Software Quality Assurance

Software Engineering

Wei Xing – 2016

Introduction
Software Quality Assurance

Software Crisis

- 40% of projects failing, 50% overcome budget
- Delays…
- Small amount of features actually used
- Below 50% of IT investments impact organizations profit (source: London School of Economics)
- Poor quality, rework

"When you're out of quality, you're out of business."

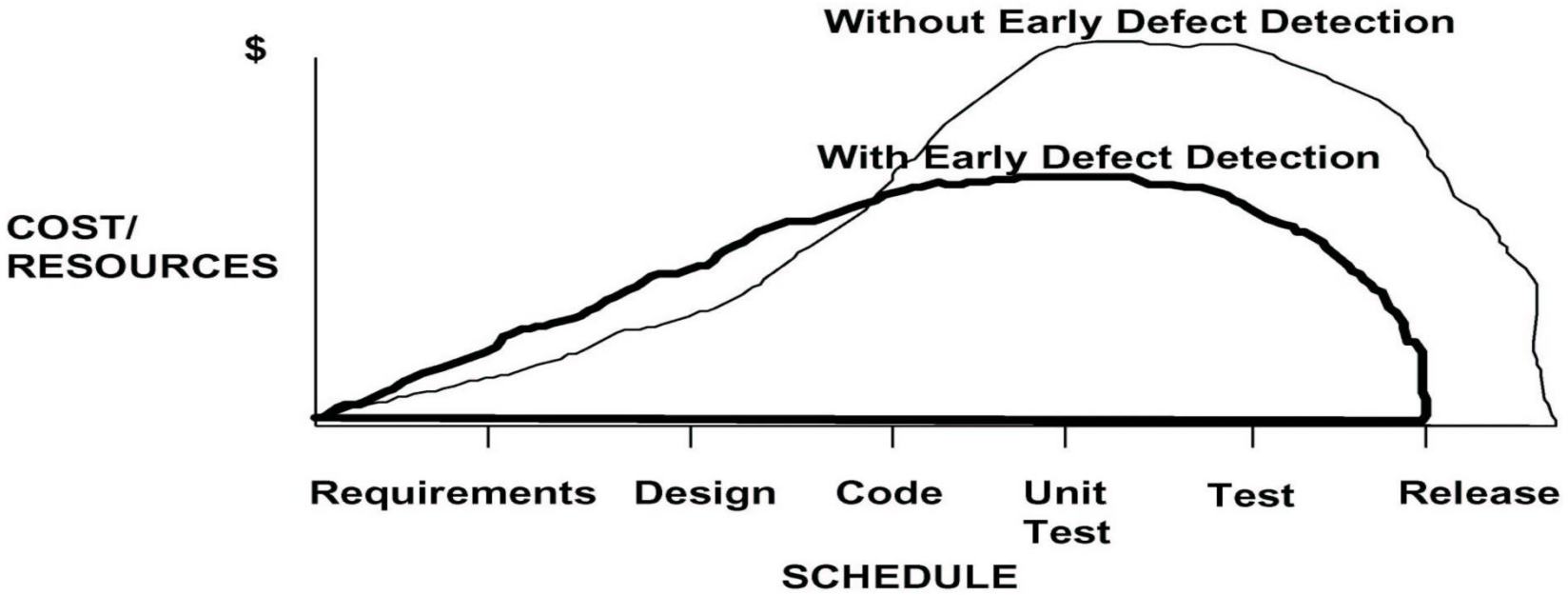
THE COST OF QUALITY		
SIGMA LEVEL	DEFECTS PER MILLION OPPORTUNITIES	COST OF QUALITY
2	308,537 *(Noncompetitive companies)*	Not applicable
3	66,807	25 – 40% of sales
4	6,210 *(Industry average)*	15 – 25% of sales
5	233	5 – 15% of sales
6	3.4 *(World class)*	< 1% of sales
Each sigma shift provides a 10 percent net income improvement.		

Immature Software

Manufacturing

Source: M. Harry and W. Schroeder, *Six Sigma*, 2002

Benefits of Software Quality Management



•Adapted from Fagan, "Advances in Software Inspections", IEEE, July 1986

Benefits of Software Quality Management

Benefit from customer perspective,

- Customer satisfaction
- Improved software reliability
- Reduced errors during software operation
- Meeting the customer's requirements

Benefit from organization perspective,

- The customer's requirements are met
- Requirements are stable
- Feature requirements have been implemented and verified
- Processes are applied in a consistent manner
- The process improves over time

Class Mission

Targets

- Set up basic ideas on software quality management
- Master （熟练掌握）
  - Main Technology of Software quality management
  - Peer review
  - CMMI/Agile framework quality assurance process
  - Defect Management

At the end of this class, you will

- Have an overall idea and structure of software quality assurance
- Have clue of how to improve product efficiency and quality of the whole project in the later study and work

Class Allocations

- Total 5 units and 32 clauses （ CW1-16 ）
- 3 exercises practices

Class Allocation

No	Unit	Knowledge Points	Requirements	Hours
1	Generial Knowledge	Introduction of software quality management	Know	2
		Main Technology of Software quality management	Master	
2	Quality Control	Concepts and classfictions of peer review	Know	12
		Peer review process & Mearsurment	Master	
3	Qulality Assurance	International quality management model: CMMI	Master	10
		International quality management model: Agile	Master	
		Quality assurance process	Understand	
		Personal software process	Understand	
4	Senior Quality Management	Defect management	Master	4
		Defect analysis	Understand	
5	Process Audit	Software process audit	Understand	4
			Understand	

Exercise – Peer Review

Choose one of below items to do peer review

- Codes
- Document
- Product

Requirements:

- Organize review team (6-7 person each team) and define roles.
- Choose a proper review method
- Create a process map of chosen review method

Presentation:

- Each team have 5 minutes to present the review result and lesson & learn

Exercise - Scrum Practice

Create a introduction manual for New Opened Restaurant/Hotel (5 Min)

- Create Brand and Logo for new Restaurant/Hotel
- Set up Price
- Address location and transportation
- Design unique features for new Restaurant/Hotel
- Design target level of service and slogan

Project Execution (40 Min)

- Sprint 0: Preparation (5Min)
  - Organize Team (Product Owner, Scrum Master)
  - Product Owner discuss product vision and objective
  - Create Backlog
- Two days Sprint (30 Min)
  - Sprint Planning meeting (10 Min)
  - First Day (10 Min)
    - Stand-up meeting (3 Min)
    - Work (7 Min)
  - Second Day(10 Min)
    - Stand-up meeting (3 Min)
    - Work (7 Min)
- Sprint Review Meeting (5 Min)
- Sprint Retrospective Meeting (5 Min)

Exercise - Process audit

Project Execution (40 Min)

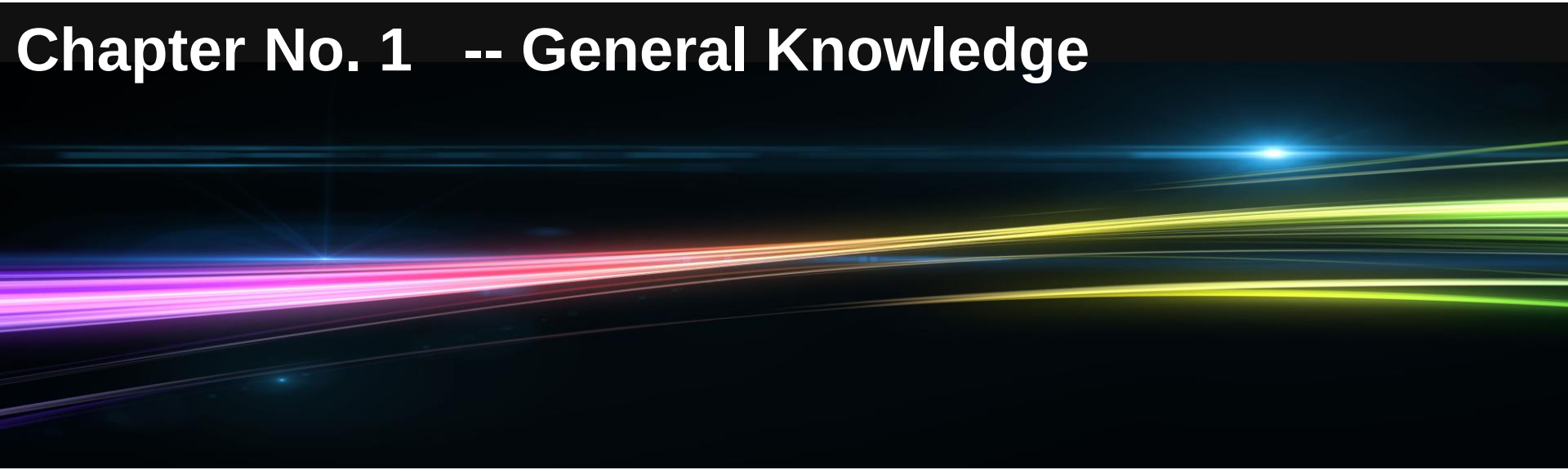
- Case introduction
- Audit planning
- Audit execute
- Audit report
- Corrective action / follow up

Assessment

Final Score

- Daily Performance (10%, Class Attendance + Discussion)
- Score from 3 exercises practice (55%)
  - PR 15%
  - Agile 15%
  - Process audit 25%
- Score from term paper (35%)
  - The final paper and case score from the class 综合实践
    - 过程及其过程裁剪
    - 里程碑与质量管控点
    - 工作结果质量评价标准
    - 项目管理KPI

Chapter No. 1 -- General Knowledge



Class Allocation

- Chapter No. 1 -- General Knowledge

- Quality Concept
  - Definition
  - Quality degrees
  - QA VS QC

- Quality System/Standards and Models
- Quality Management Framework

Definition of Software Quality

Software quality is WHAT? :

The word quality is often used indiscriminately for many different meanings. Quality can be defined as “fitness for use,” “customer satisfaction,” “doing things right the first time,” or “zero defects.”

These definitions are acceptable because quality can refer to degrees of excellence.”

The simplest definition is given by Crosby (1979): “Quality is conformance to requirement”

SW Quality Concept

CMMI 1.3

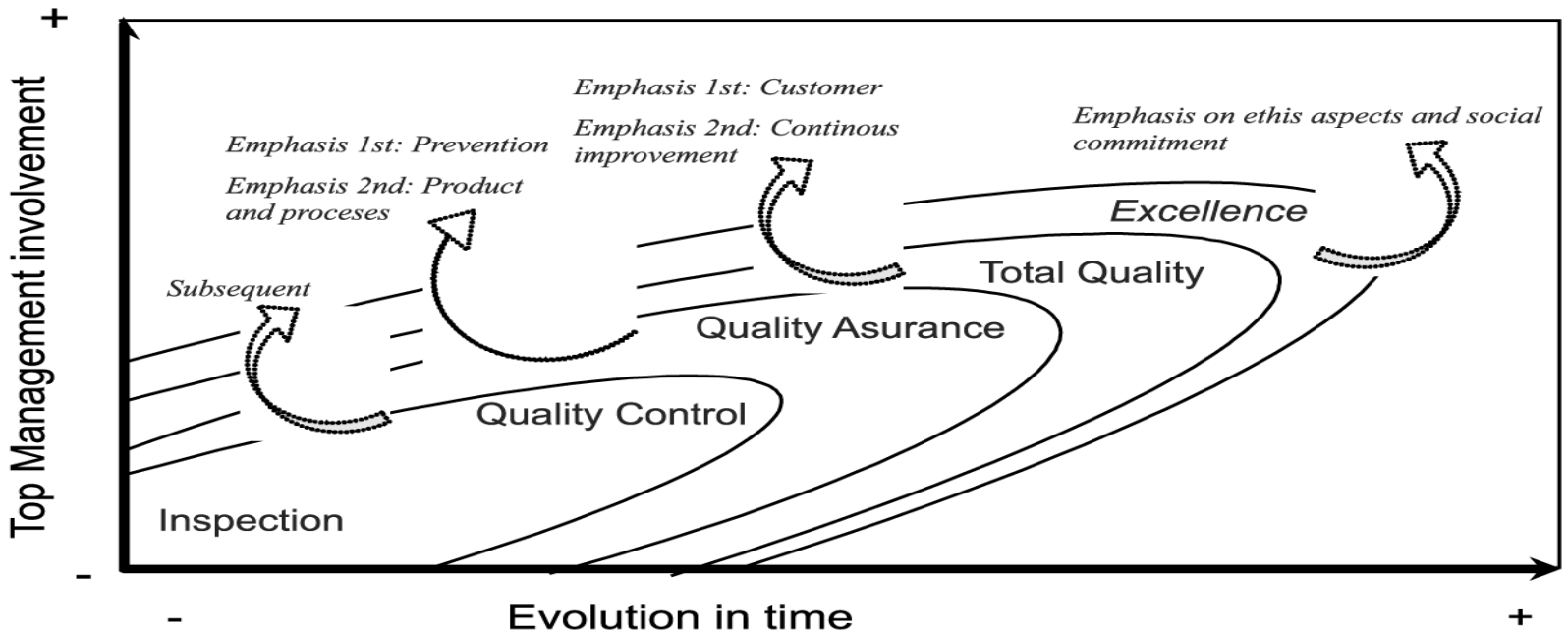
- The degree to which a set of inherent characteristics fulfills requirements.

ISO 9000

Degree to which a set of inherent characteristics (3.5.1) fulfils requirements (3.1.2)

- NOTE 1 The term “quality” can be used with adjectives such as poor, good or excellent.
- NOTE 2 “Inherent”, as opposed to “assigned”, means existing in something, especially as a permanent characteristic.
  - Requirement
    - need or expectation that is stated, generally implied or obligatory
    - NOTE 1 “Generally implied” means that it is custom or common practice for the organization, its customers and other interested parties , that the need or expectation under consideration is implied.
    - NOTE 2 A qualifier can be used to denote a specific type of requirement, e.g. product requirement, quality management requirement, customer requirement.
    - NOTE 3 A specified requirement is one that is stated, for example in a document (3.7.2).
    - NOTE 4 Requirements can be generated by different interested parties (3.3.7).
    - NOTE 5 This definition differs from that provided in 3.12.1 of ISO/IEC Directives, Part 2:2004.

Software Quality Degrees



Source: Martín-Castilla and Rodrigo (2003)

Quality Assurance vs “Quality Control”

	“Quality Control”	Quality Assurance
Scope	Products	Processes and Products
Objective	Detect problems in the work products	Verify adherence to processes
When	Once the work product is available	While the product is being developed

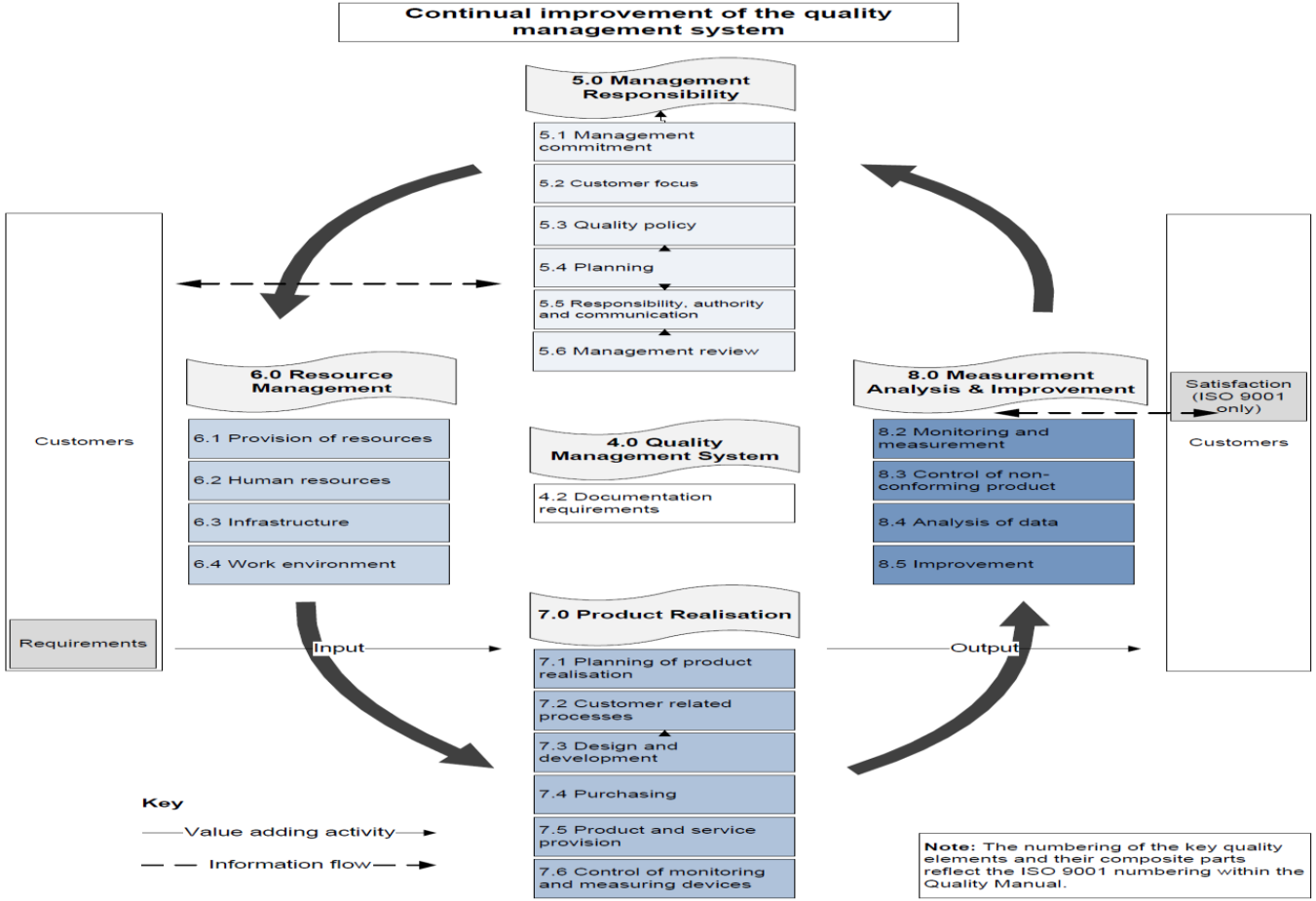
“Quality control” meaning of Final inspection

