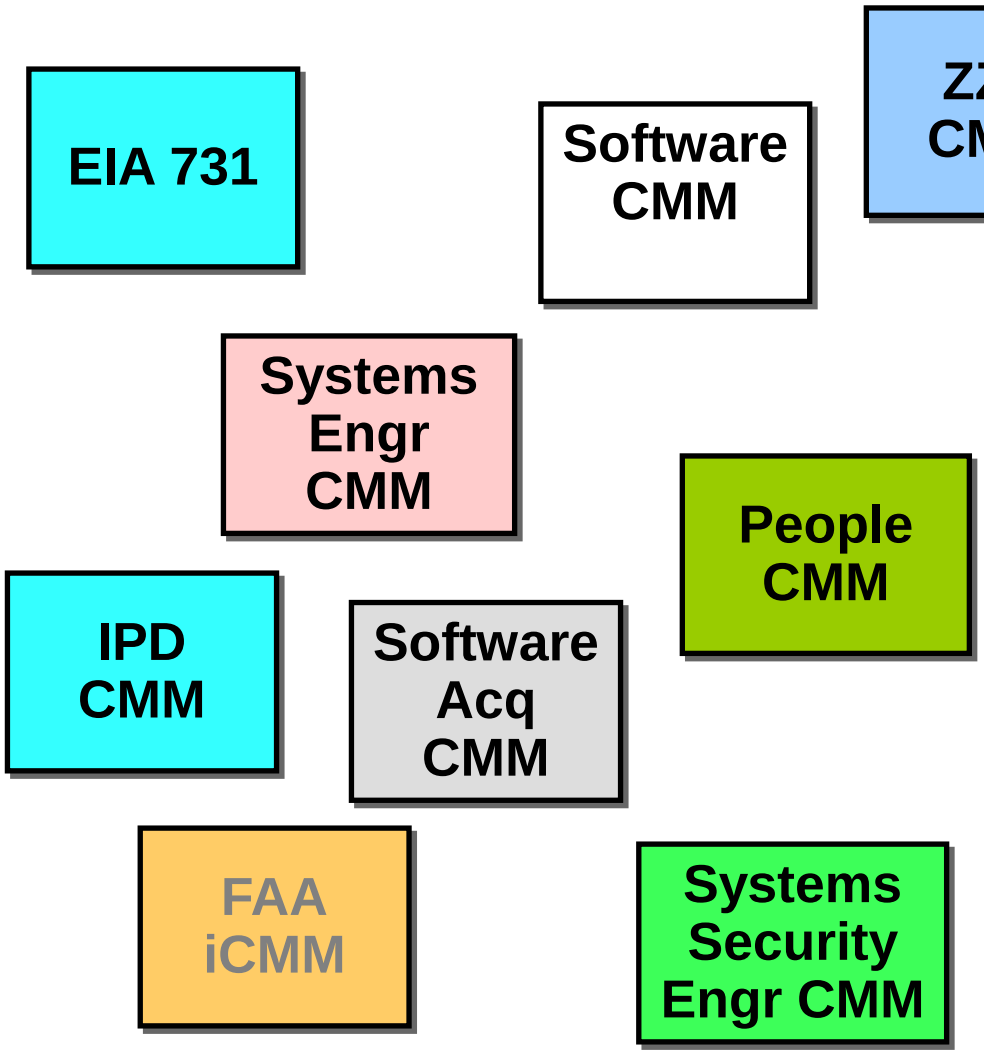
The Problem

- Systems and software disciplines have traditionally not been well integrated
- The importance of software in systems has increased dramatically
  - Example: % of requirements allocated to software: *
    - B-2 -- 65%
    - F-22 -- 80%
- The DOD has emphasized the need to make the systems/software interface more seamless



* Source: Standish Group *Chaos Report*

So Many Models, So Little Time



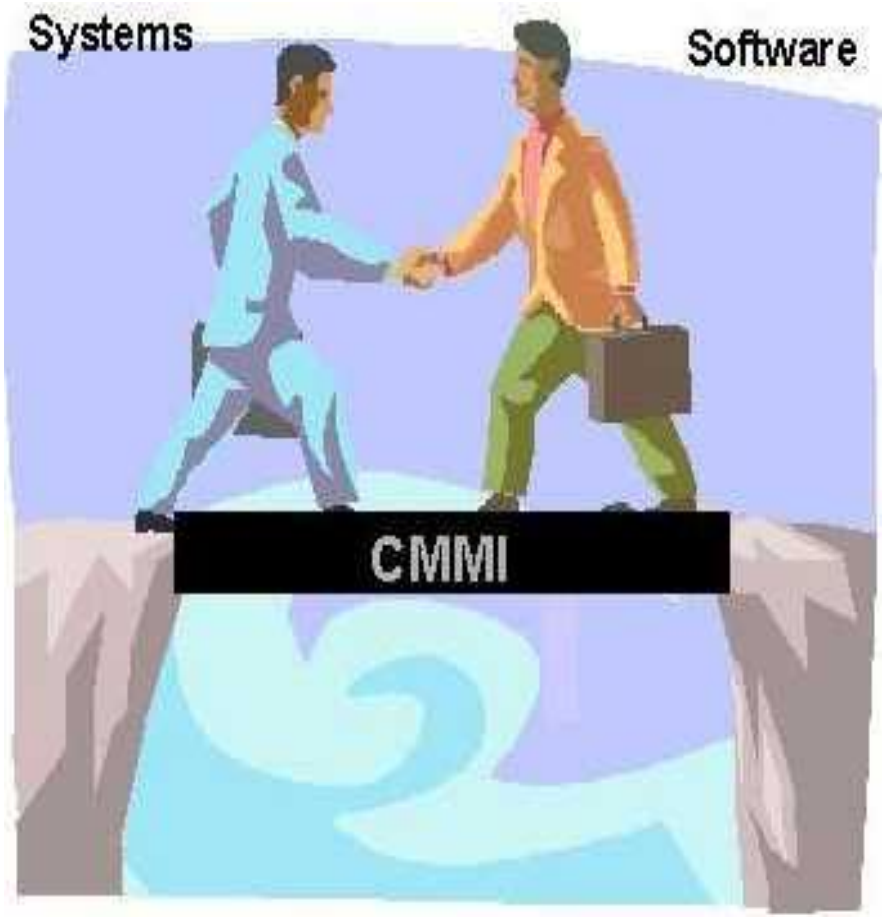
- **Different structures**, formats, terms, ways of measuring maturity
- **Causes confusion**, especially when using more than one model
- **Hard to integrate** them in a combined improvement program
- **Hard to use multiple models** in supplier selection

CMMI to the Rescue!

- Integrates systems and software disciplines into one process improvement framework.
- Provides a framework for introducing new disciplines as needs arise.

Bridging the Divide

- Systems engineering and software engineering processes are integrated.
- Integrates systems and software disciplines into one process improvement framework.
- Provides a framework for introducing new disciplines as needs arise.



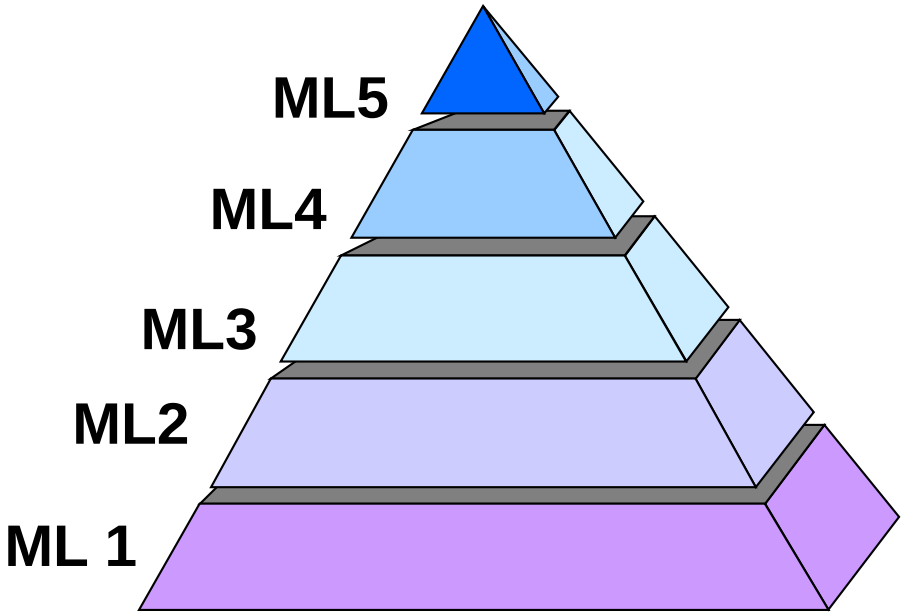
The CMMI Project

- DoD sponsored collaboration between industry, Government, SEI

- U.S. Army, Navy, Air Force
- Federal Aviation Administration
- National Security Agency
- Software Engineering Institute
- ADP, Inc.
- AT&T Labs
- BAE
- Boeing
- Computer Sciences Corporation
- EER Systems
- Ericsson Canada
- Ernst and Young
- General Dynamics
- Harris Corporation
- Honeywell
- KPMG
- Lockheed Martin
- Motorola
- Northrop Grumman
- Pacific Bell
- Q-Labs
- Raytheon
- Reuters
- Rockwell Collins
- SAIC
- Software Productivity Consortium
- Sverdrup Corporation
- TeraQuest
- Thomson CSF
- TRW