Software Standards

ISO (the International Organization for Standardization)

Three Standards of the ISO9000:2000 series are:	
ISO9000:2000	Quality Management Systems - Fundamentals and vacabulary
ISO9001:2000	Quality Management Systems - Requirements
ISO9004:2000	Quality Management Systems - Guidance for performance improvement

ISO -- Quality Management System Framework



QMS - ISO Elements – SW Highlight

ISO/IEC 90003:2004	Quality Management Guideline for - Computer Software and related services

- Section 4, Quality Management System
  - QMS for software products and related services **must developed**
  - The effectiveness of software-oriented quality management system process should be **monitored, measured, and improved**
  - QMS system should include documented process **procedure** which need **review and approved**

Section 5, Quality Management System

- **Management support** is required for QMS including Quality policies, objectives, and resources.
- **Quality policy** is mandatory to build up, and communicate with all level employee
- **Quality objectives** should be formulated for organization on all function levels
- QMS should be reviewed for **effectiveness** – Peer review, process audit ,etc…

Software Standards

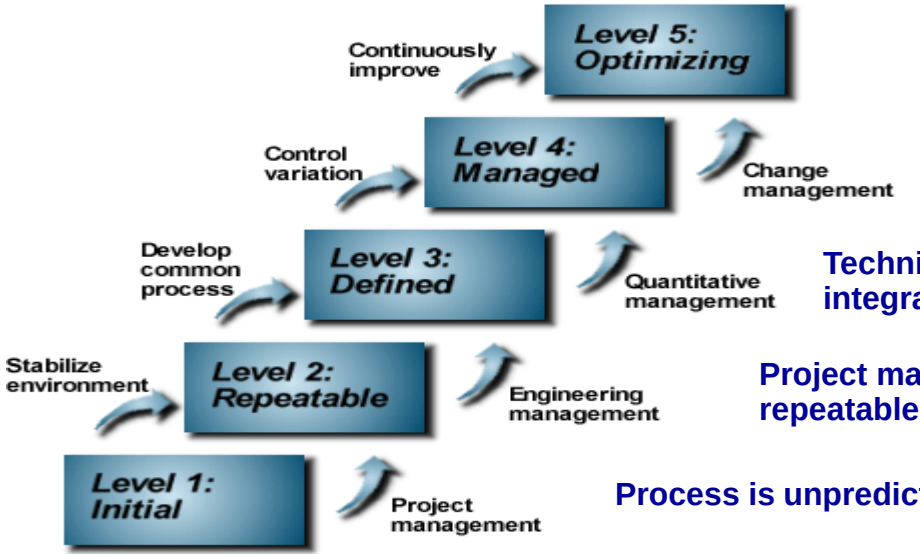
IEEE (Institute of Electrical and Electronics Engineers)

- IEEE Computer Society is the world's leading computing membership organization
- Technology leaders including: professors, researchers, software engineers, IT professionals, employers, and students
- IEEE Standard 730, Standard for Software Quality Assurance Plan first publish in 1979, and official as permanent standard in 1981
- Main IEEE standards interest to Software Quality

IEEE Standard 610.12	Software Engineering Terminology
IEEE Standard 730	Standard for Software Quality Assurance Plans
IEEE Standard 828	Software Configuration Management Plans
IEEE Standard 1028	Software Review and Audits
IEEE Standard 1058	Software Projects Management Plan
IEEE Standard 1058	Software Projects Metrics Methodology
IEEE Standard 1074	Software Projects Life Cycle Process

Quality System - CMMI

The CMM® describes an evolutionary improvement path from an ad-hoc, immature process to a mature, disciplined process.



Process are continuously examined and improved

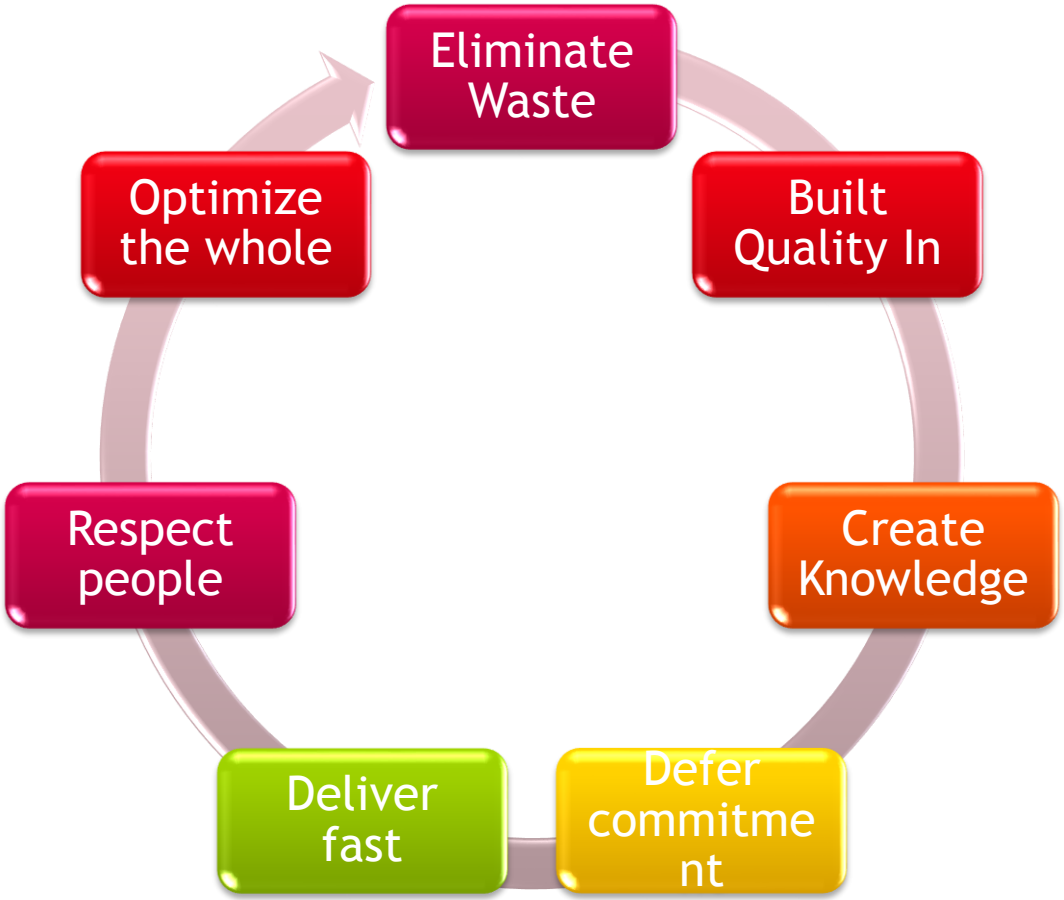
Product and process quality is quantitatively measured and controlled

Technical and management practices are integrated into a well-defined process

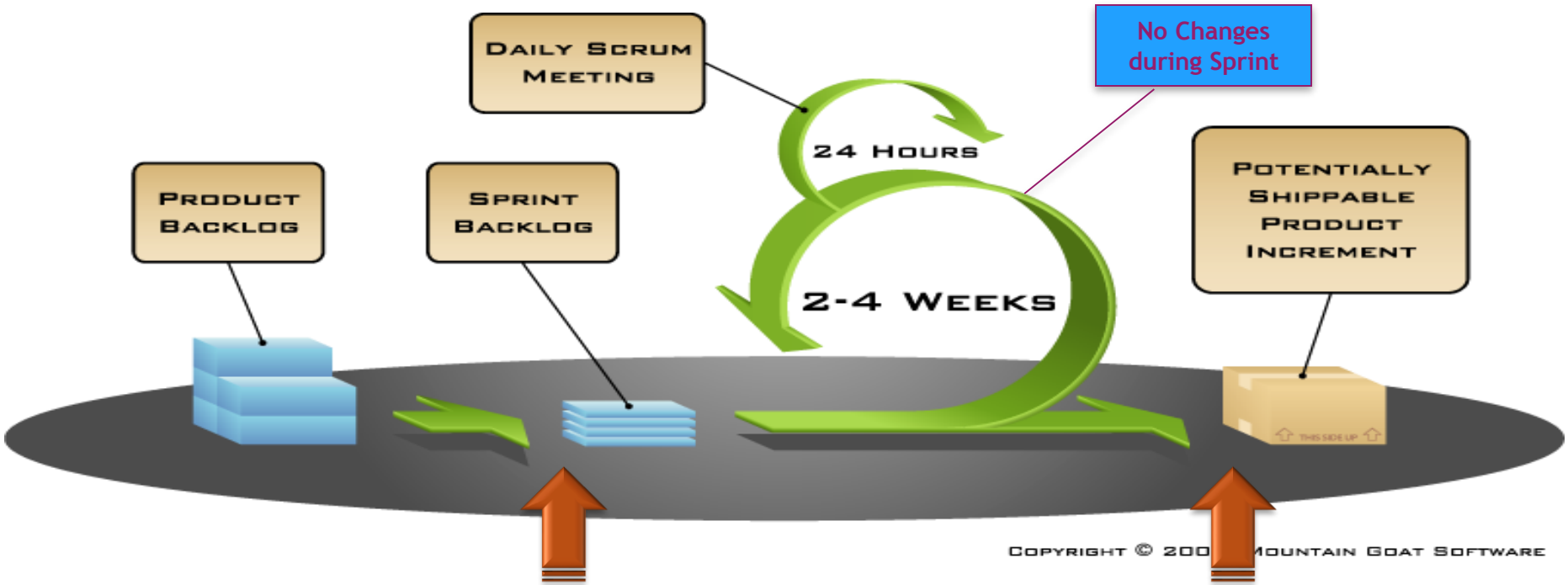
Project management practices are mastered and repeatable