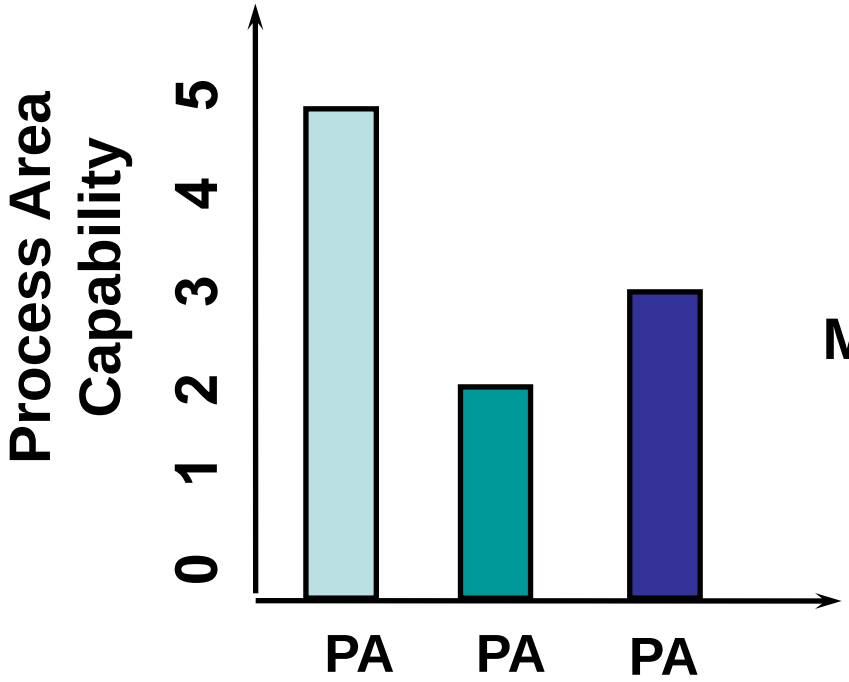
Comparing Model Representations

Staged



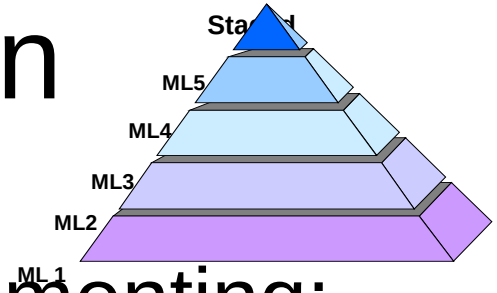
. . .for an established set of process areas across an organization

Continuous



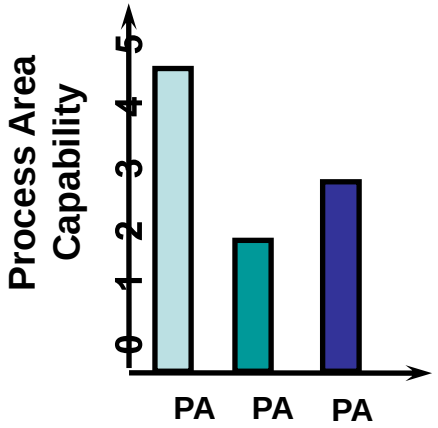
. . .for a single process area or a set of process areas

Advantages of the Staged Representation



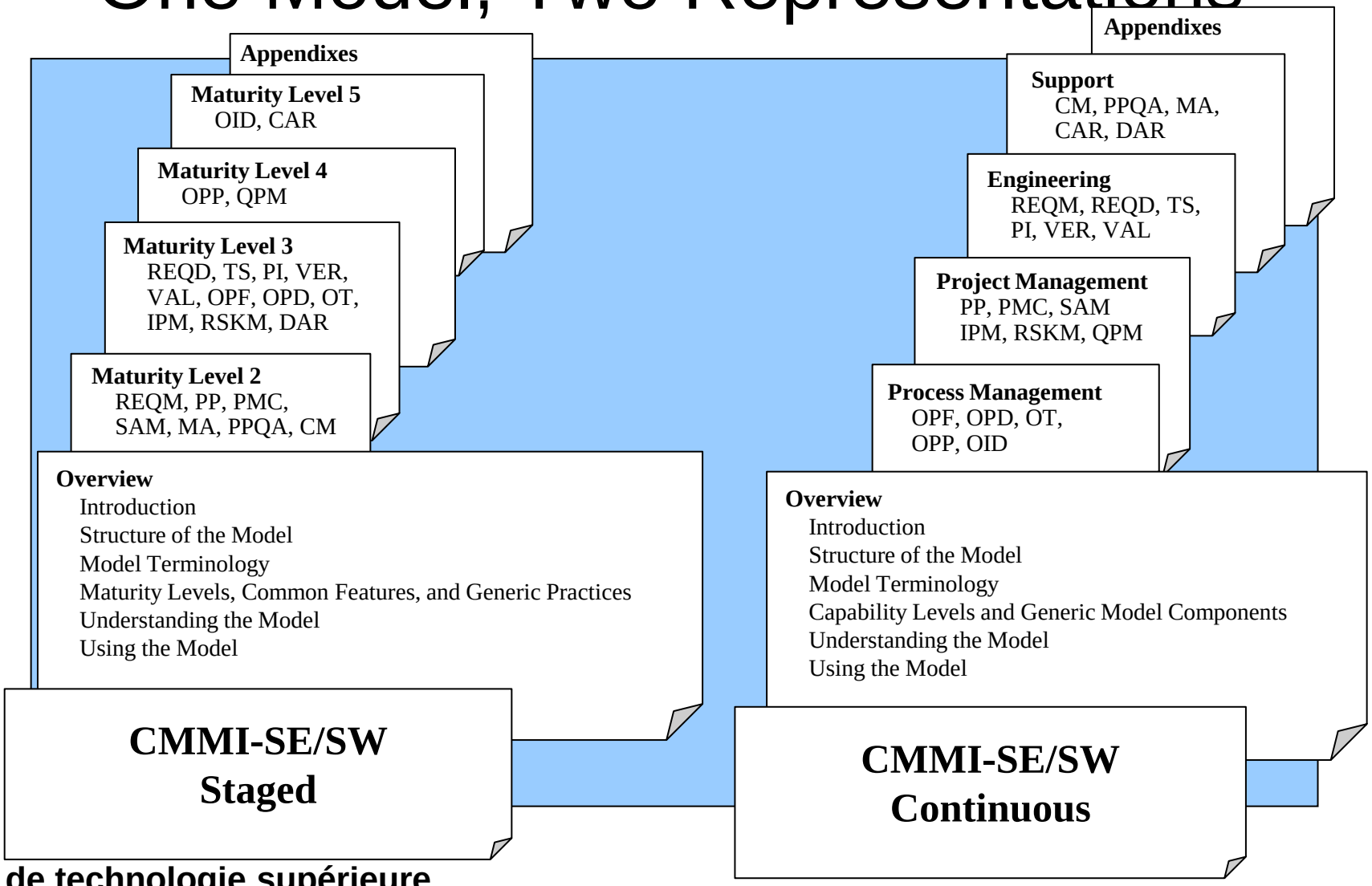
- Provides a roadmap for implementing:
  - groups of process areas
  - sequencing of implementation
- Familiar structure for those transitioning from the SW-CMM

Advantages of the Continuous Representation

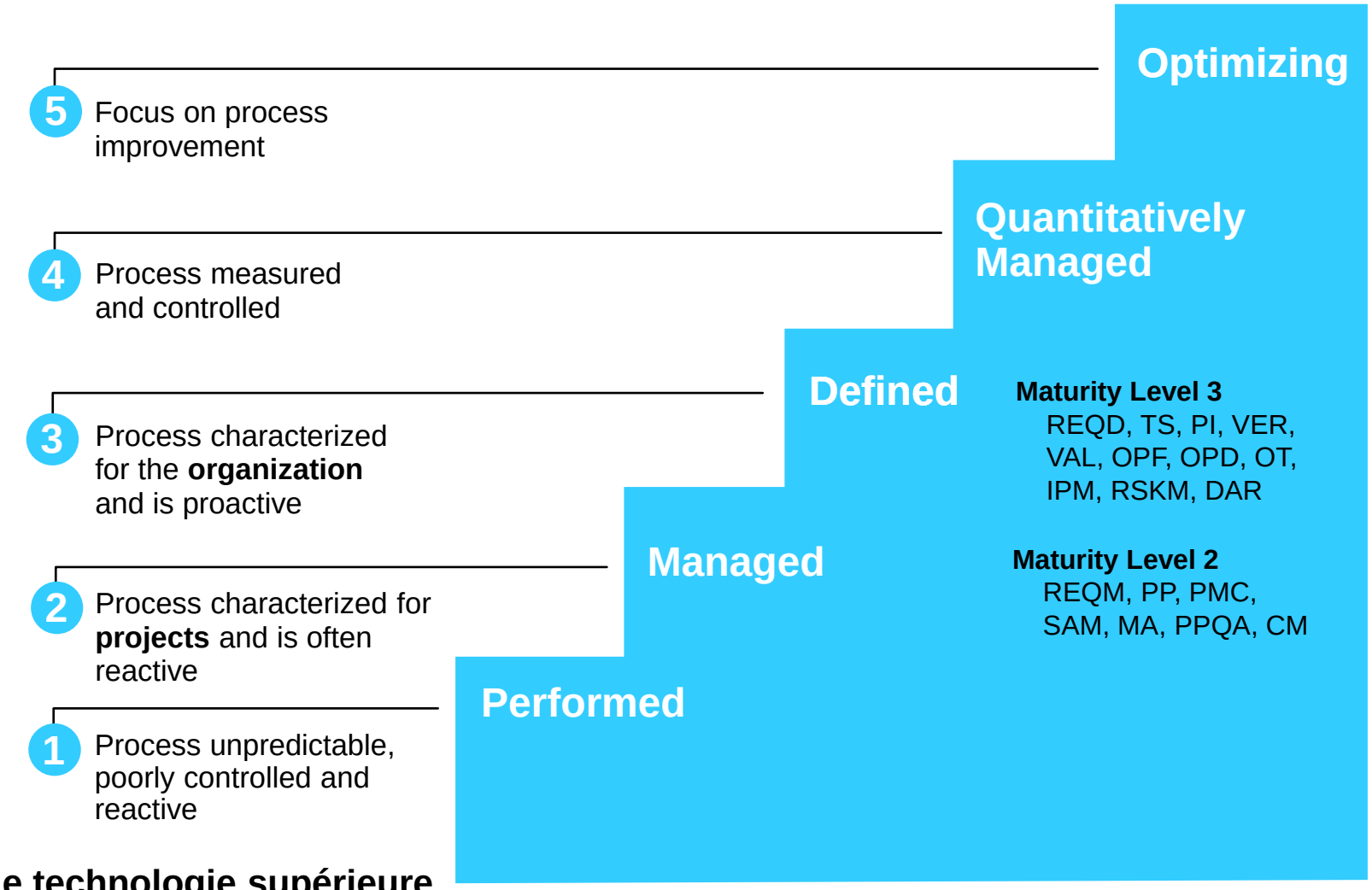


- Provides maximum flexibility for focusing on specific process areas according to business goals and objectives.
- Familiar structure for those transitioning from the systems engineering community.

CMMI Structure
One Model, Two Representations



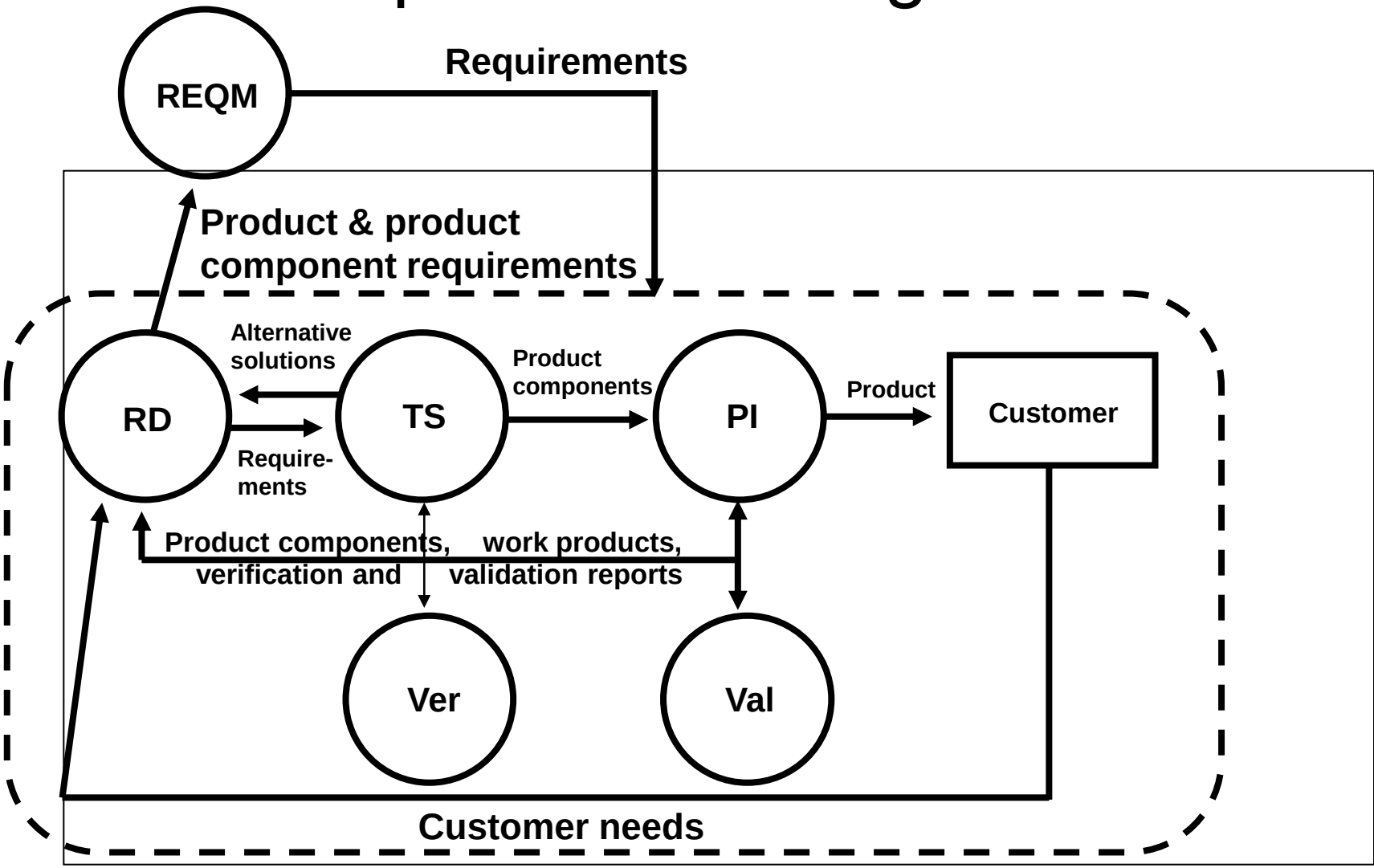
The Maturity Levels (staged)



Engineering Process Areas

- There are six (6) Engineering Process Areas.
  - Requirements Management
  - Requirements Development
  - Technical Solution
  - Product Integration
  - Verification
  - Validation