Process is unpredictable and poorly controlled

Quality System - Lean SW Development 7 principles



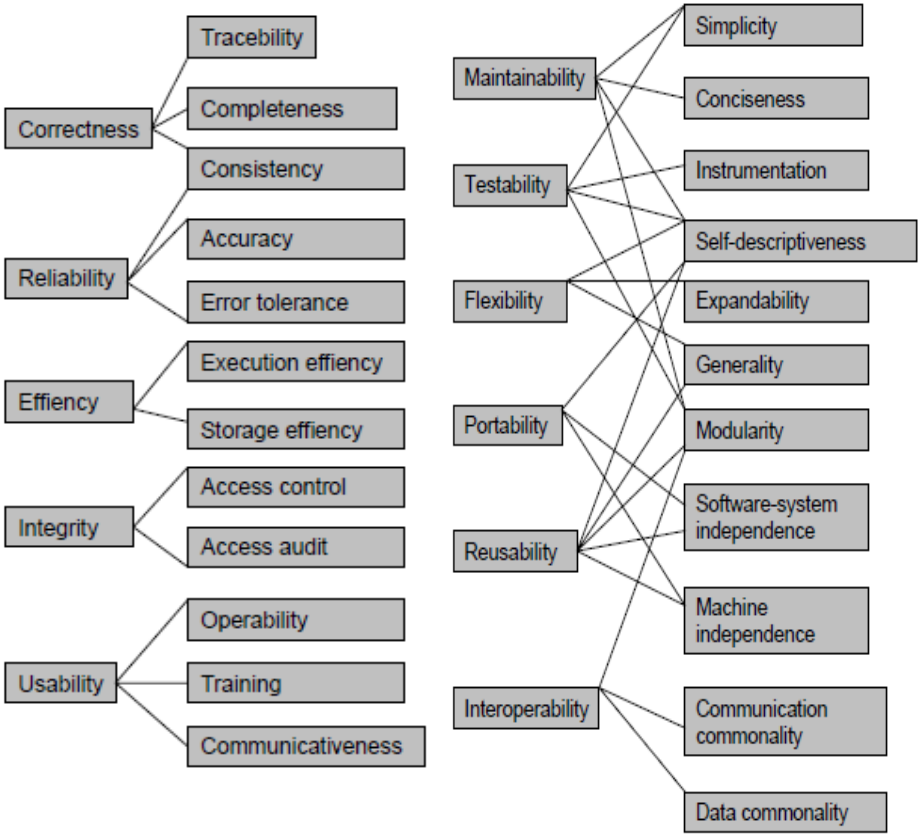
Quality System - Agile



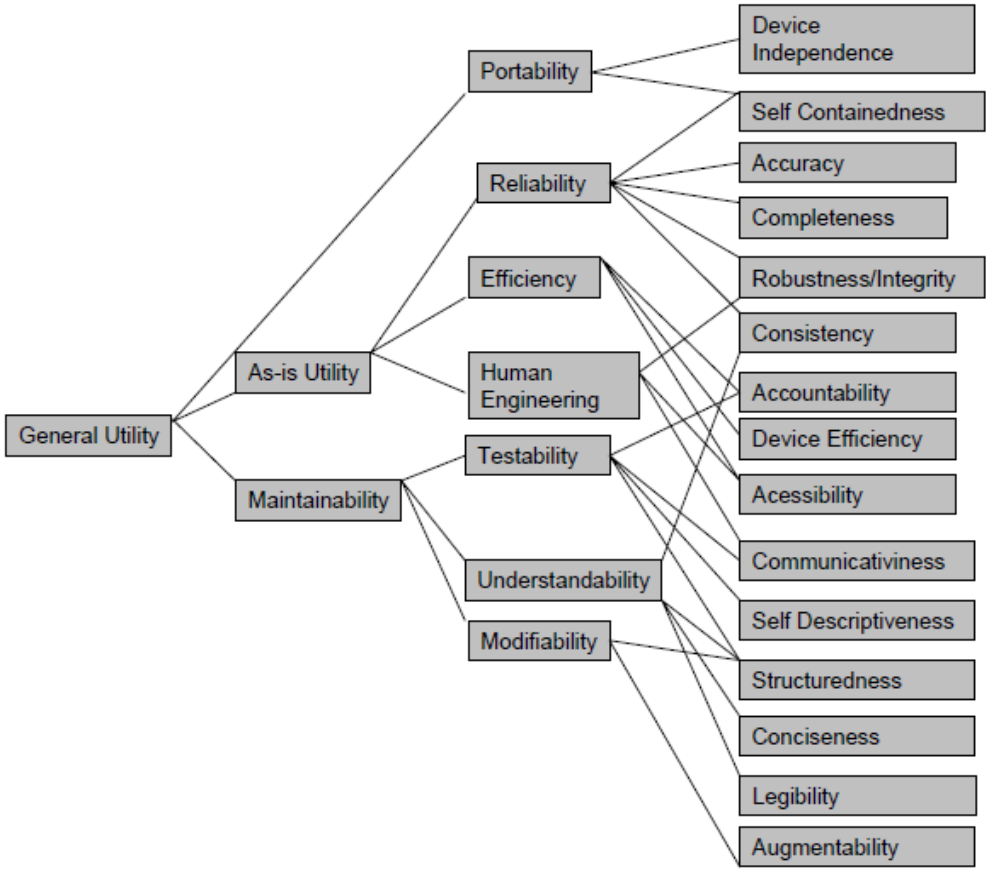
Sprint Planning meeting

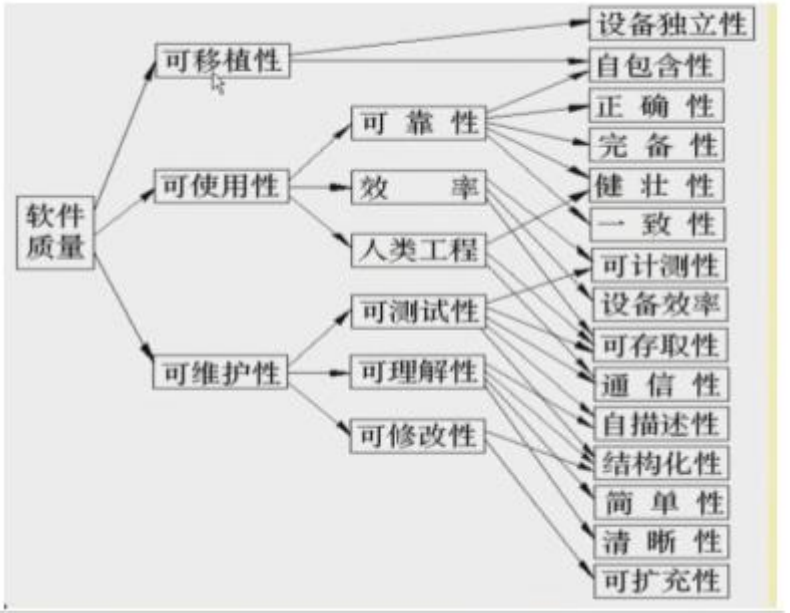
Sprint Review
Sprint Retrospective

McCall Quality Model

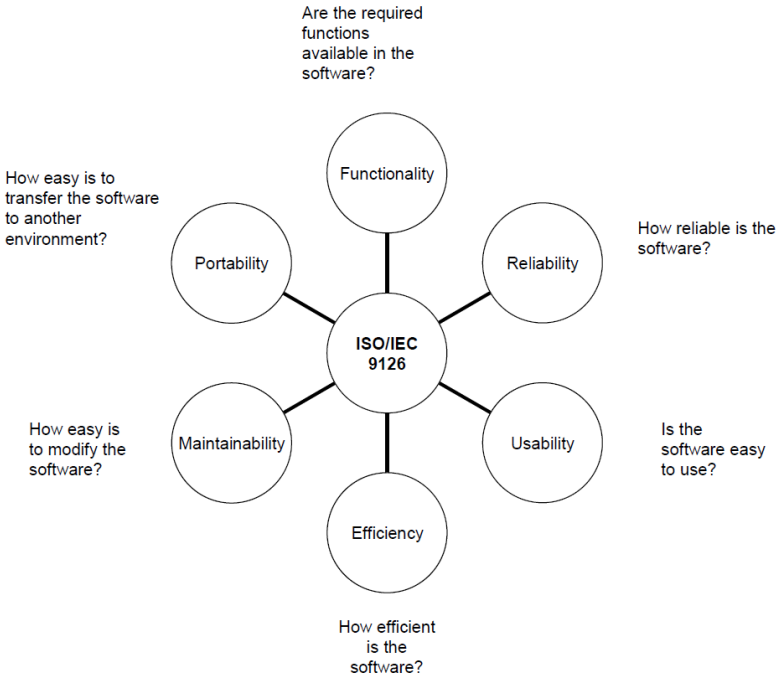


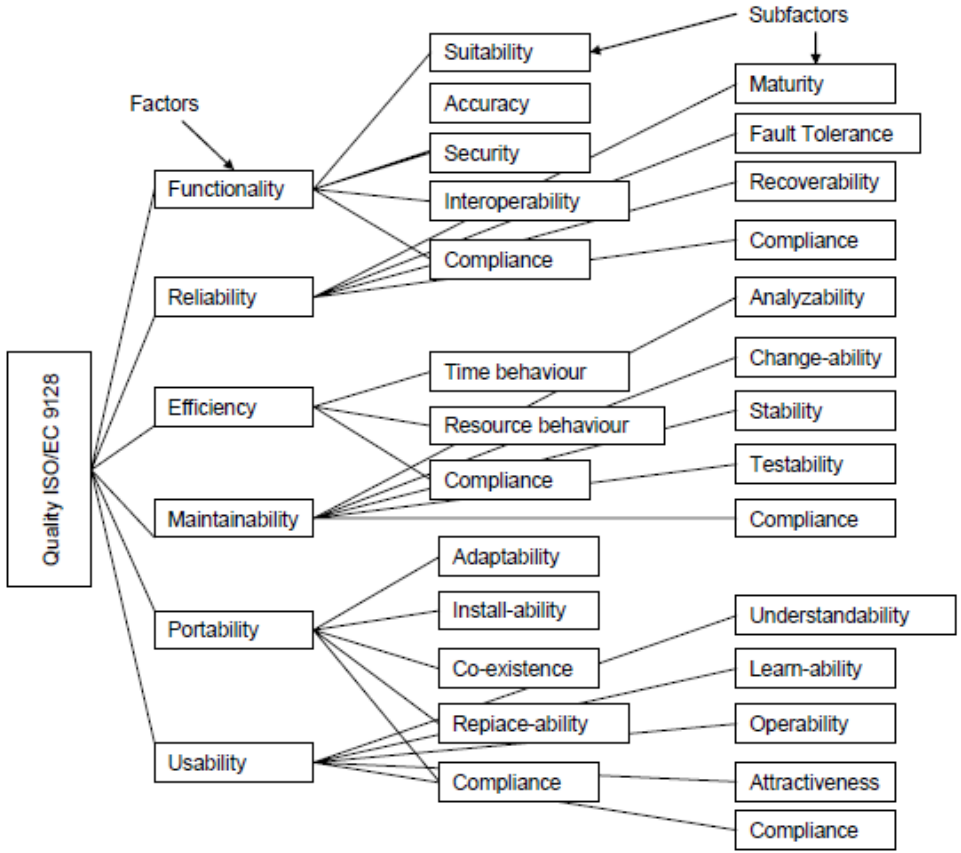
Boehm Quality Model





ISO 9126 Quality Model

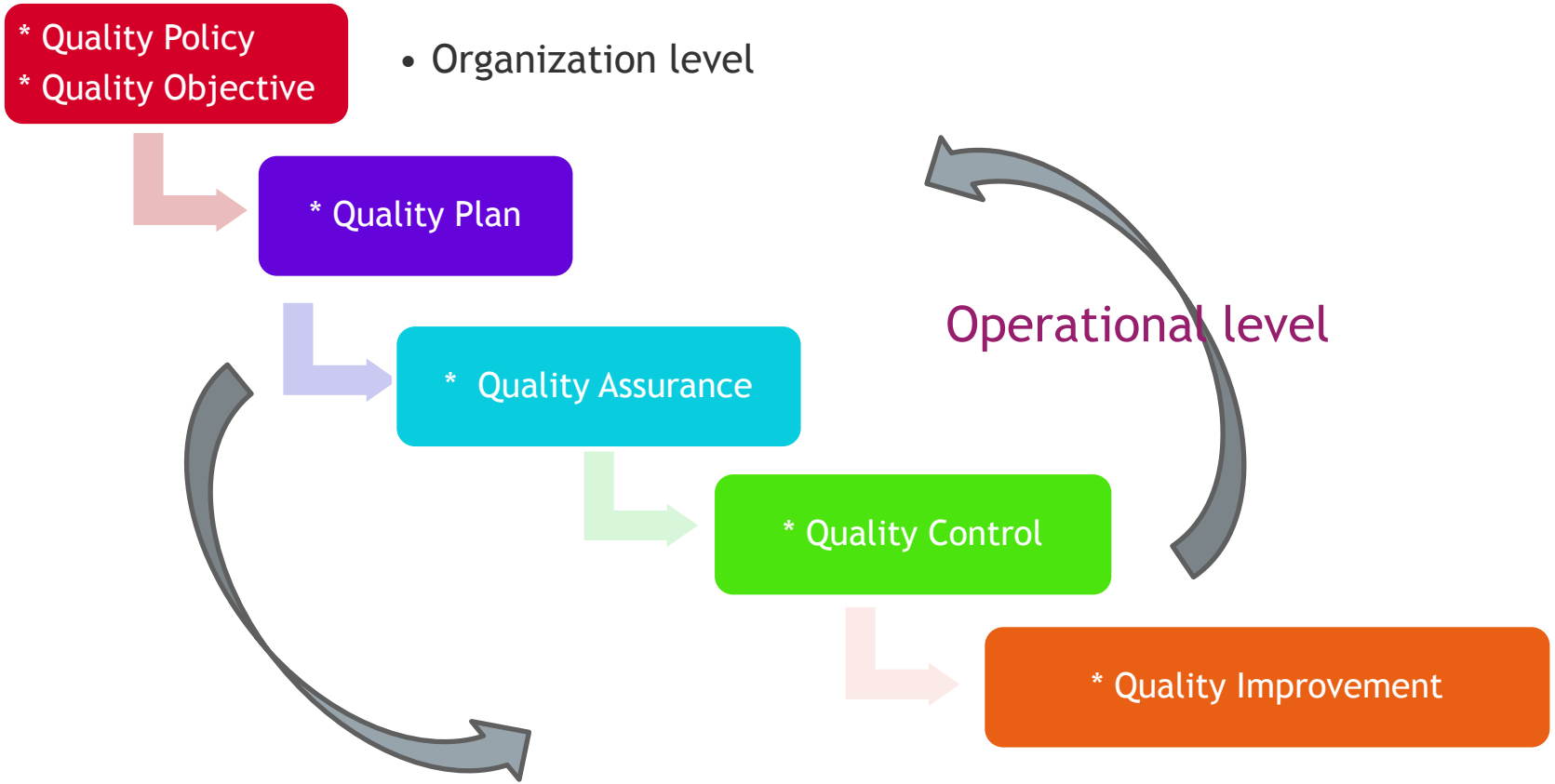




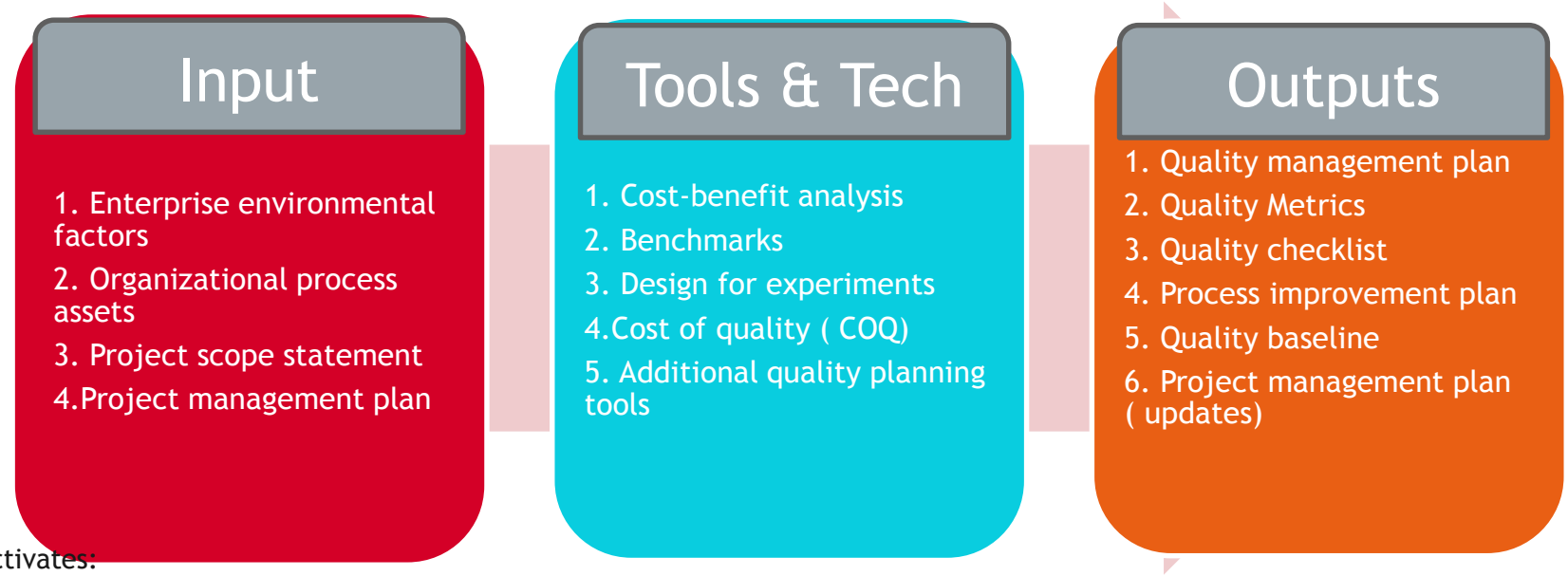
Quality Model Comparison

Criteria/goals	*McCall, 1977*	*Boehm, 1978*	*ISO 9126, 1993*
Correctness	*	*	maintainability
Reliability	*	*	*
Integrity	*	*	
Usability	*	*	*
Effiency	*	*	*
Maintainability	*	*	*
Testability	*		maintainability
Interoperability	*		
Flexibility	*	*	
Reusability	*	*	
Portability	*	*	*
Clarity		*	
Modifiability		*	maintainability
Documentation		*	
Resilience		*	
Understandability		*	
Validity		*	maintainability
Functionality			*
Generality		*	
Economy		*	

Quality Management Framework



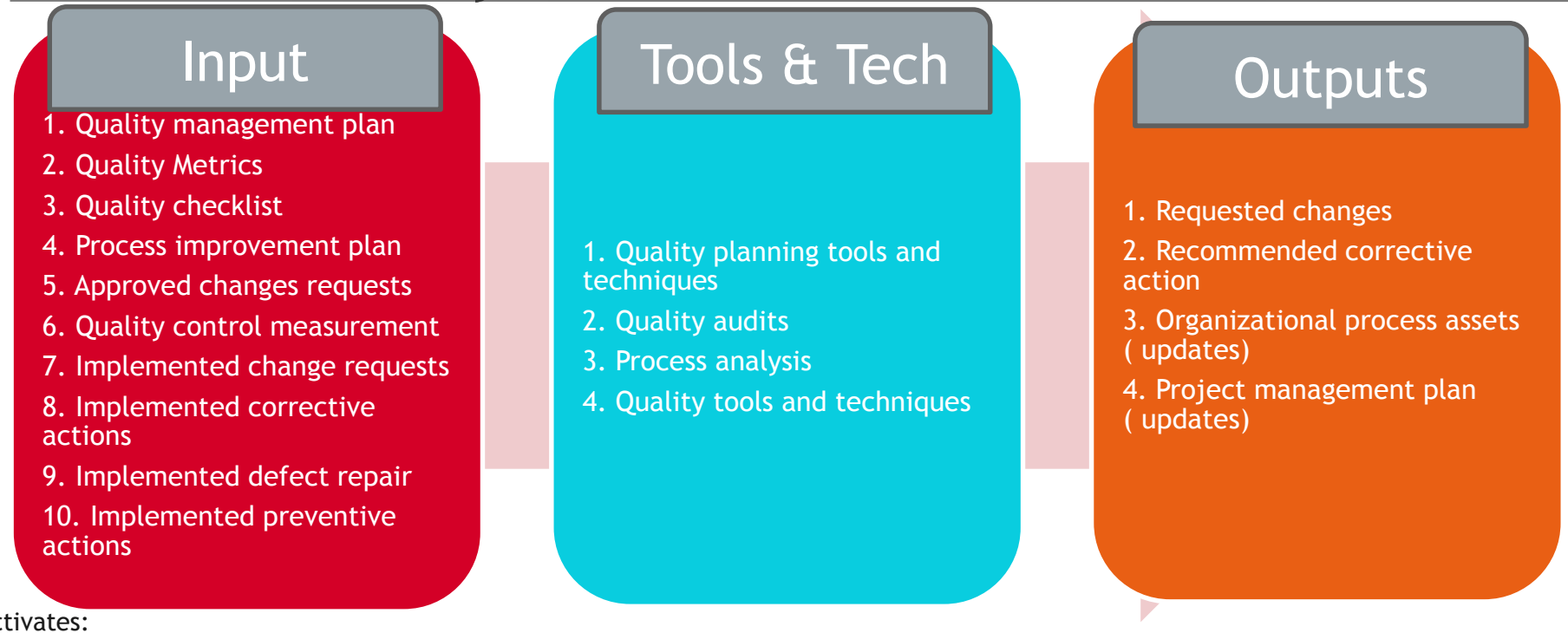
Quality Plan



Activates:

- A project level quality plan written by each project for declaring project commitment to follow an applicable set of standards, regulations, procedures and tools during the development lifecycle. In addition, SQP should contain quality goals to be achieved, expected risks and risk management. SQP sources are derived from
- SQA components that are adopted as is or customized to the project's needs
- New procedures, standards and tools complementing missing or not-applicable SQA components that have been written in particular for the project, or imported from outside the organization.
- Any deviation of an SQP from SQA should be justified by the project manager and be confirmed by the company management.

Perform Quality Assurance



Activates:

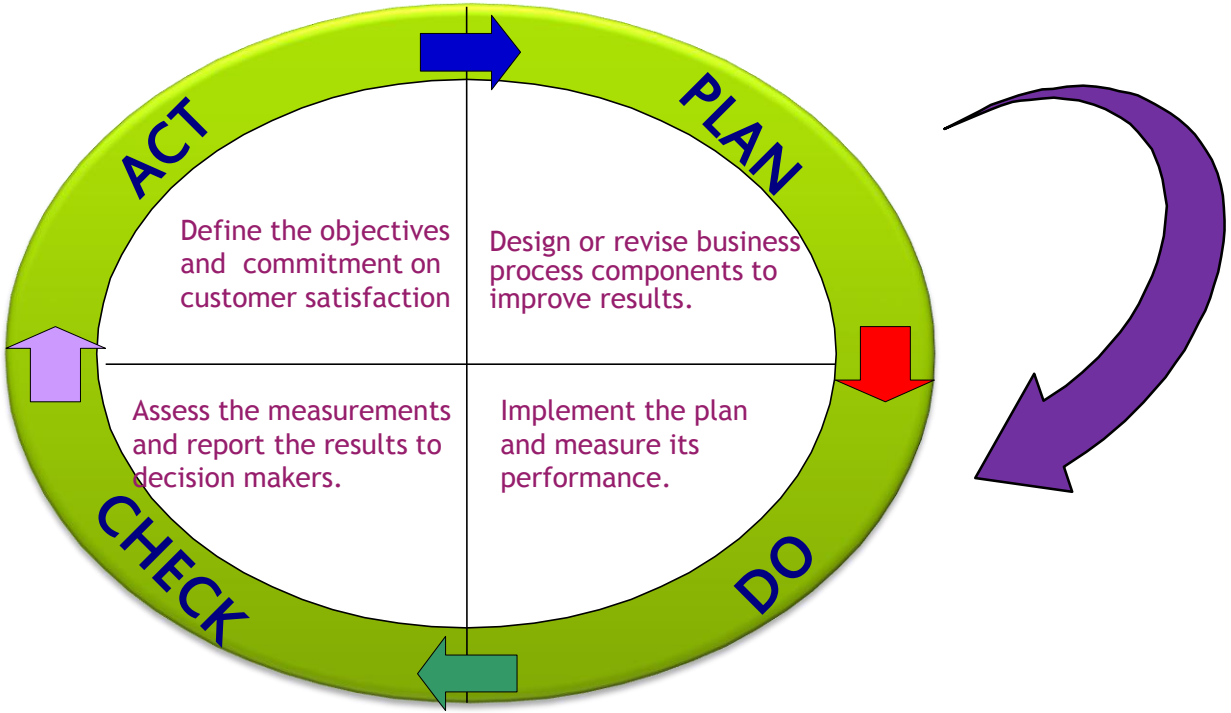
- An Organizational quality guide of Standards, regulations, and procedures to produce, verify, evaluate and confirm work products during the software development lifecycle
- Incorporated knowledge base of best practices
- Off-the-shelf software tools selected to apply the above

Perform Quality Control

Activates:

- Mentoring how to produce artifacts, such as well-defined engineering documents using standard templates
- Mentoring how to conduct standard processes, such as quality reviews / Inspection
- Perform in-process quality reviews to verify, evaluate and confirm artifacts
- Verify and evaluate to improve the use of methods, procedures and adopted software tools

Quality Continuous Improvement wheel



The concept of the PDCA Cycle was originally developed by Walter Shewhart at Bell Laboratories in the 1930s. It was then popularized by W. Edwards Deming in the 1950s onward and is frequently referred to as “the Deming Wheel.”