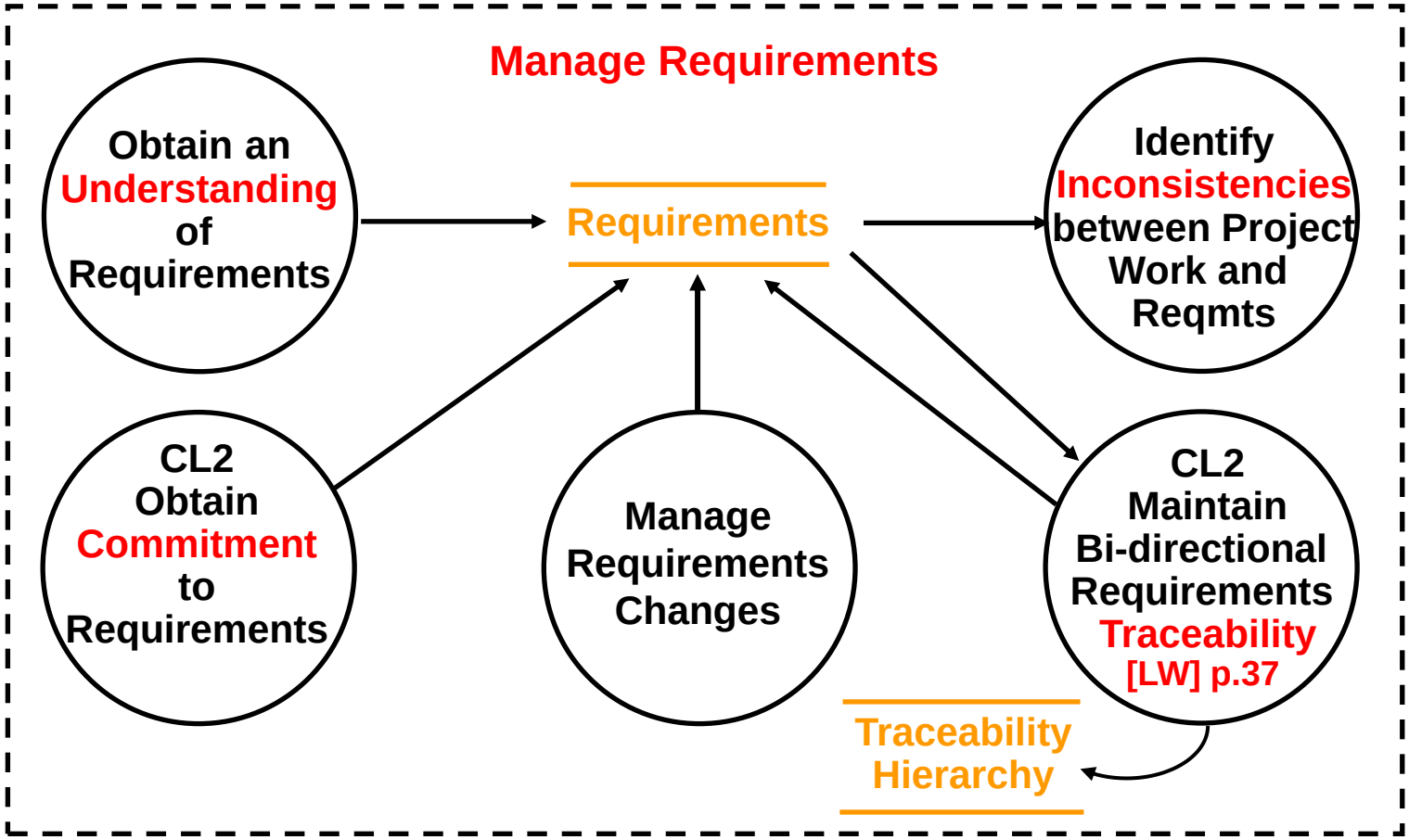
Engineering Process Areas
“Zones de processus d’ingénierie”



Requirements Management

- Purpose:
  - Manage the requirements of the project's product and product components and identify inconsistencies between:
    - those requirements and;
    - the project's plans and work products.