Chapter No. 2 -- Quality Control

Teacher - introduction

Liu Jia(Julia)

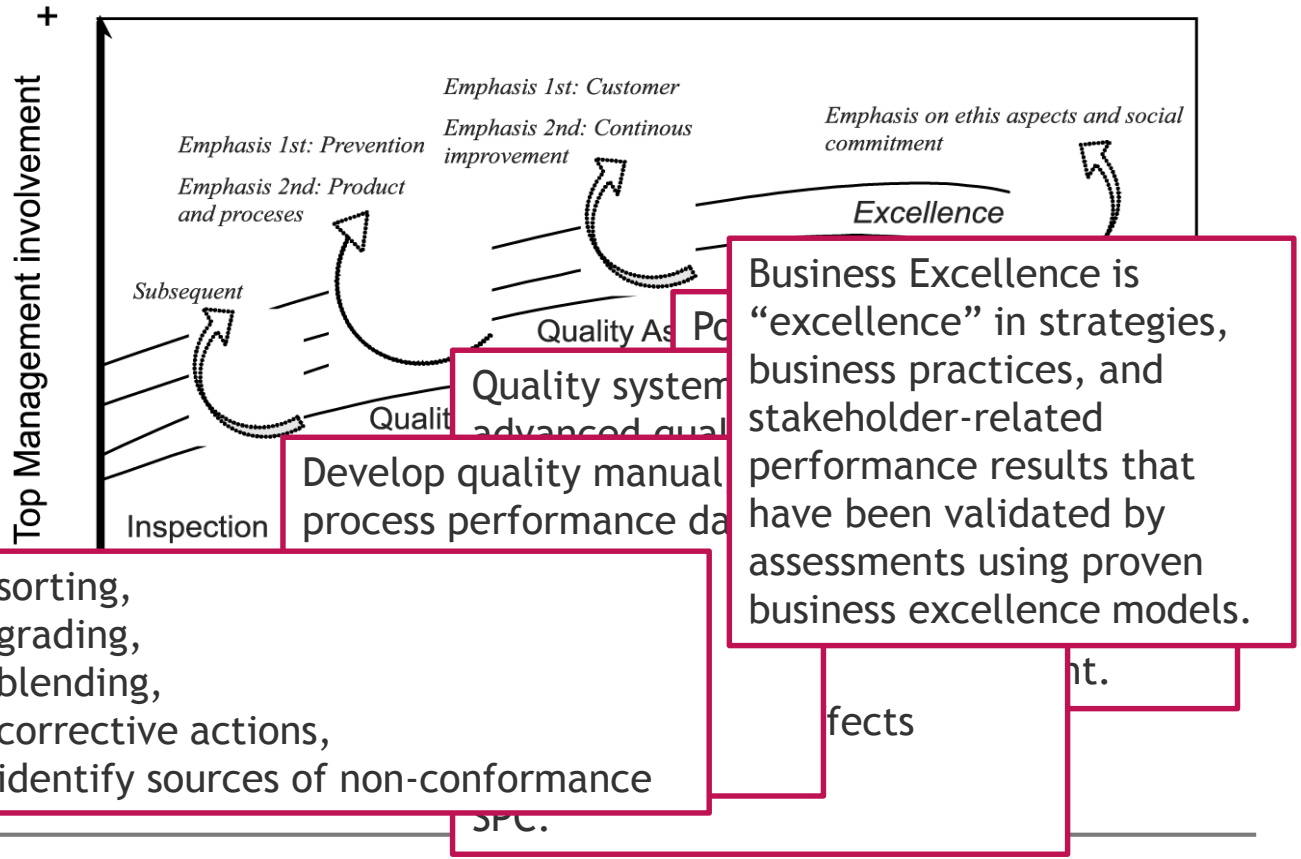
- Current Position
  - Quality Management System Manager of APAC
- Technicolor & Thomson Experience
  - Joined Technicolor in 2006
  - Quality Management System
  - Product Quality Assurance
  - Quality of Project Office
- Education & Certifications
  - Master of Software Computer Science – Huazhong University of Science & Technology
  - Certified Quality Engineer (CQE) - ASQ
  - Certified Project Management Professional (PMP) - PMI

Class Objectives

No.	Unit	Knowledge Point	Requirement	Suggest Hours	Content
2	Quality Control	Quality Control Concept & Tools	Know	2	- Concept of Quality Control - Quality Control Tools & Techniques
		Concept and classification of peer review	Know	4	- Concept of Peer Review - Classification of Peer Review
		Inspection Process & Measurement	Master	4	- Inspection Process - Inspection Technical
				2	- Inspection Measurement - Improve Effectiveness and efficiency of Inspection

Quality Control

Quality Concept : Evolution of Quality Management



Quality Control Concept

- **Quality control**, or **QC** for short, is a process by which entities review the quality of all factors involved in production. ISO 9000 defines quality control as "A part of quality management focused on fulfilling quality requirements".
- Focus: Detection
- Quality Assurance vs Quality Control

	Quality Control	Quality Assurance
Scope	Products	Processes and Products
Objective	Detect problems in the work products	Verify adherence to processes
When	Once the product is made available	While (or even before) the product is being developed

Notable Approches to Quality Control

Terminology	Approximate year of first use	Description
Statistical quality control (SQC)	1930s	The application of statistical methods (specifically control charts and acceptance sampling) to quality control.[5]:556
Total quality control (TQC)	1956	Popularized by Armand V. Feigenbaum in a Harvard Business Review article[6] and book of the same name.[7] Stresses involvement of departments in addition to production (e.g., accounting, design, finance, human resources, marketing, purchasing, sales).
Statistical process control (SPC)	1960s	The use of control charts to monitor an individual industrial process and feed back performance to the operators responsible for that process. Inspired by control systems.
Company-wide quality control (CWQC)	1968	Japanese-style total quality control[7]
Total Quality Management (TQM)	1985	Quality movement originating in the United States Department of Defense that uses (in part) the techniques of statistical quality control to drive continuous organizational improvement.[8]
Six Sigma (6σ)	1986	Statistical quality control applied to business strategy.[9] Originated by Motorola.

Quality Control Tools

➢ 7 Basic Quality Control Tools

1. Check Sheet
2. Scatter Diagrams
3. Cause and Effect Analysis
4. Histogram: The most commonly used graph for showing frequency distributions, or how often each different value in a set of data occurs
5. Pareto Charts: Shows on a bar graph which factors are more significant.
6. Control Charts **:** Graphs used to study how a process changes over time
7. Stratification: A technique that separates data gathered from a variety of sources so that patterns can be seen (some lists replace “stratification” with “flowchart” or “run chart”).

➢ Some Other QC Tools

1. Flow Chart
2. Run Chart
3. Review & Inspection

Check sheet

➢ Concept

- Also called: defect concentration diagram.
- A check sheet is a structured, prepared form for collecting and analyzing data. This is a generic tool that can be adapted for a wide variety of purposes.

➢ Check Sheet Example

- The figure below shows a check sheet used to collect data on telephone interruptions. The tick marks were added as data was collected over several weeks.

Telephone Interruptions

Reason	Day					
	Mon	Tues	Wed	Thurs	Fri	Total
Wrong number	𝍸	||	|	𝍸	𝍸 ||	20
Info request	||	||	||	||	||	10
Boss	𝍸	||	𝍸 ||	|	||||	19
Total	12	6	10	8	13	49

Check sheet

Project Name: _____

Name of Data Recorder: _____

Location: _____

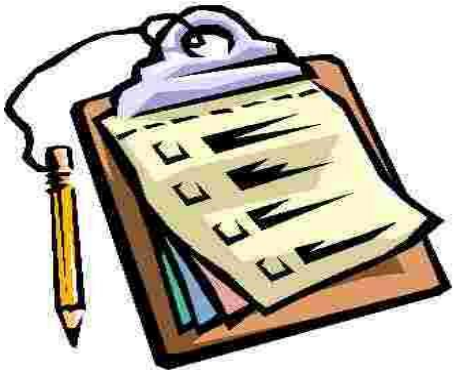
Data Collection Dates: _____

Defect Types/ Event Occurrence	Dates							TOTAL
	Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	
Defect 1								
Defect 2								
Defect 3								
Defect 4								
Defect 5								
Defect 6								
Defect 7								
Defect 8								
Defect 9								
Defect 10								
TOTAL								

Check Sheet

➢ When to Use a Check Sheet

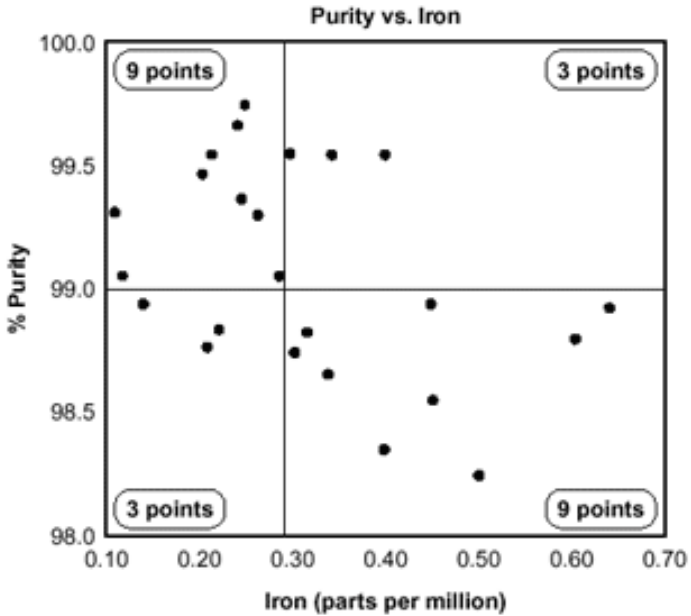
- When data can be observed and collected repeatedly by the same person or at the same location.
- When collecting data on the frequency or patterns of events, problems, defects, defect location, defect causes, etc.
- When collecting data from a production process.



Scatter Diagrams

➢ Concept

- Also called: scatter plot, X–Y graph
- The scatter diagram graphs pairs of numerical data, with one variable on each axis, to look for a relationship between them. If the variables are correlated, the points will fall along a line or curve. The better the correlation, the tighter the points will hug the line.



Scatter Diagrams

Quality Tools

Scatter Diagram

Description

This template illustrates a Scatter Diagram, also called a Scatter Plot or XY Graph. Scatter Diagrams show the relationship between an input, X and the output, Y.

Learn About Scatter Diagrams

Instructions

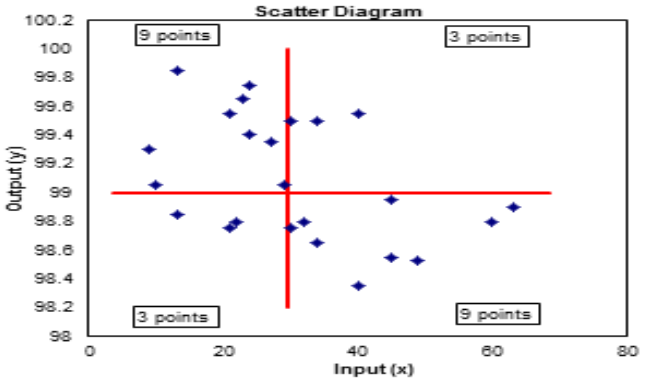
Enter up to 90 data points in the cells provided. The Scatter Diagram is displayed. The number of points in each quadrant is calculated. A test for correlation is performed using a trend table. Further information on this test can be found by

Learn More

To learn more about other quality tools, visit the ASQ Learn About Quality web site.

Learn About Quality

Results	
# in 1st Q	3
# in 2nd Q	9
# in 3rd Q	3
# in 4th Q	9
A = 9 + 9	18
B = 3 + 3	6
Q = min	6
N = # pts	24
Trend test limit	6
Q < Limit?	No
Correlated?	No



	Input (x)	Output (y)
1	63	98.9
2	60	98.8
3	49	98.53
4	40	98.35
5	45	98.55
6	45	98.95
7	30	98.75
8	34	98.65
9	32	98.8
10	21	98.75
11	22	98.8
12	29	99.05
13	27	99.35
14	13	99.85
15	10	99.05
16	9	99.3
17	21	99.55
18	24	99.4

	Input (x)	Output (y)
19	40	99.55
20	34	99.5
21	30	99.5
22	24	99.75
23	23	99.65
24	13	98.85
25		
26		
27		
28		
29		
30		
31		
32		
33		
34		
35		
36		

	Input (x)	Output (y)
37		
38		
39		
40		
41		
42		
43		
44		
45		
46		
47		
48		
49		
50		
51		
52		
53		
54		

	Input (x)	Output (y)
55		
56		
57		
58		
59		
60		
61		
62		
63		
64		
65		
66		
67		
68		
69		
70		
71		
72		

	Input (x)	Output (y)
73		
74		
75		
76		
77		
78		
79		
80		
81		
82		
83		
84		
85		
86		
87		
88		
89		
90		

Scatter Diagrams

➢ When to Use a Scatter Diagram

- When you have paired numerical data.
- When your dependent variable may have multiple values for each value of your independent variable.
- When trying to determine whether the two variables are related, such as…
  - When trying to identify potential root causes of problems.
  - After brainstorming causes and effects using a fishbone diagram, to determine objectively whether a particular cause and effect are related.
  - When determining whether two effects that appear to be related both occur with the same cause.
  - When testing for autocorrelation before constructing a control chart.

Fish Bone Diagram

➢ Concept

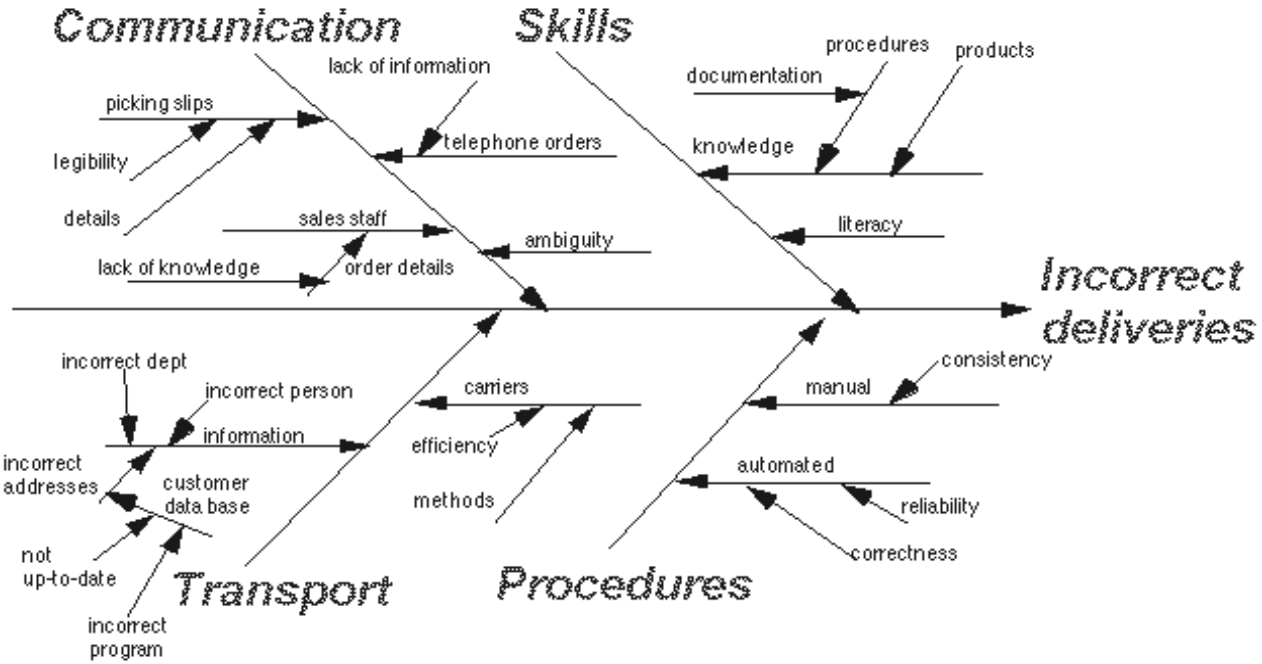
- Also Called: Cause–and–Effect Diagram, Ishikawa Diagram
- Variations: cause enumeration diagram, process fishbone, time–delay fishbone, CEDAC (cause–and–effect diagram with the addition of cards), desired–result fishbone, reverse fishbone diagram
- The fishbone diagram identifies many possible causes for an effect or problem. It can be used to structure a brainstorming session. It immediately sorts ideas into useful categories

➢ When to Use a Fishbone Diagram

- When identifying possible causes for a problem.
- Especially when a team’s thinking tends to fall into ruts

Fish Bone Diagram

Example: When a production team is about to launch a new product, the factors that will affect the final product must be recognized. The fishbone diagram can depict problems before they have a chance to begin.



Fish Bone Diagram

Example 2:

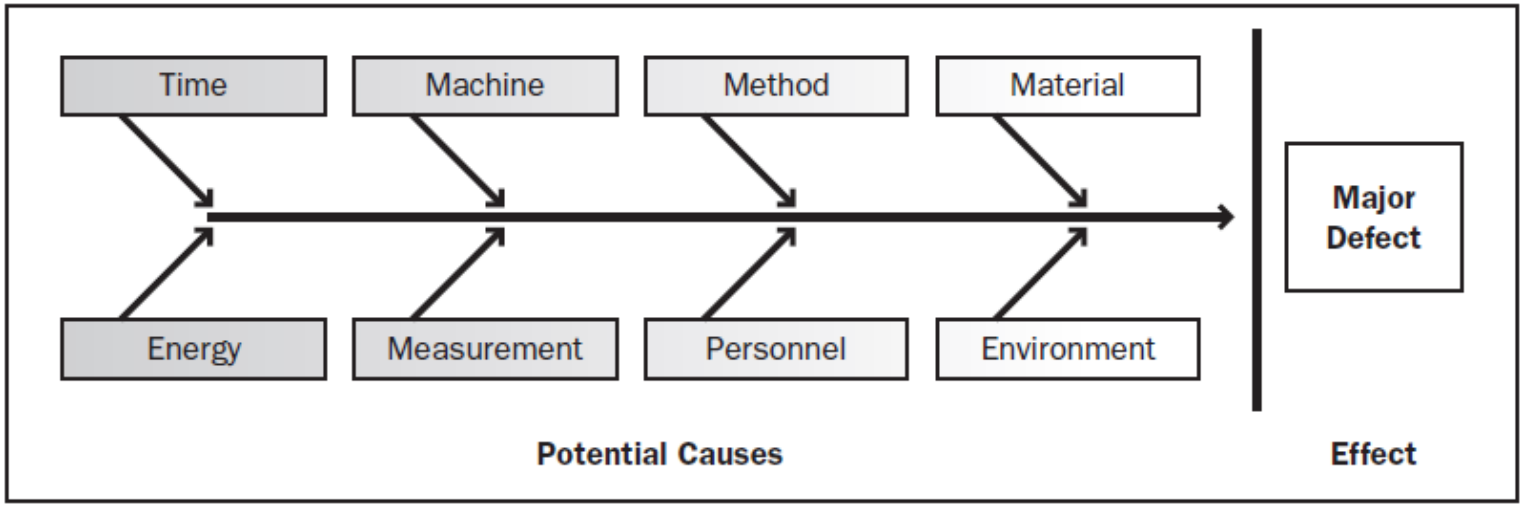


Figure 8-12. Classic Sources of Problems to Consider

Fish Bone Diagram

Quality Tools

Cause and Effect Diagram

Description

This template illustrates a Cause and Effect Diagram, also called a Fishbone or Ishikawa Diagram. A detailed discussion of Cause and Effect Diagrams can be found at www.ASQ.org

Learn About C and E Diagrams

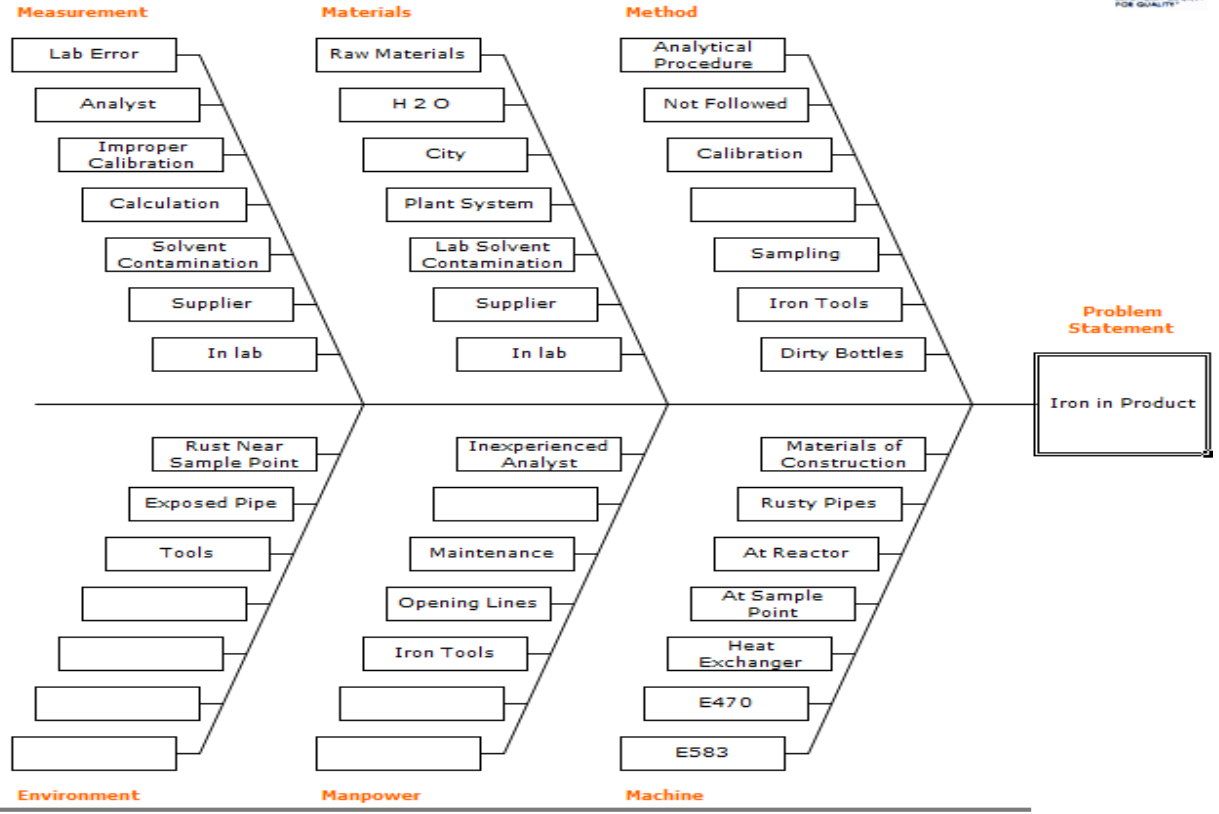
Instructions

- Enter the Problem Statement in box provided.
- Brainstorm the major categories of the problem. Generic headings are provided.
- Write the categories of causes as branches from the main arrow.

Learn More

To learn more about other quality tools, visit the ASQ Learn About Quality web site.

Learn About Quality



Cause and Effect Diagram - 5M & 4P

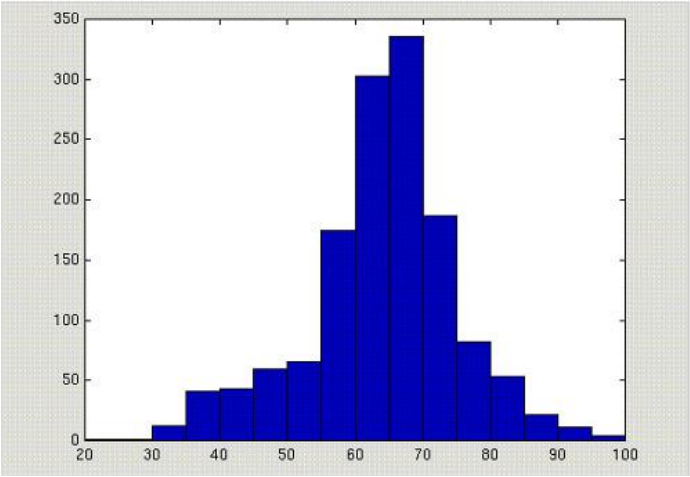
5M: The main “bone” structure or branches typically comprise machinery, manpower, method, material and maintenance.

4P: For non manufacturing areas, the 4Ps [policies, procedures, people and plant] are sometimes found to be more appropriate.

Histograms

➢ Histogram Definition

- A frequency distribution shows how often each different value in a set of data occurs. A histogram is the most commonly used graph to show frequency distributions. It looks very much like a bar chart, but there are important differences between them.



Histograms

Quality Tools

Histogram

Description

This template illustrates a Frequency Distribution, commonly refered to as a Histogram.

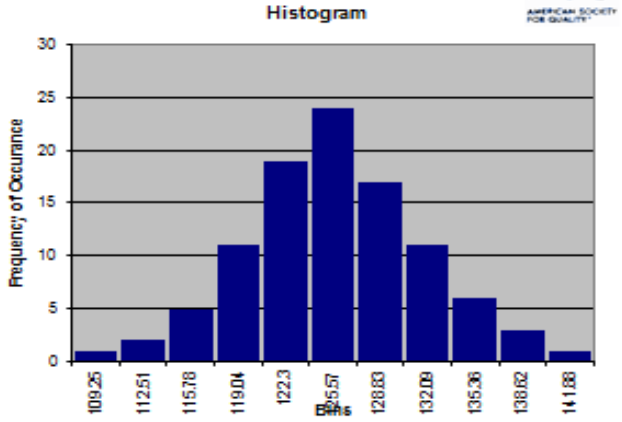
Learn About Histograms

Instructions

Enter up to 200 data points in the cells provided

Data Table

Bin	Frequency
109	1
113	2
116	5
119	11
122	19
126	24
129	17
132	11
135	6
139	3
142	1



1	125.36	26	119.42	51	123.72	76	131.48	101		126		151		176	
2	126.66	27	120.83	52	124.56	77	127.92	102		127		152		177	
3	130.28	28	136.53	53	116.03	78	114.08	103		128		153		178	
4	133.74	29	120.4	54	121.06	79	125.45	104		129		154		179	
5	126.92	30	136.58	55	124.5	80	124.65	105		130		155		180	
6	120.85	31	121.73	56	131.93	81	127.93	106		131		156		181	
7	119.42	32	132.72	57	133.25	82	123.88	107		132		157		182	
8	128.61	33	109.25	58	122.08	83	114.25	108		133		158		183	
9	123.53	34	125.42	59	117.88	84	123.68	109		134		159		184	
10	130.15	35	117.67	60	126.1	85	117.06	110		135		160		185	
11	126.02	36	124.01	61	125.26	86	130.69	111		136		161		186	
12	116.65	37	118.74	62	125.03	87	119.8	112		137		162		187	
13	125.24	38	128.99	63	122.74	88	119.72	113		138		163		188	
14	126.84	39	131.11	64	117.57	89	110.99	114		139		164		189	
15	125.95	40	112.27	65	120.63	90	123.12	115		140		165		190	
16	114.41	41	118.76	66	120.83	91	125.37	116		141		166		191	
17	138.62	42	119.15	67	132.6	92	127.01	117		142		167		192	
18	127.4	43	122.42	68	131.25	93	121.31	118		143		168		193	
19	127.59	44	122.22	69	130.83	94	119.26	119		144		169		194	
20	123.57	45	134.71	70	116.68	95	124.52	120		145		170		195	
21	133.76	46	126.22	71	114.99	96	128.85	121		146		171		196	
22	124.6	47	130.33	72	126.69	97	123.87	122		147		172		197	
23	113.48	48	120.52	73	121.57	98	122.87	123		148		173		198	
24	128.6	49	126.88	74	124.43	99	126.61	124		149		174		199	
25	121.04	50	117.4	75	137.04	100	117.96	125		150		175		200	

Histograms

➢ When to Use a Histogram

- When the data are numerical.
- When you want to see the shape of the data's distribution, especially when determining whether the output of a process is distributed approximately normally.
- When analyzing whether a process can meet the customer's requirements.
- When analyzing what the output from a supplier's process looks like.
- When seeing whether a process change has occurred from one time period to another.
- When determining whether the outputs of two or more processes are different.
- When you wish to communicate the distribution of data quickly and easily to others
- Histograms can be used to determine distribution of sales. Say for instance a company wanted to measure the revenues of other companies and wanted to compare numbers.

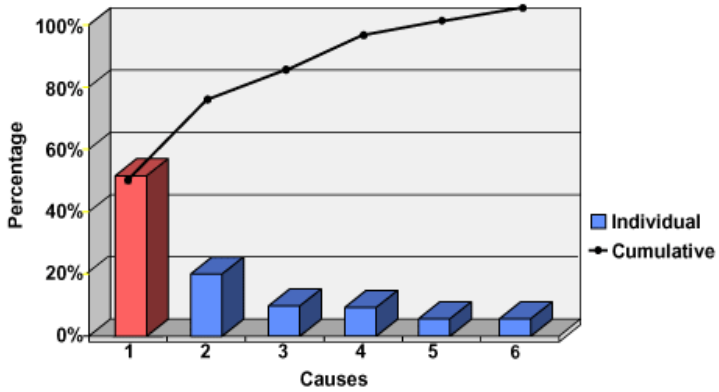
Pareto Chart

➢ Concept

- Also called: Pareto diagram, Pareto analysis
- Variations: weighted Pareto chart, comparative Pareto charts
- A Pareto chart is a bar graph. The lengths of the bars represent frequency or cost (time or money), and are arranged with longest bars on the left and the shortest to the right. In this way the chart visually depicts which situations are more significant.

➢ When to Use a Pareto Chart

- When analyzing data about the frequency of problems or causes in a process.
- When there are many problems or causes and you want to focus on the most significant.
- When analyzing broad causes by looking at their specific components.
- When communicating with others about your data.



Pareto Analysis

Quality Tools

Pareto Chart

Description

This template illustrates a Pareto Chart, also called a Pareto Diagram or Pareto Analysis. A detailed discussion of Pareto Charts can be found at www.ASQ.org

Learn About Pareto Charts

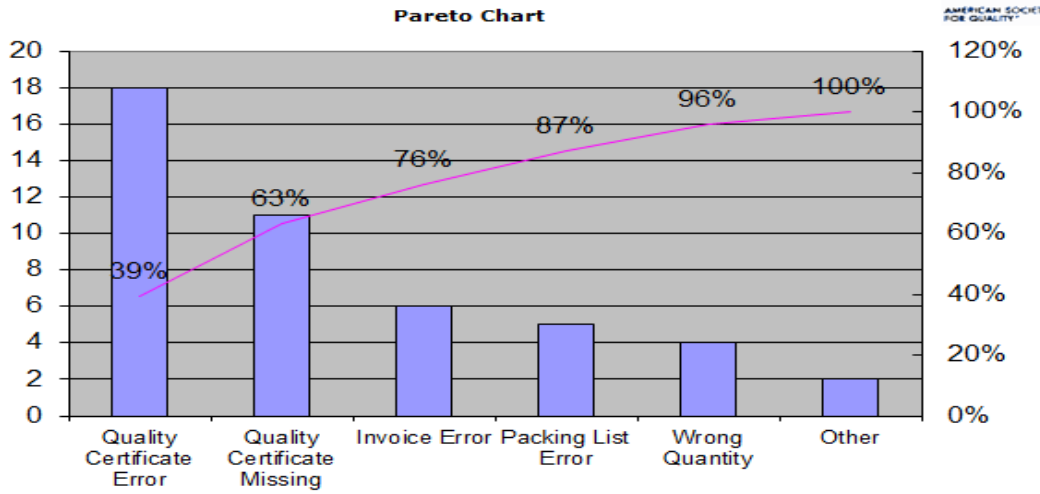
Instructions

- Enter up to 45 categories along with the total for each category in the cells provided.
- Items and their totals do not need to be entered in rank order. This will be done automatically.
- Do not enter duplicate categories.

Learn More

To learn more about other quality tools, visit the ASQ Learn About Quality web site.

Learn About Quality



Category	Total
Invoice Error	6
Wrong Quantity	4
Quality Certificate Missing	11
Packing List Error	5
Quality Certificate Error	18
Other	2

Category	Total

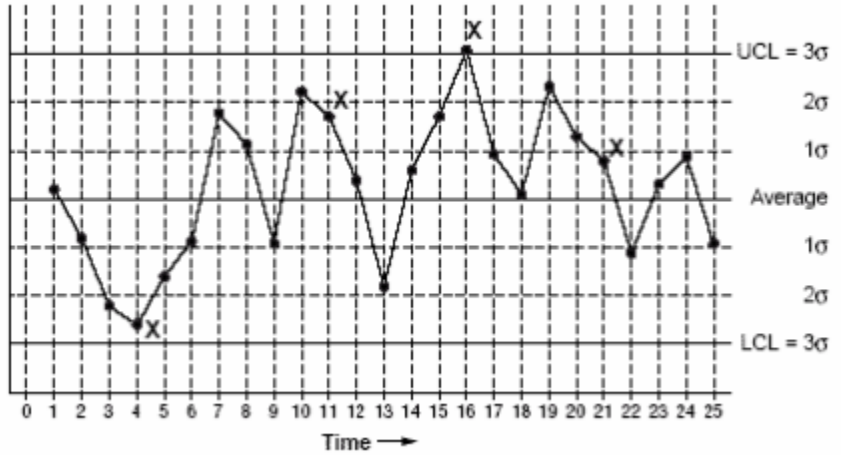
Category	Total

Control Charts

➢ Concept

- Also called: statistical process control
- The control chart is a graph used to study how a process changes over time. Data are plotted in time order. A control chart always has a central line for the average, an upper line for the upper control limit and a lower line for the lower control limit. These lines are determined from historical data. By comparing current data to these lines, you can draw conclusions about whether the process variation is consistent (in control) or is unpredictable (out of control, affected by special causes of variation).