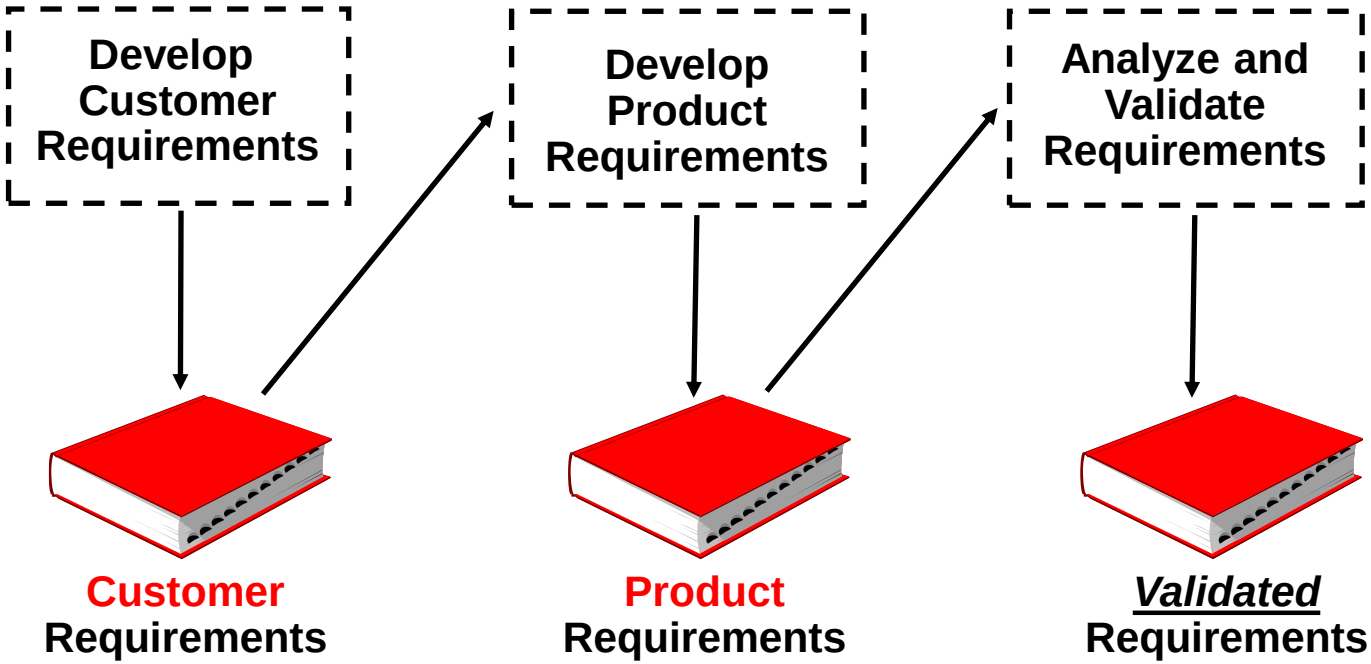
Requirements Management Context



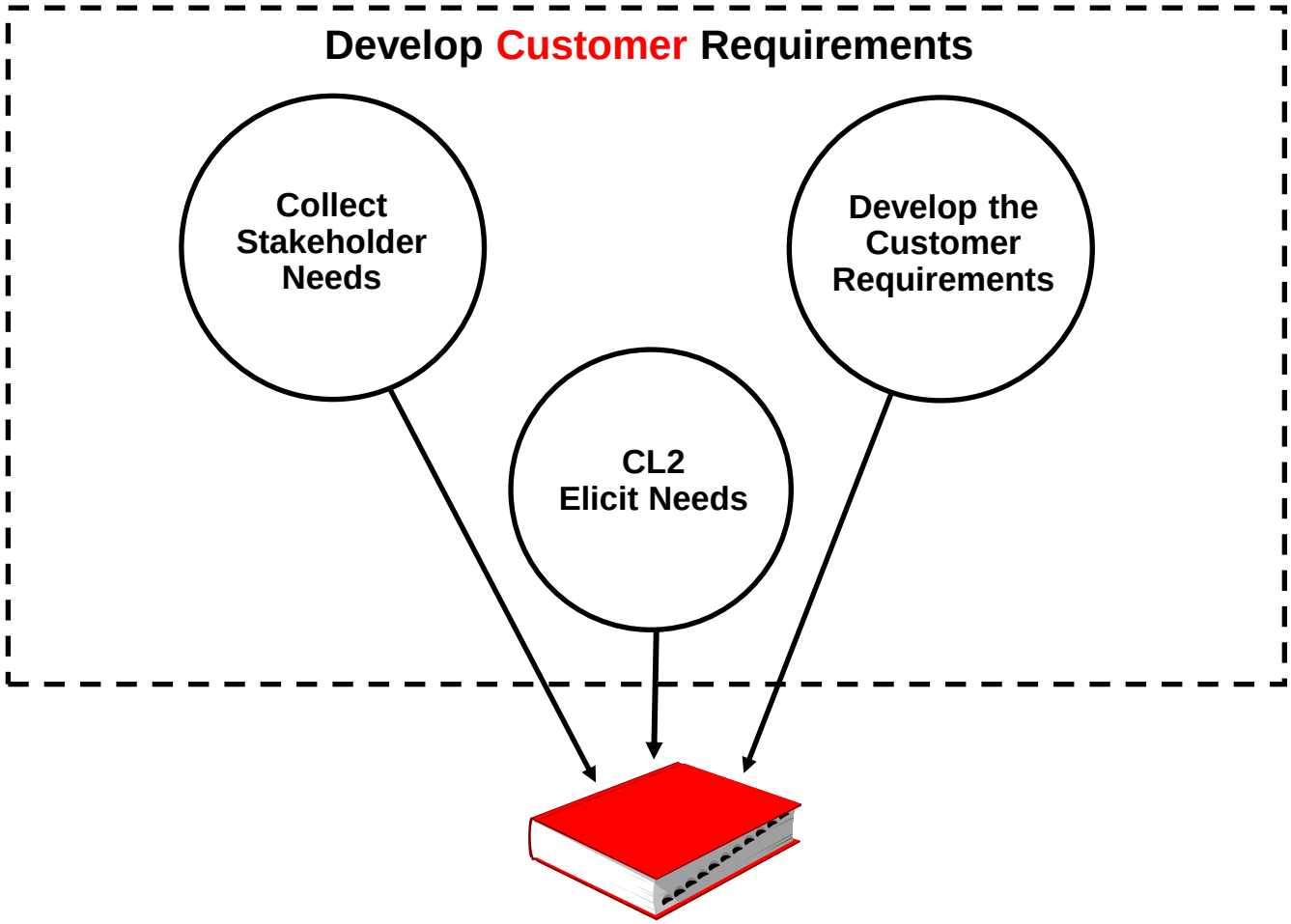
Requirements Development

- Purpose:
  - Produce and analyze requirements of:
    - customer;
    - product and;
    - product component.