- Control charts for variable data are used in pairs. The top chart monitors the average, or the centering of the distribution of data from the process. The bottom chart monitors the range, or the width of the distribution. If your data were shots in target practice, the average is where the shots are clustering, and the range is how tightly they are clustered. Control charts for attribute data are used singly



Control Charts

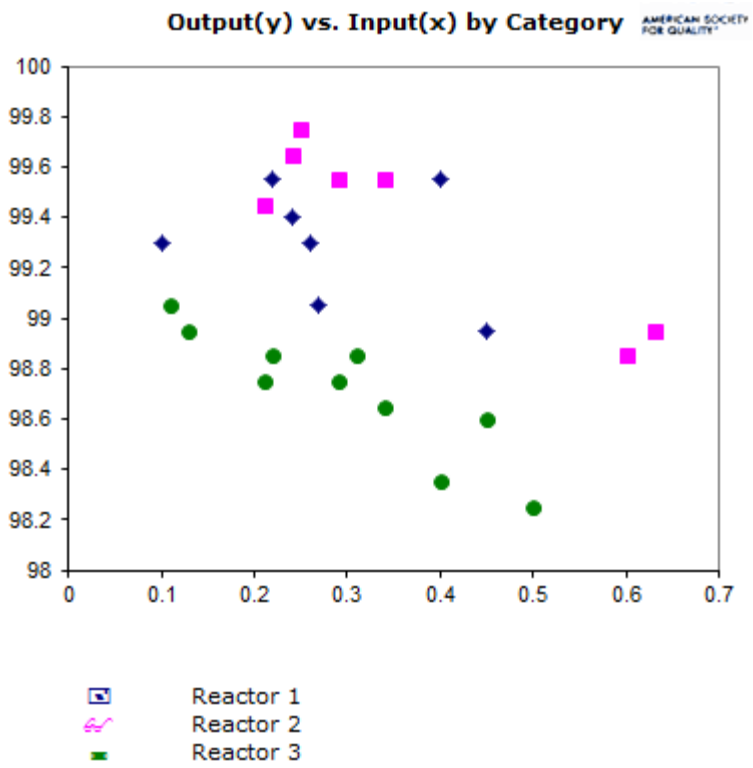
➢ When to Use a Control Chart

- When controlling ongoing processes by finding and correcting problems as they occur.
- When predicting the expected range of outcomes from a process.
- When determining whether a process is stable (in statistical control).
- When analyzing patterns of process variation from special causes (non-routine events) or common causes (built into the process).
- When determining whether your quality improvement project should aim to prevent specific problems or to make fundamental changes to the process.

Stratification Diagram

➢ Concept

- Stratification is a technique used in combination with other data analysis tools. When data from a variety of sources or categories have been lumped together, the meaning of the data can be impossible to see. This technique separates the data so that patterns can be seen.



Stratification Diagram

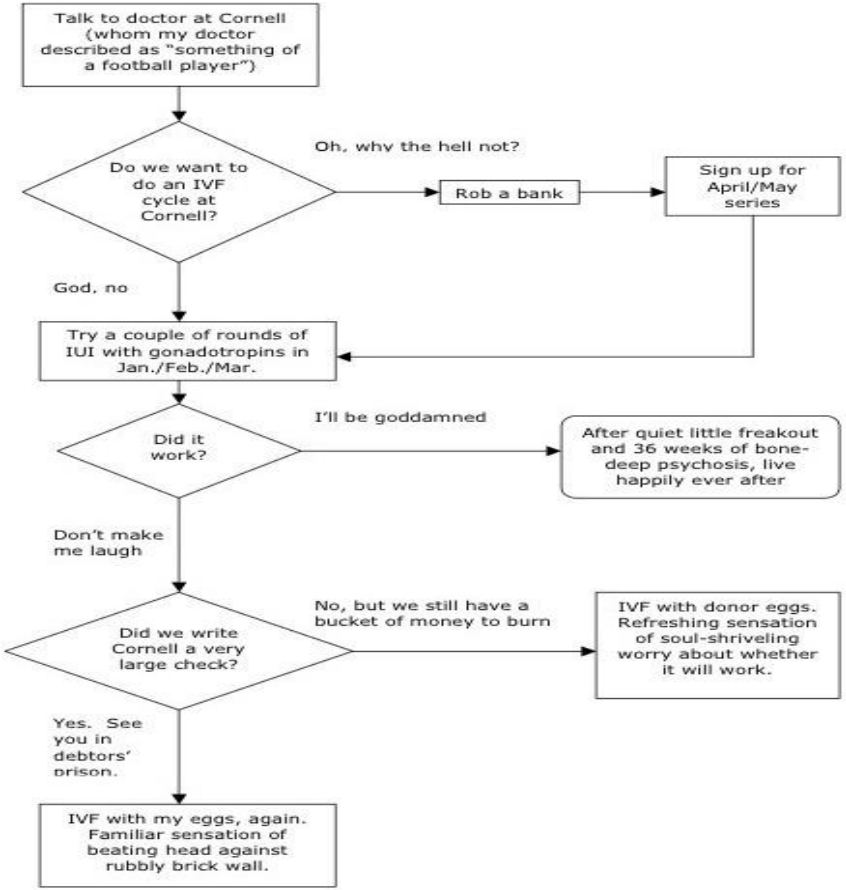
➢ When to Use Stratification

- Before collecting data.
- When data come from several sources or conditions, such as shifts, days of the week, suppliers or population groups.
- When data analysis may require separating different sources or conditions.

Flowcharts

➢ Concept

- Diagrammatic picture
- Show the complete process.
- Each “activity” is individual to focus attention on.
- Uncover irregularity and potential problem points through data analysis.



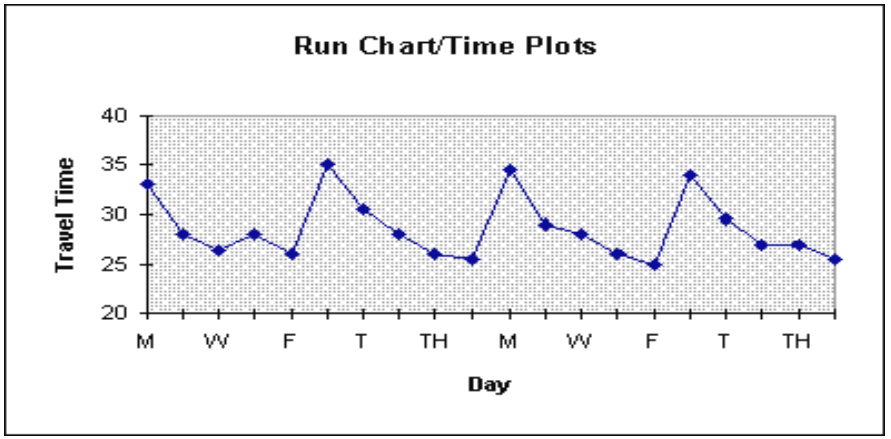
Run Charts

➢ Concept

- Run charts are used to analyze processes according to time or order.

➢ An Example of Using a Run Chart

- An organization's desire is to have their product arrive to their customers on time, but they have noticed that it doesn't take the same amount of time each day of the week. They decided to monitor the amount of time it takes to deliver their product over the next few weeks.



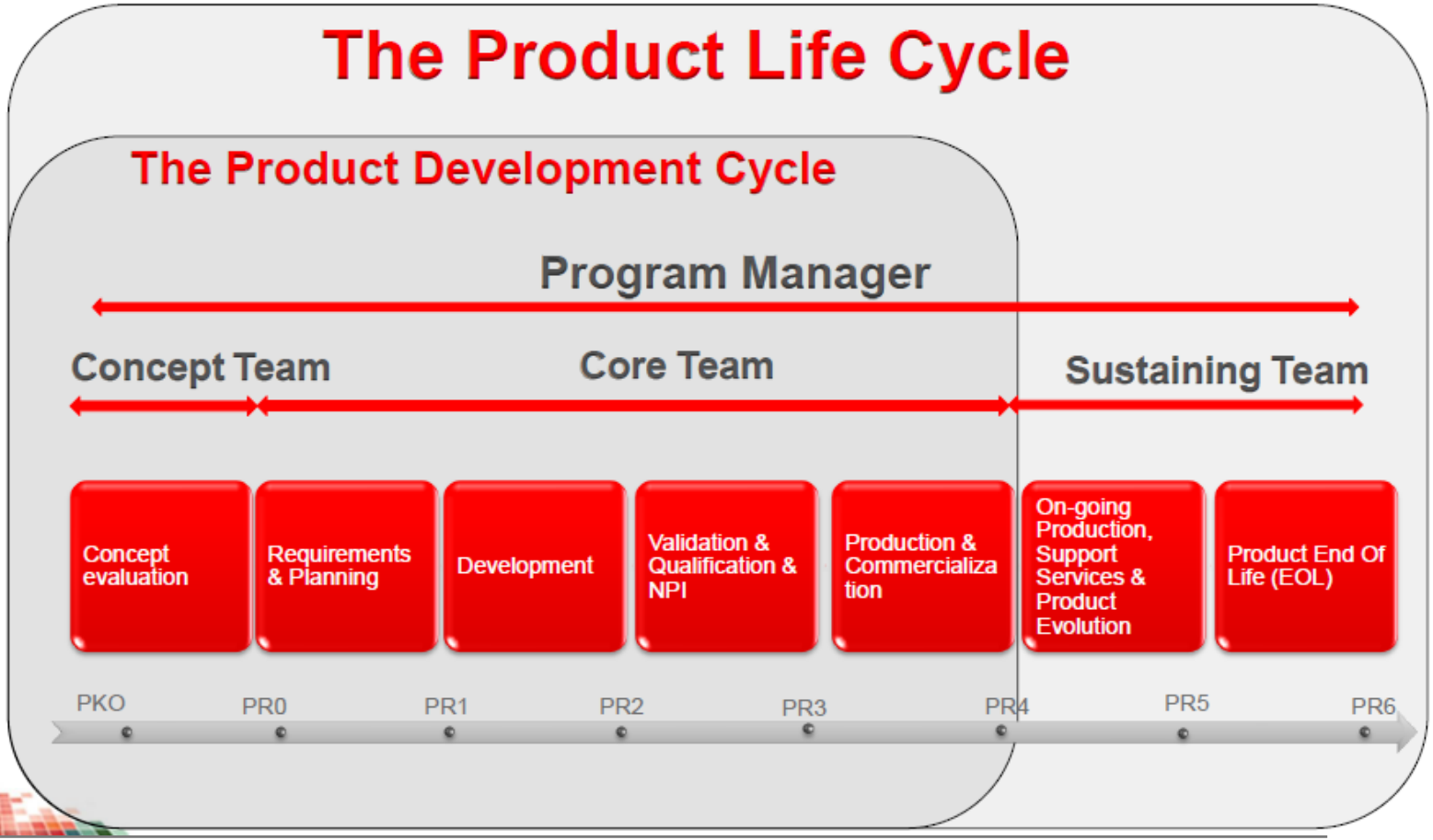
Exercises 2-1

- Time: 10 minutes discussion + 5 minutes/team Presentation

- Task
  - Select one Quality Control Tool
  - Set your team in a scenario that need to use this tool
  - Create your presentation:
    - Team name
    - Speaker’s name
    - Scenario introduction (or what issue you want to solve?)
    - How your team use this tool in this scenario?

Review & Inspection

Product Life Cycle



Types of Project Review & Why?

Review Type	Purpose
Educational Review	Break other stakeholders up to speed on technical topics pertinent to the project
Management, Readiness or Gate Review	Provide information to senior managers so they can decide to release a product, continues for cancel a development project, approve or reject a proposal change project scope, adjust resources, or alter commitment
Post-Project Review	Reflect on a recently completed project or phase to learn lessons for future projects
Status Review	Update the project manager and other team members on progress toward milestones, problem encountered, and risks identified or controlled
Peer Review	Look for defects and improvement opportunities in a work product

Exercises 2-2

- Time: 10 minutes discussion + 2minutes/team Answer Question

- Task
  - List review checkpoints in the whole Product Development Cycle – As many as possible.
  - What types are these reviews?
  - Who & when to review what?

SW Peer Review

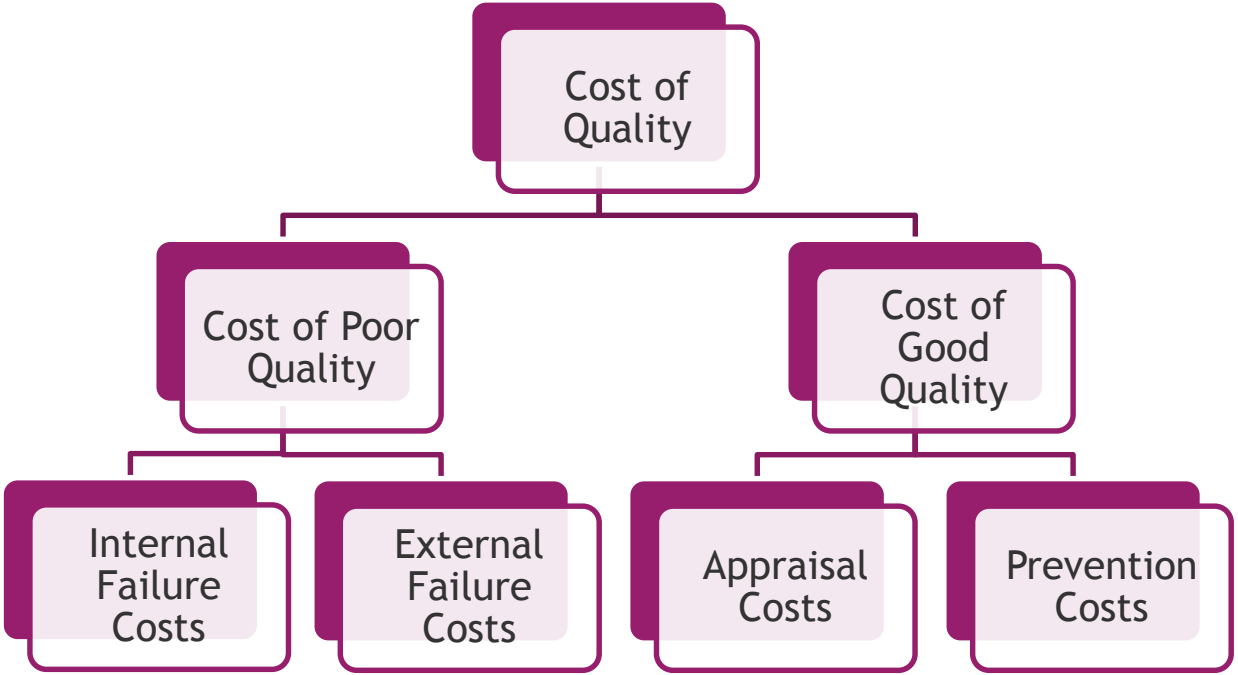
Why Peer Reviews?

Work products of higher quality

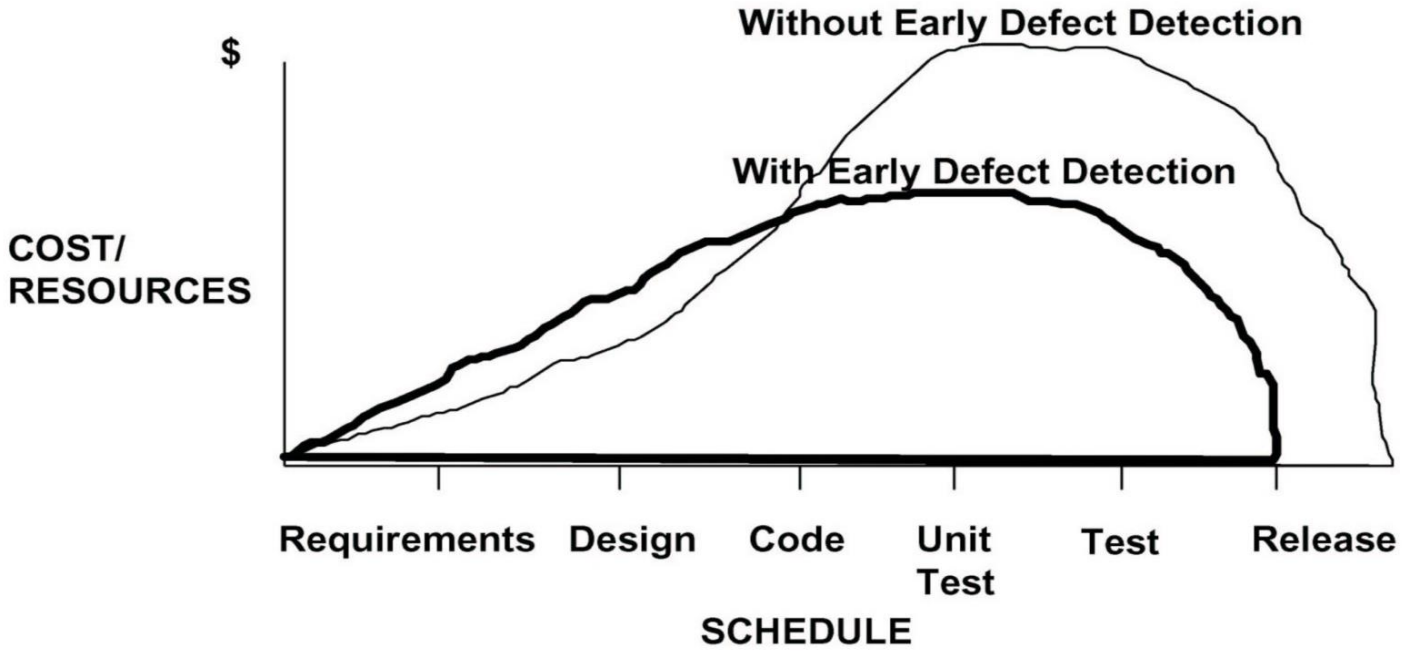
A good way to learn about work products

Peer reviews encourage teamwork and people to work together.

Quality Cost

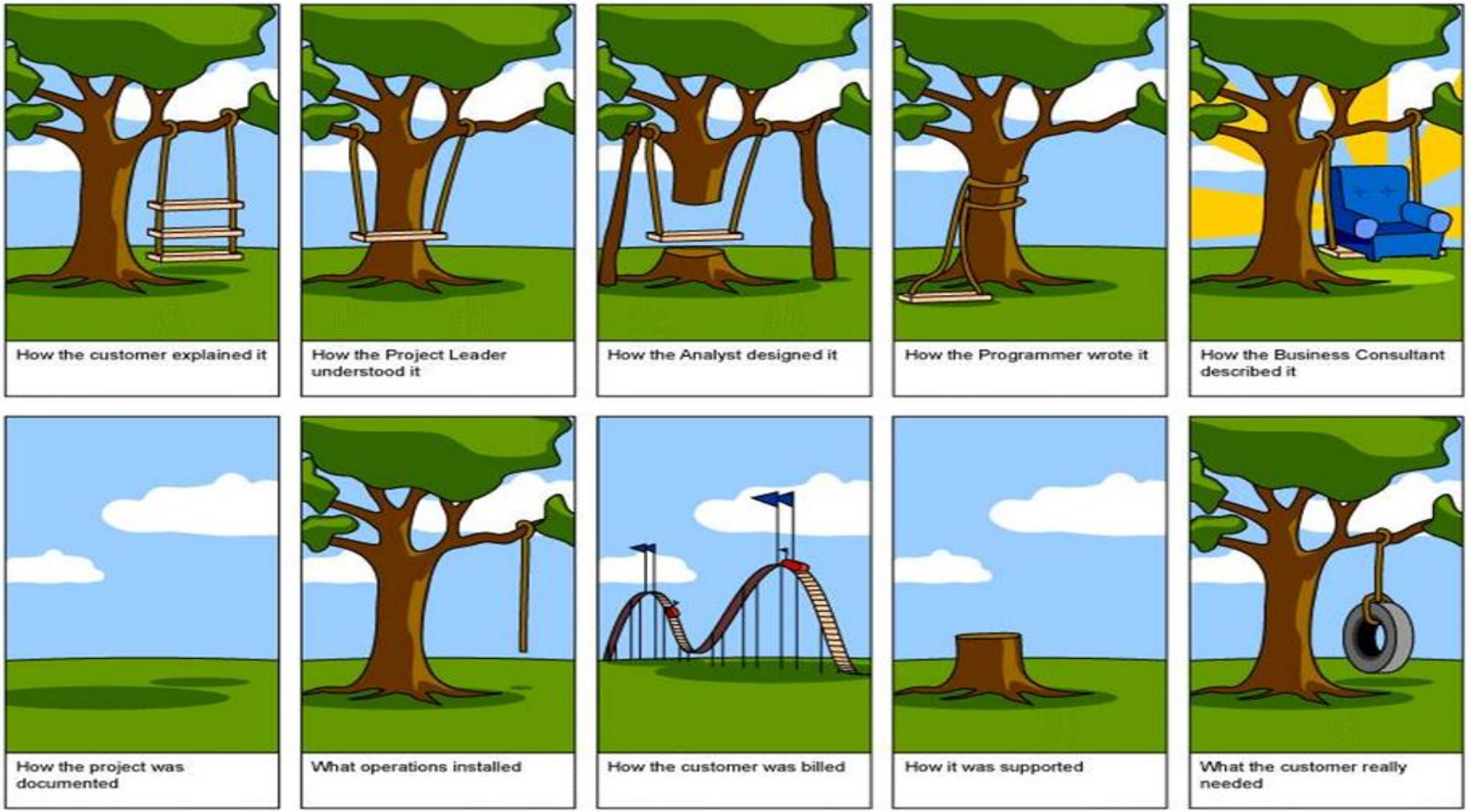


Quality Cost

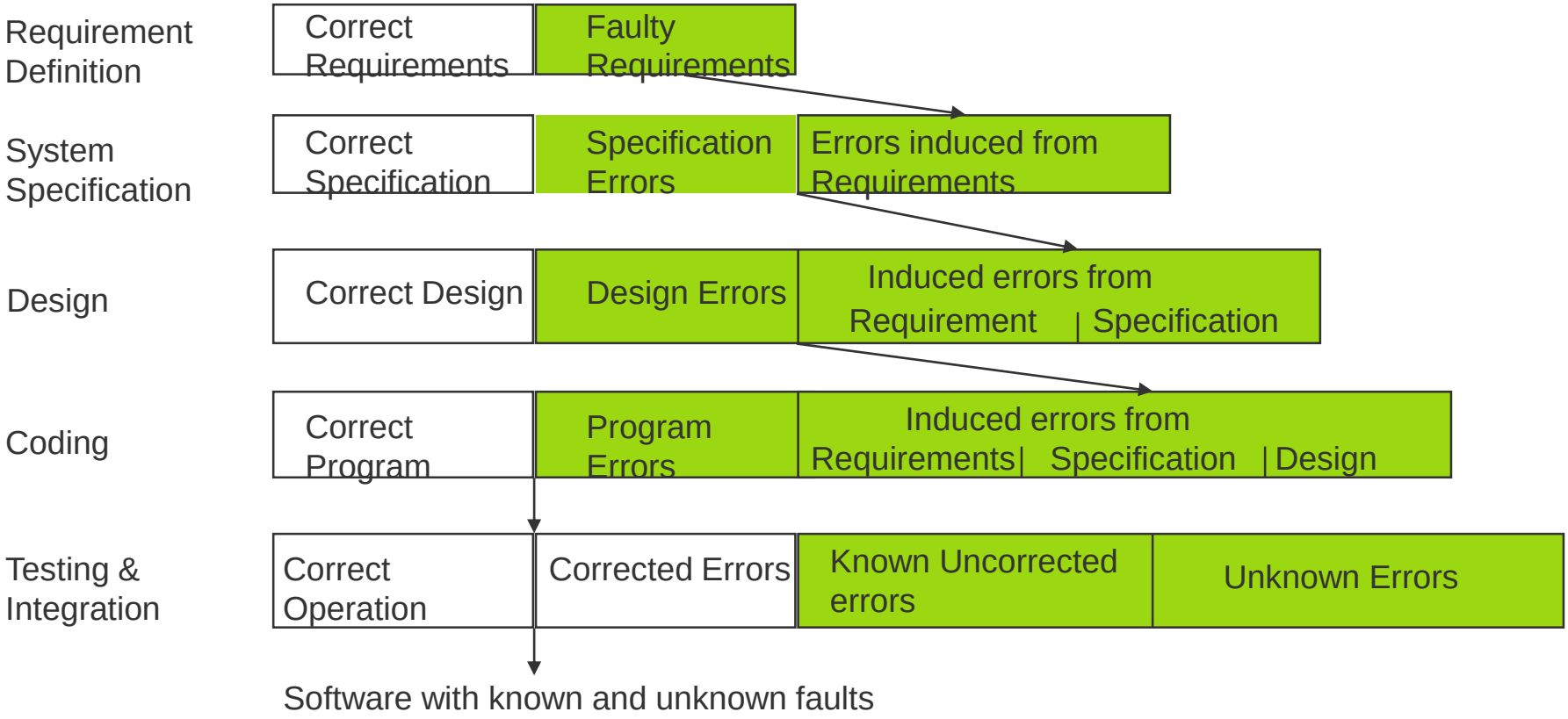


•Adapted from Fagan, "Advances in Software Inspections", IEEE, July 1986

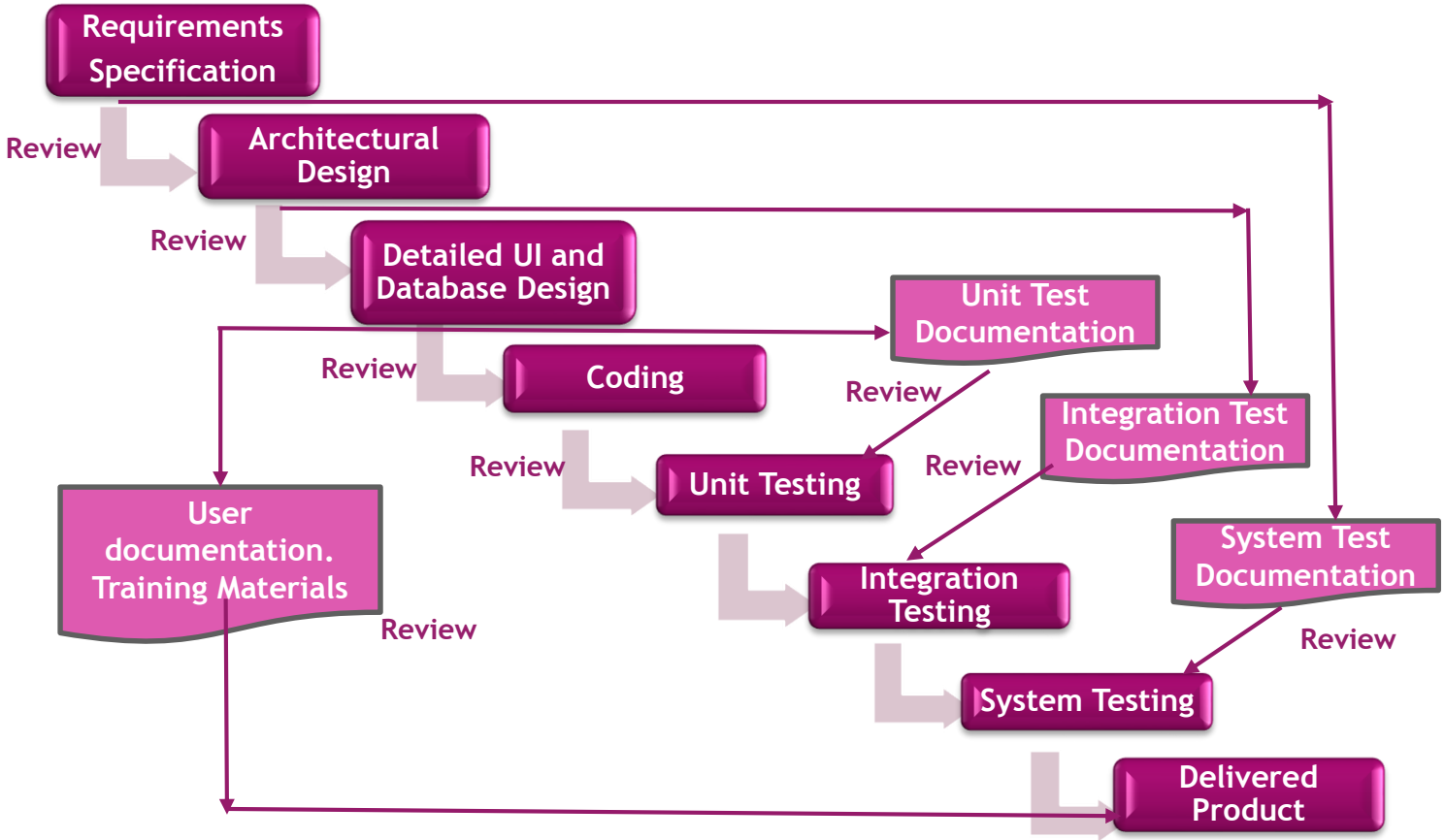
Accumulating Errors



Accumulating Errors



Peer Review Checkpoints in a SW Project



What Can Be Reviewed

1. Marketing documents, requirements spec, use cases, and analysis models
2. Business process models and business rules
3. Project Charter documents and all kinds of project plan
4. Architecture descriptions
5. User interface designs and prototypes
6. SW and DB design descriptions and models
7. Source code, including scripts, macros, stored procedures
8. Program documentation and system maintenance documentation
9. Test plans, designs, cases, and procedures
10. User guides, reference manuals, help screens, tutorials, training materials and field and customer support manuals
11. Build, release and installations procedures
12. SW DEV procedures, standards, and process descriptions

Peer Review Formality Spectrum & Activities

Most Formal

Least Formal

Walkthrough
- Informal
- The auth
solicits c
- Not follo
managem
- Finding
agreement as
- It's a good way
- It's appropriate
product.

Pair
- It
SV
- 2
si
d

Peer Deskcheck
- A deskcheck is a kind of self-re
- Only one person besides the au
- The peer deskcheck is the cheapest review approach.

Pass Around
- It's a multiple, concurrent peer deskcheck.
- It helps to mitigate two risks of a peer deskcheck: 1. No feedback from reviewer. 2. reviewers doing a poor job.

At Hoc Review
- It's fast and most informal

Peer Review Formality Spectrum & Activities

Most Formal — Least Formal

	Inspection	Team Review	Walkthrough	Pair Programming	Peer Deskcheck, Passaround	Ad hoc Review
Planning	**Yes**	**Yes**	**Yes**	**Yes**	**No**	**No**
Preparation	Yes	Yes	No	No	Yes	No
Meeting	Yes	Yes	Yes	Continuous	Possible	Yes
Correction	Yes	Yes	Yes	Yes	Yes	Yes
Verification	Yes	No	No	Yes	No	No

Suggested Review Method for Certain Objectives

Review Objective	Inspec -tion	Team Review	Walk-through	Pair Programming	Peer Deskcheck	Pass around
Find product defects	X	X	X	X	X	X
Check conformance to specifications	X	X			X	X
Check conformance to standard	X				X	X
Verify product completeness and correctness	X		X			
Assess understandability and maintainability	X	X		X		X
Demonstrate quality of critical or high-risk components	X					
Collect data for process improvement	X	X				
Measure document quality	X					
Educate other team member about products		X	X	X		X
Reach consensus on an approach		X	X	X		
Ensure that changes or bug fixes were correct		X	X		X	
Explore alternative approaches			X	X		
Simulate execution of a program minimize review cost			X		X	

Exercises 2-3

- Time: 10 minutes

- Task
  - Select one deliverable from slide 77 to do review
  - Which development phase should the review happen in?
  - What is the best type of this review?
  - Who should be invited to this review?
  - What's the objective of this review?
  - What activities should be carried out?
  - Advantages and disadvantages?

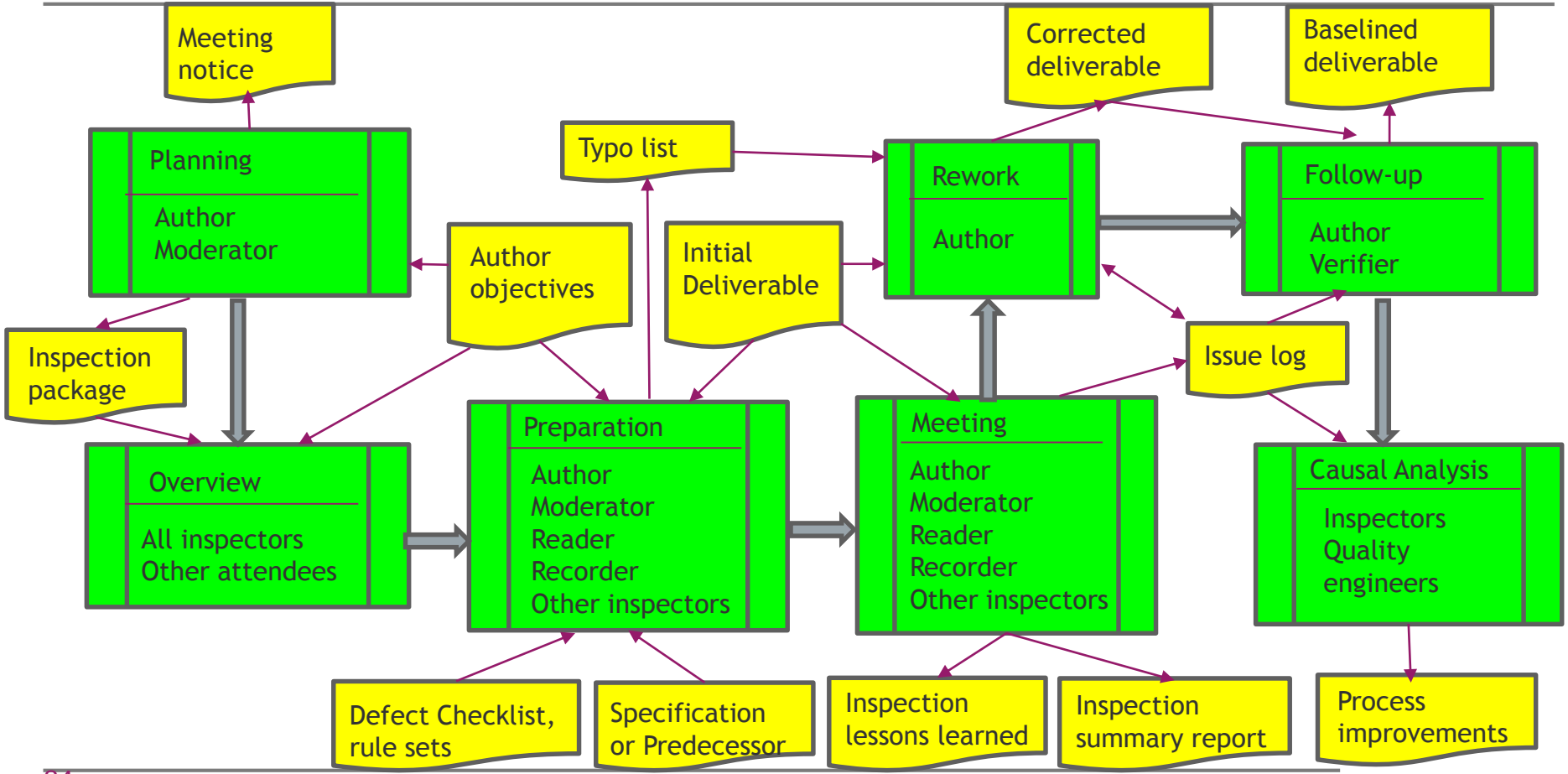
Think about：Is it correct to review all deliverables with inspection?