Requirements Development - Context

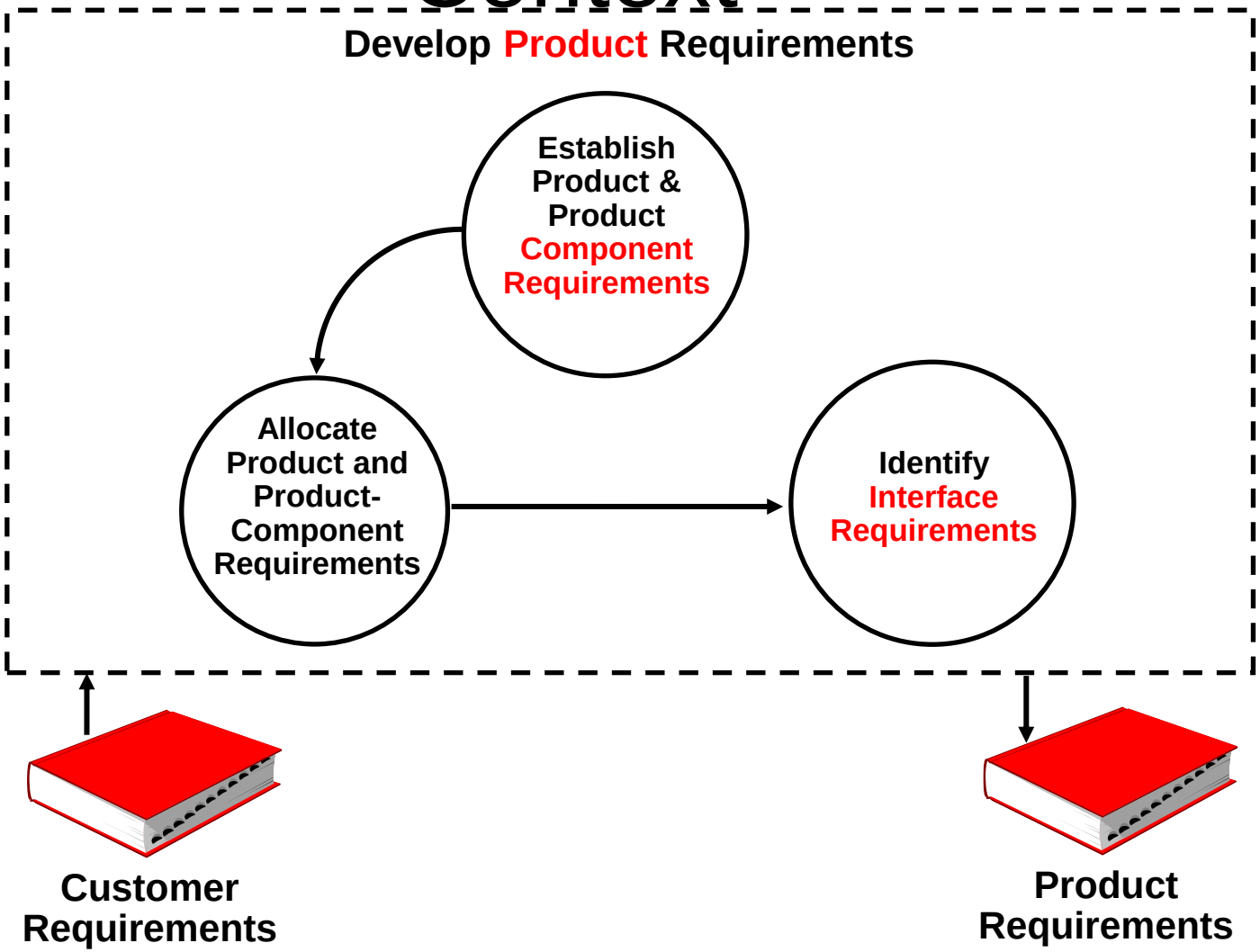


Requirements Development Context

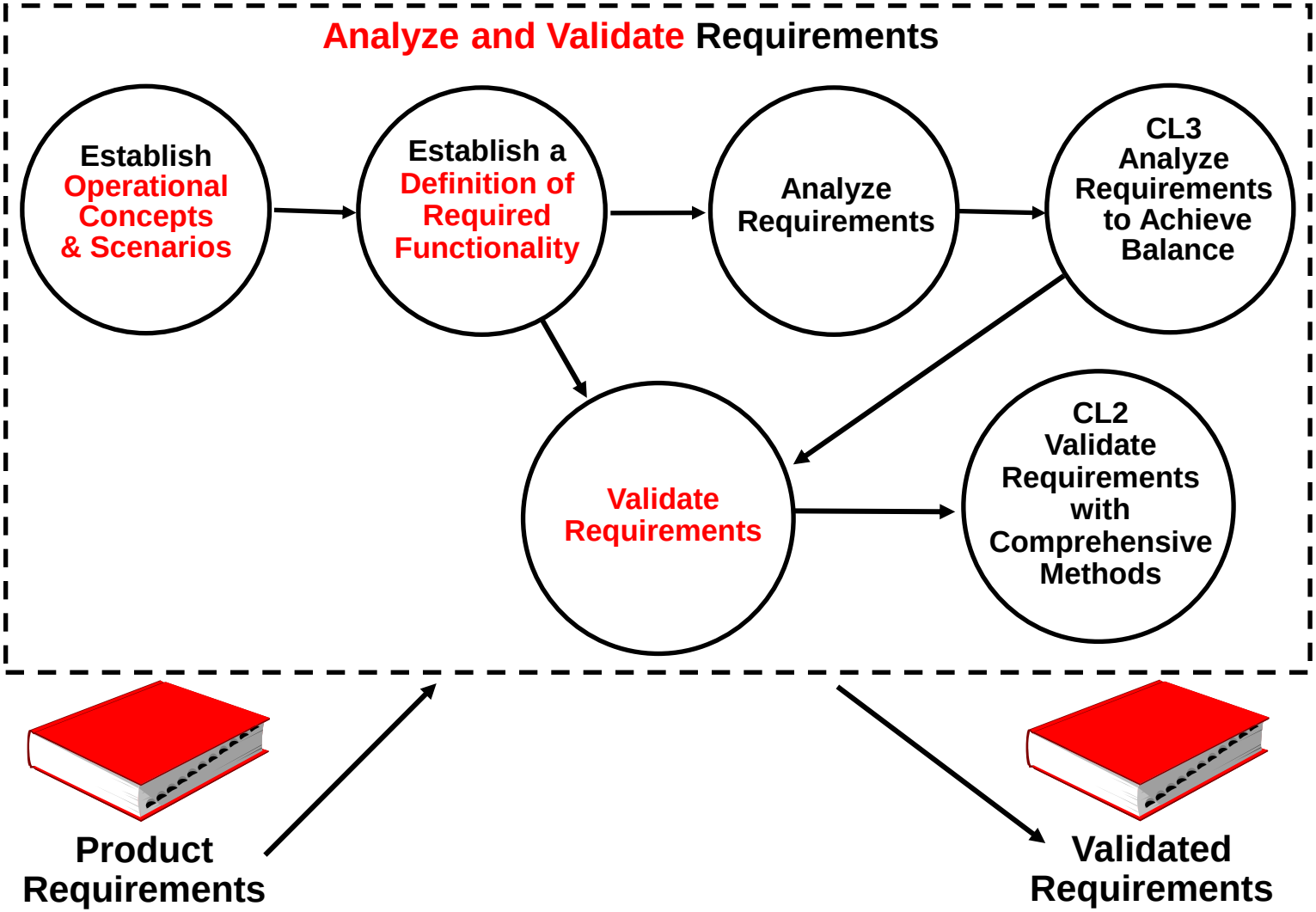


École de technologie supérieure
Département de génie électrique

Requirements Development Context

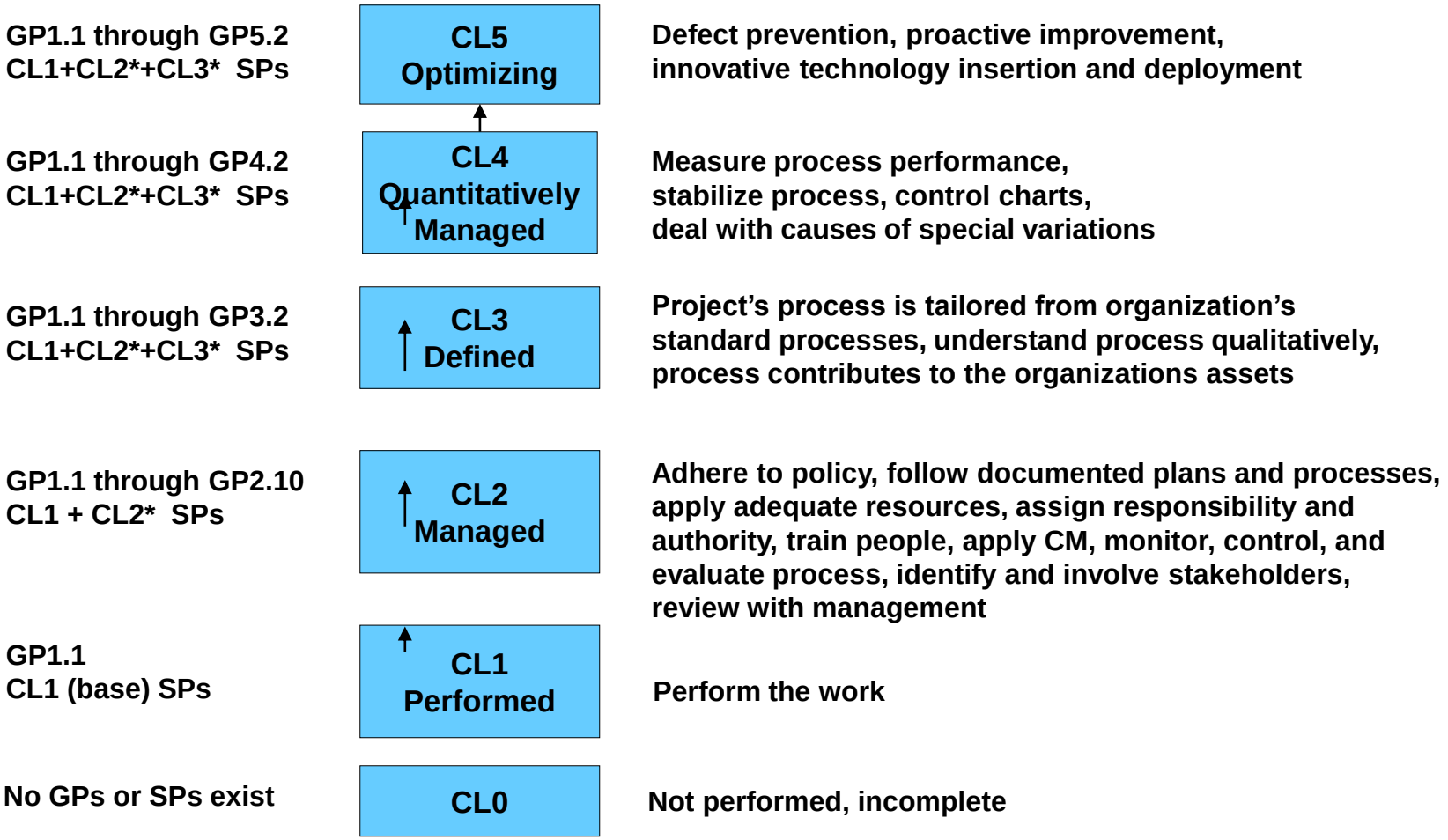


Requirements Development Context



École de technologie supérieure
Département de génie électrique

Improving a Process Area



*** Advanced practices exist only in the Engineering PAs.**

École de technologie supérieure
Département de génie électrique

REQM - Capability Levels 1 & 2

Requirements Management

Specific practices (CL1 - "base")

SP1.1-1: Obtain an Understanding of Requirements
SP1.3-1: Manage Requirements *Changes*
SP1.5-1: Identify *Inconsistencies* Between Project Work and Requirements

Generic practices (CL1)

GP1.1: Perform Base Practices

Specific practices (CL2 - "advanced")

SP1.2-2: Obtain Commitment to Requirements
SP1.4-2: Maintain Bi-directional Traceability of Requirements

Generic practices (CL2)

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: Manage Configurations
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

REQM - Capability Level 3

Requirements Management

Specific practices (CL1 & CL2)

SP1.1-1: Obtain an Understanding of Requirements
SP1.2-2: Obtain Commitment to Requirements
SP1.3-1: Manage Requirements Changes
SP1.4-2: Maintain Bi-directional Traceability of Requirements
SP1.5-1: Identify Inconsistencies Between Project Work and Requirements

Generic practices (CL1 & CL2)

GP1.1: Perform Base Practices
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: Manage Configurations
GP2.7: Identify and Involve Relevant Stakeholders
GP2.8: Monitor and Control the Process
GP2.9: Objectively Evaluate Adherence
GP2.10: Review Status w/Higher Level Management

Specific practices (CL3)

All the CL1 & CL2 Specific Practices

Generic practices (CL3)

All the CL1 & CL2 Generic Practices plus(+):

GP3.1: Establish a Defined Process
GP3.2: Collect Improvement Information
ex.: « Leçons apprises »

REQM - Capability Levels 4 & 5

Requirements Management

Specific practices (CL4)

All the CL1 & CL2 Specific Practices

Generic practices (CL4)

All the CL1 & CL2 & CL3 Generic Practices plus(+):

GP4.1: Establish Quantitative Objectives for the Process
GP4.2: Stabilize Subprocess Performance

Specific practices (CL5)

All the CL1 & CL2 Specific Practices

Generic practices (CL5)

All the CL1 & CL2 & CL3 & CL4 Generic Practices plus(+):

GP5.1: Ensure Continuous Process Improvement
GP5.2: Correct Root Causes of Problems

Analyse de besoins et spécifications (LOG410)

Gestion des changements

École de technologie supérieure
Département de génie électrique

Le changement est inévitable

- Où que l'on soit dans le cycle de vie du système, les demandes de changement vont se présenter.
- Prendre en charge ces changements représente un défi majeur de gestion.

Gestion des changements
(Chapitre 28 de [LW])

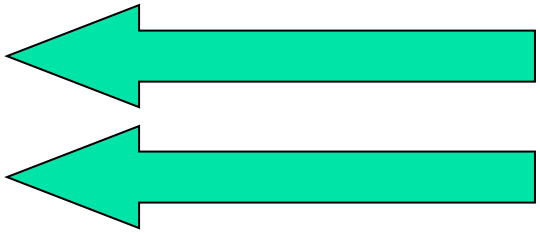
- Les changements sont inévitables
- Le processus de gestion:
  - Planifier
  - Établir les références (le 'baseline')
  - Organiser le contrôle des changements
  - Gérer les changements depuis le document de vision
- Gestion de configuration

Software Configuration Management

- SCM – identifies software configuration of a system
  - Controls changes to the configuration
  - Maintains integrity and traceability of the configuration
- Change Control
  - Management of change as one part of the SCM process
- Version Control
  - Management of the product versions generated as part of the SCM process
- Release Control
  - Transformation of configuration items into a deliverable product

Software Configuration Items

- Source Code
- Design Documents
- Requirements
- Test Cases
- Measurement Data
- Program Trouble Reports (BTS)
- Manuals and Tutorials



Software Configuration

- Four (4) software configuration management functions:
  - Identification
  - Control
  - Auditing
  - Status Accounting

Implémentation dans RUP et la suite d'outils Rational

Summary of UCM Roles

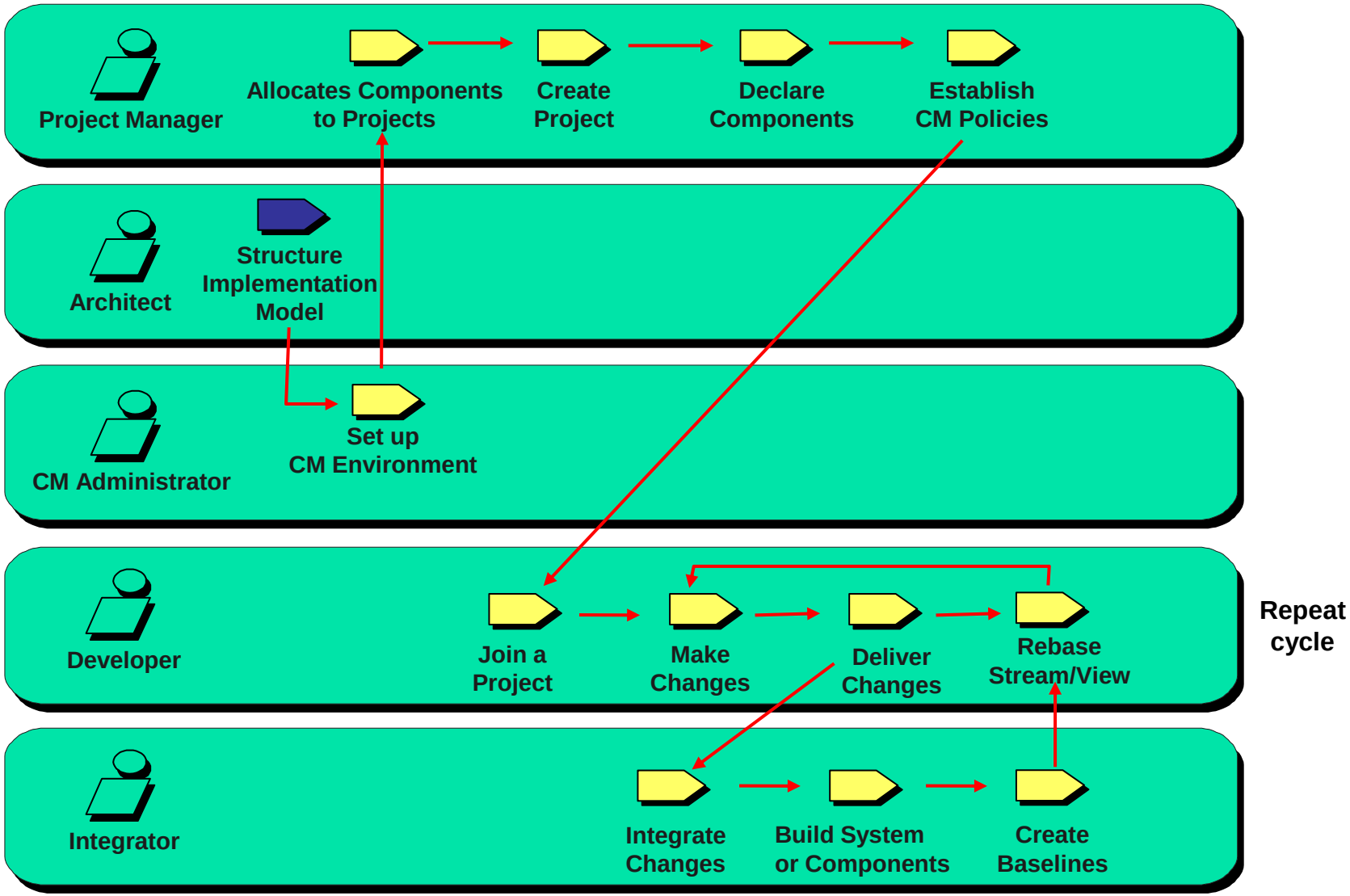
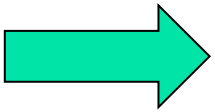


Table des matières



- Introduction – la suite Rational
- Gestion de configuration
- Gestion du changement

What is ClearQuest?

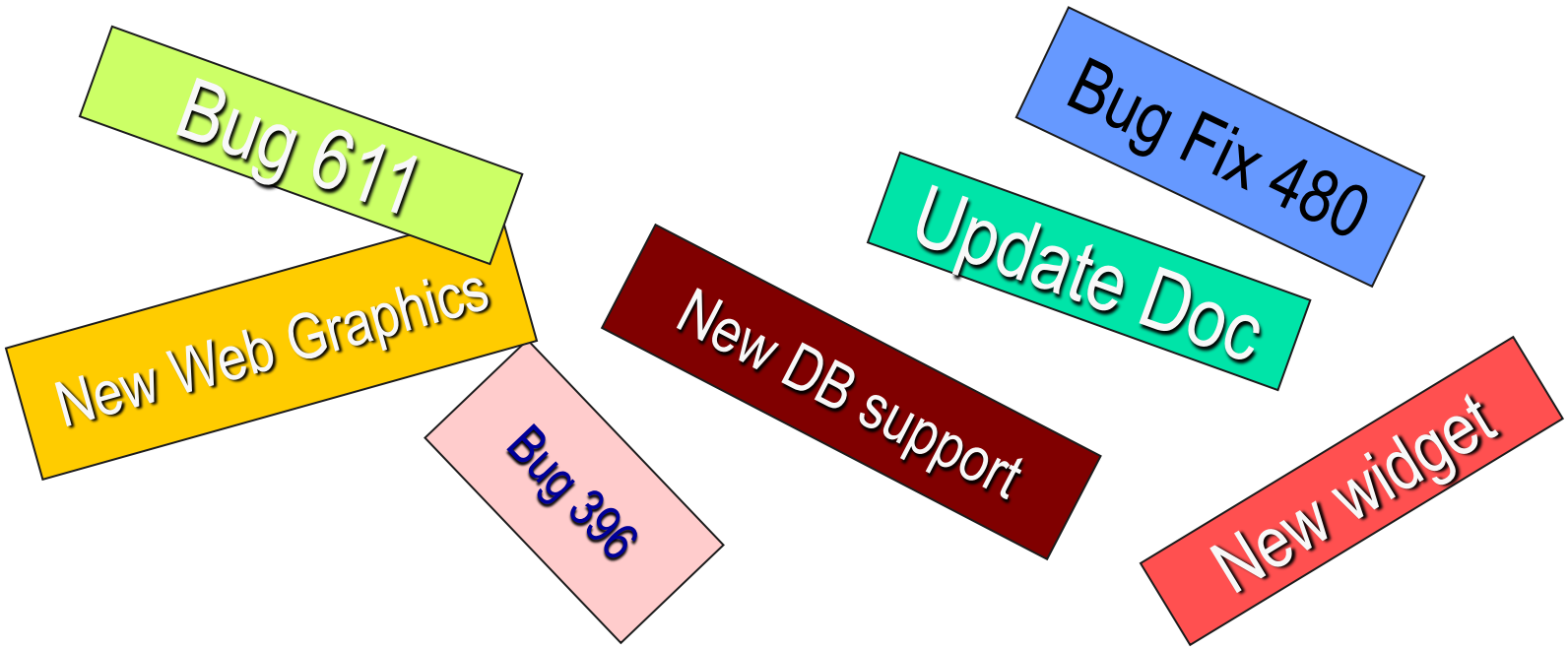
Rational ClearQuest is a customizable defect/change request management system that scales to the needs of any size project.

What is ClearQuest?

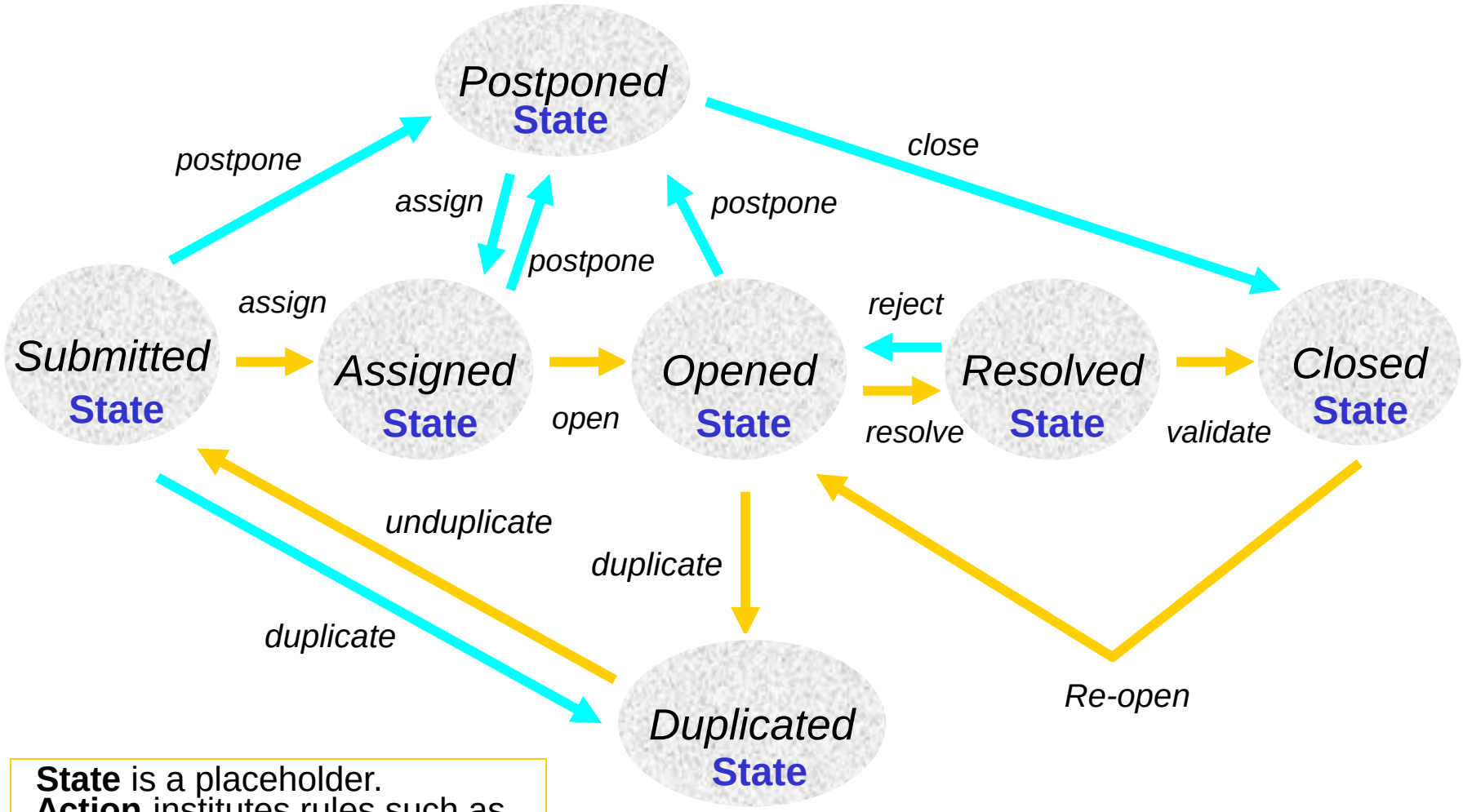
How ClearQuest Can Help Your Change Management Needs

ClearQuest Activities

- Change activities include:
  - Enhancement requests
  - Defect reports
  - Documentation modifications



Sample Process Model for Defect Tracking

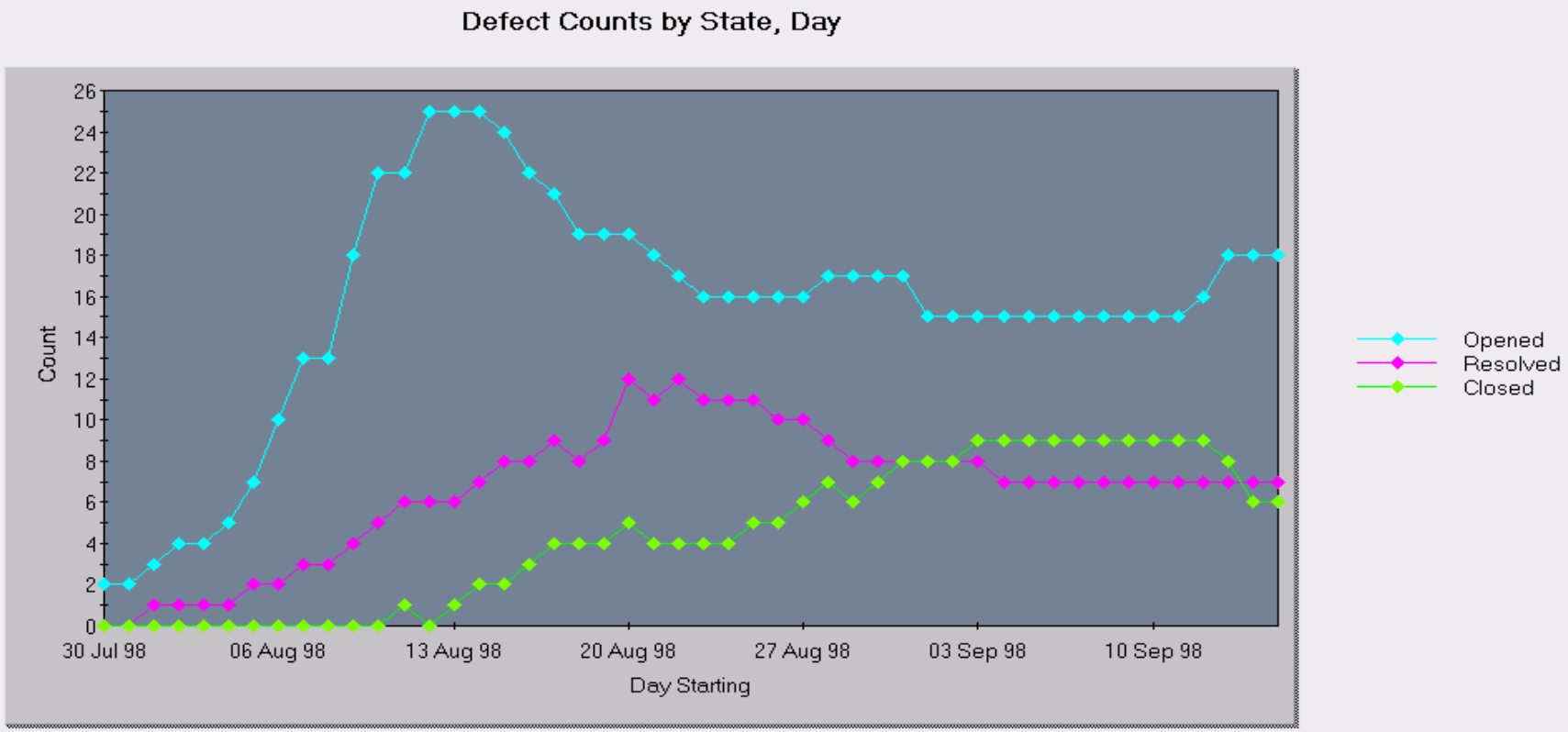


State is a placeholder.
Action institutes rules such as state transition.

Trend Chart

date_value	record_count	new_state
7/30/1998	2	Opened
7/31/1998	2	Opened
8/1/1998	3	Opened
8/1/1998	1	Resolved
8/2/1998	4	Opened

Result set | Query editor



Aging Chart

record_count	new_state	priority
8	Opened	
1	Opened	1-Resolve Immediately
5	Opened	2-Give High Attention
3	Opened	3-Normal Queue

Result set | Query editor