SW Inspection

Main Roles in Inspection Process

- **Author:** The person who created the work product being inspected, also called the producer or owner.
- **Moderator:** This is the leader of the inspection. The moderator works with the author to plans the inspection , keep the meetings on track, and leads the inspection team to a successful outcome.
- **Reader:** The person reading through the documents, one item at a time. The other inspectors then point out defects.
- **Recorder/Scribe:** The person that documents the defects that are found during the inspection.
- **Inspector:** The person that examines the work product to identify possible defects.
- **Verifier:** At the end of most inspection meetings, a verifier might be seleceted to check the author's rework.

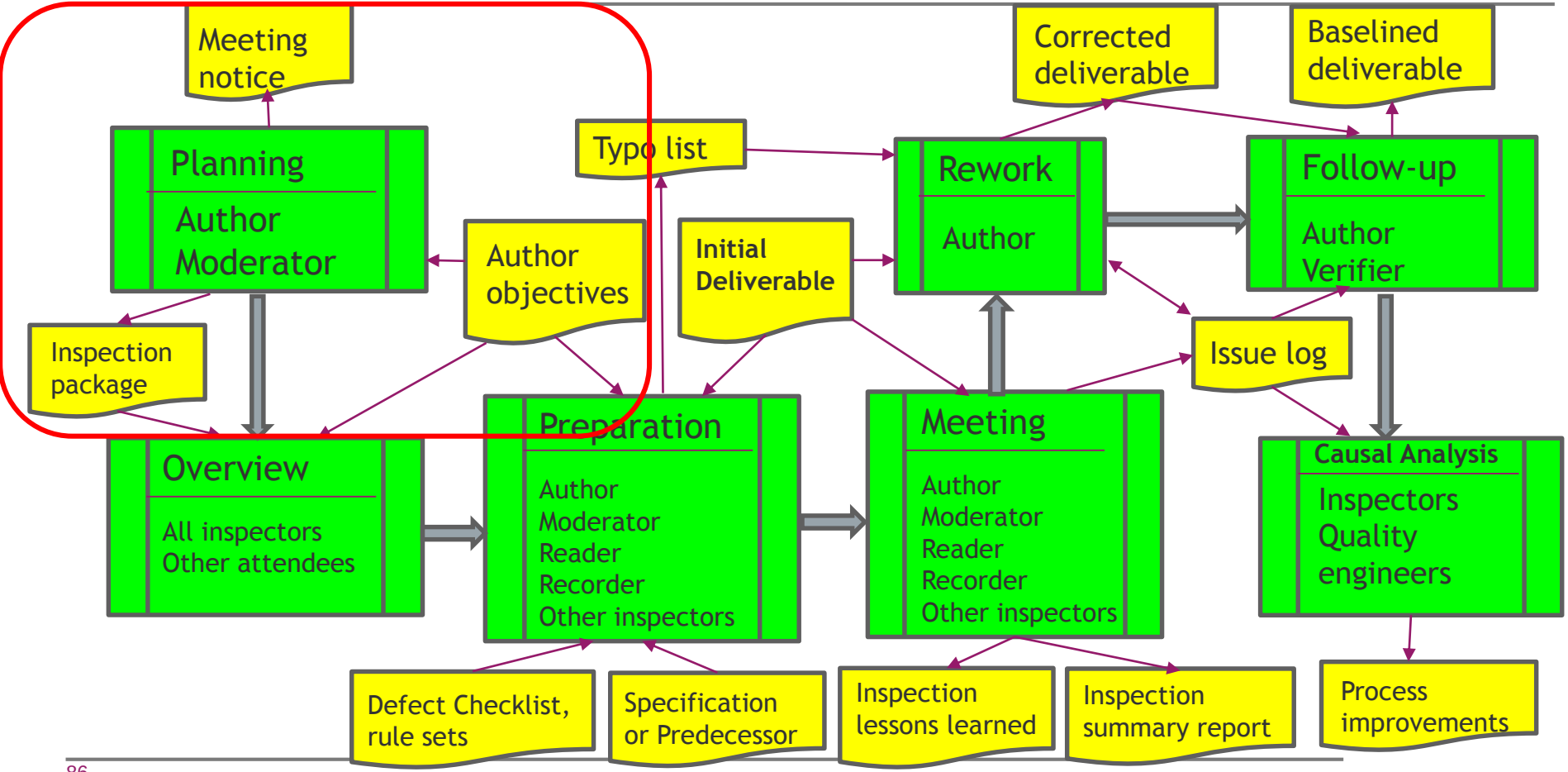
Inspection Process



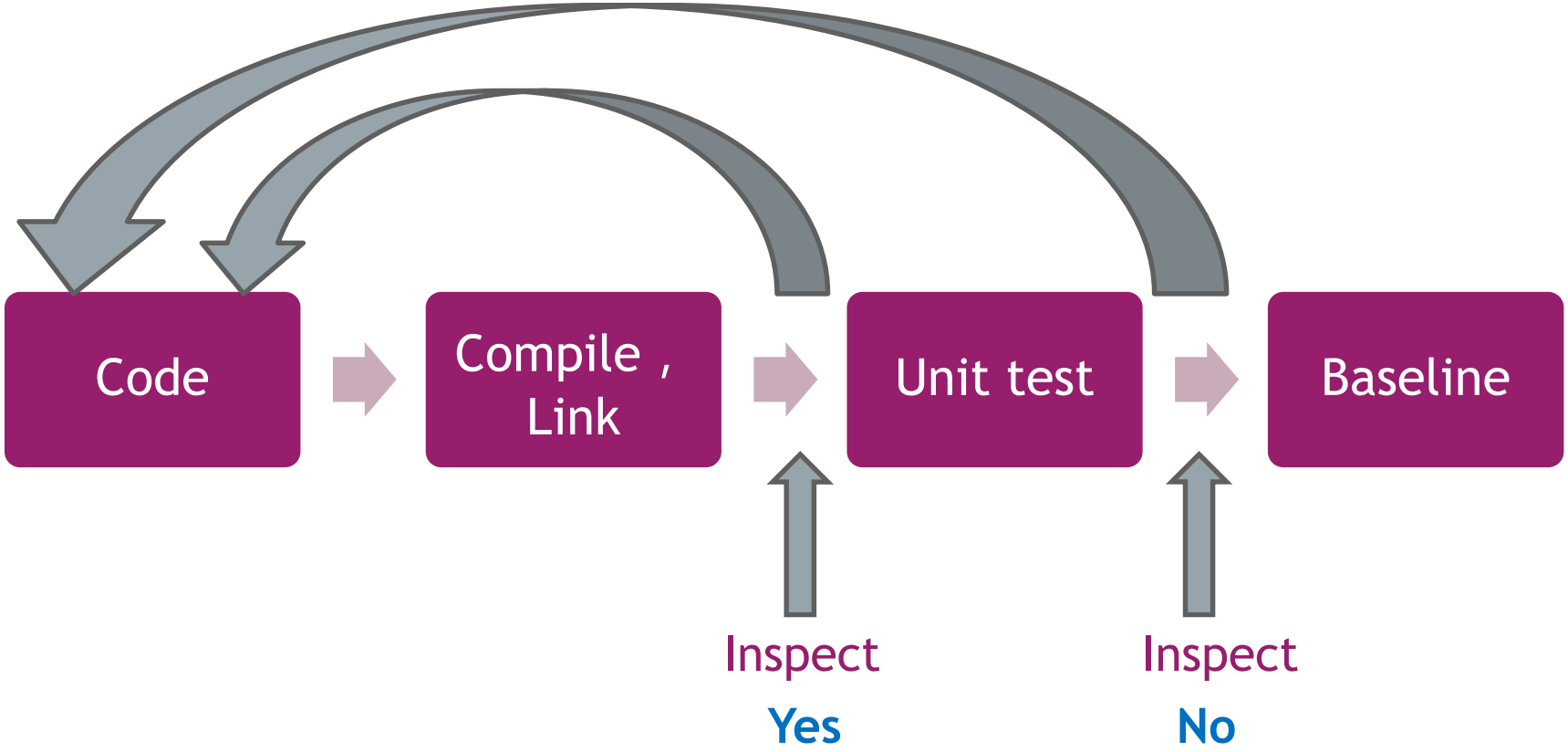
Inspection Documents

Documents	Comes From	Goes To
Author objectives	Author	Moderator, other inspector
Meeting notice	Moderator	Inspector, overview meeting attendees
Inspection package	Moderator, Author	Inspector
Initial deliverable	Author	Inspector
Defect checklist, rules sets	Process assets library	Inspector
Specification or predecessor	Author of spec for the initial deliverable or its predecessor document	Inspector
Typo lists	Inspector	Author
Issue log	Recorder	Author, moderator, verifier
Corrected deliverable	Author	Verifier
Baseline deliverable	Author	Project configuration management system
Inspection summary report	Moderator	Peer review coordinator, management
Inspection lesson & learn	Moderator	Peer review process owner, coordinator
Process improvements	Inspector	Organization SEPG

Inspection Process – Planning (Individual Inspection)



Inspection Process – Planning (Individual Inspection)



Inspection Process – Planning (Individual Inspection)

Major Planning Steps:

1. Assign a moderator

The best moderator share several characteristics:

Select the material based on the risk of its containing errors that could introduce

General 1st: All participants have been trained in the inspection process.

Roles:

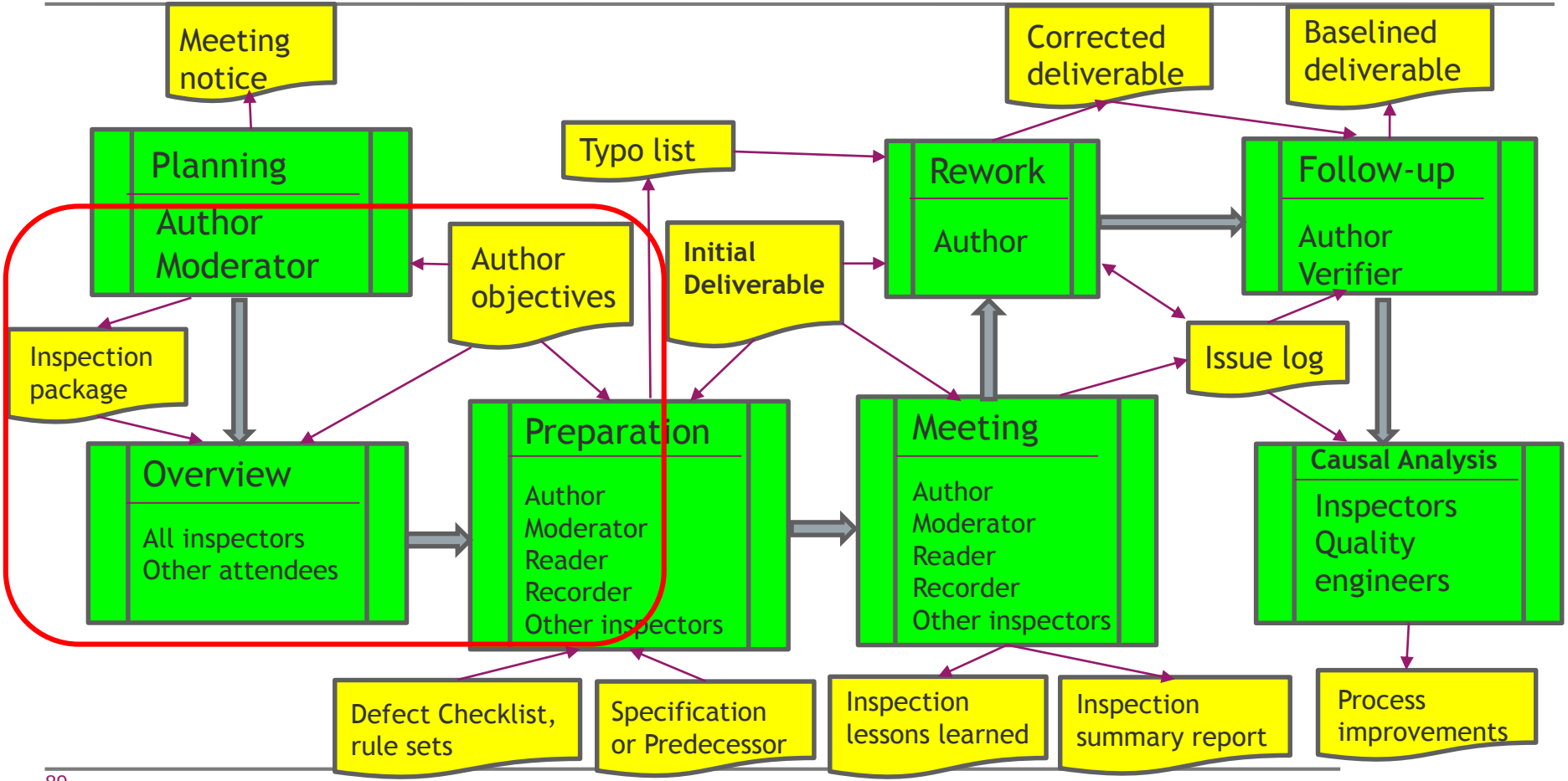
Typical inspection package:

Meeting notice:

- Time
- Location
- People

PS: send out meeting notice one week in advance. One inspection per day (for one team)

Inspection Process - Overview

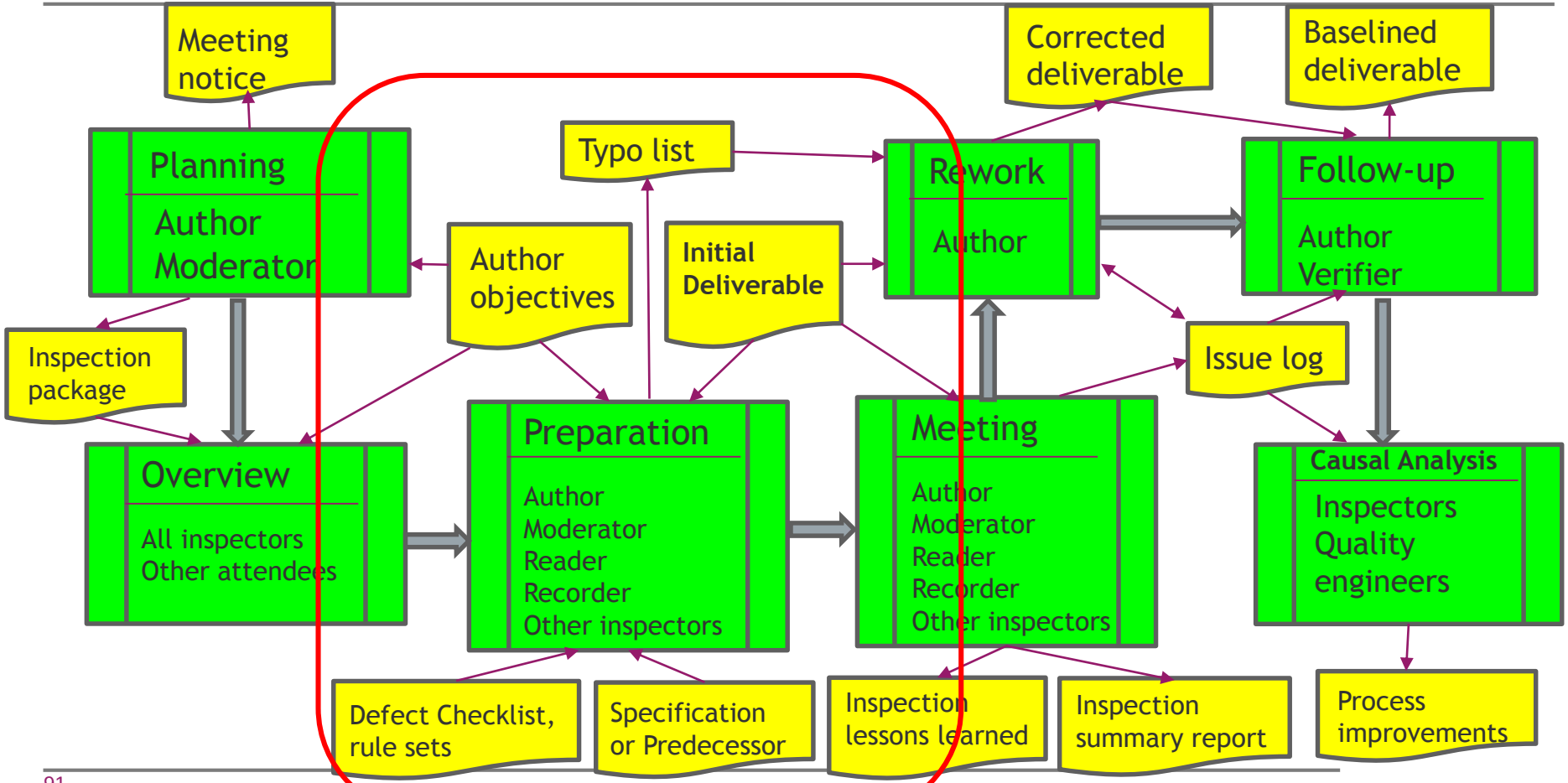


Inspection Process - Overview

Topics might address

1. The program's operating environment
2. Inputs and outputs parameters
3. Module preconditions and post conditions
4. Exception-handling strategies
5. Significant data structures
6. Interfaces to users or to other SW/HW components.
7. Usage scenarios
8. Architectural context

Inspection Process - Preparation

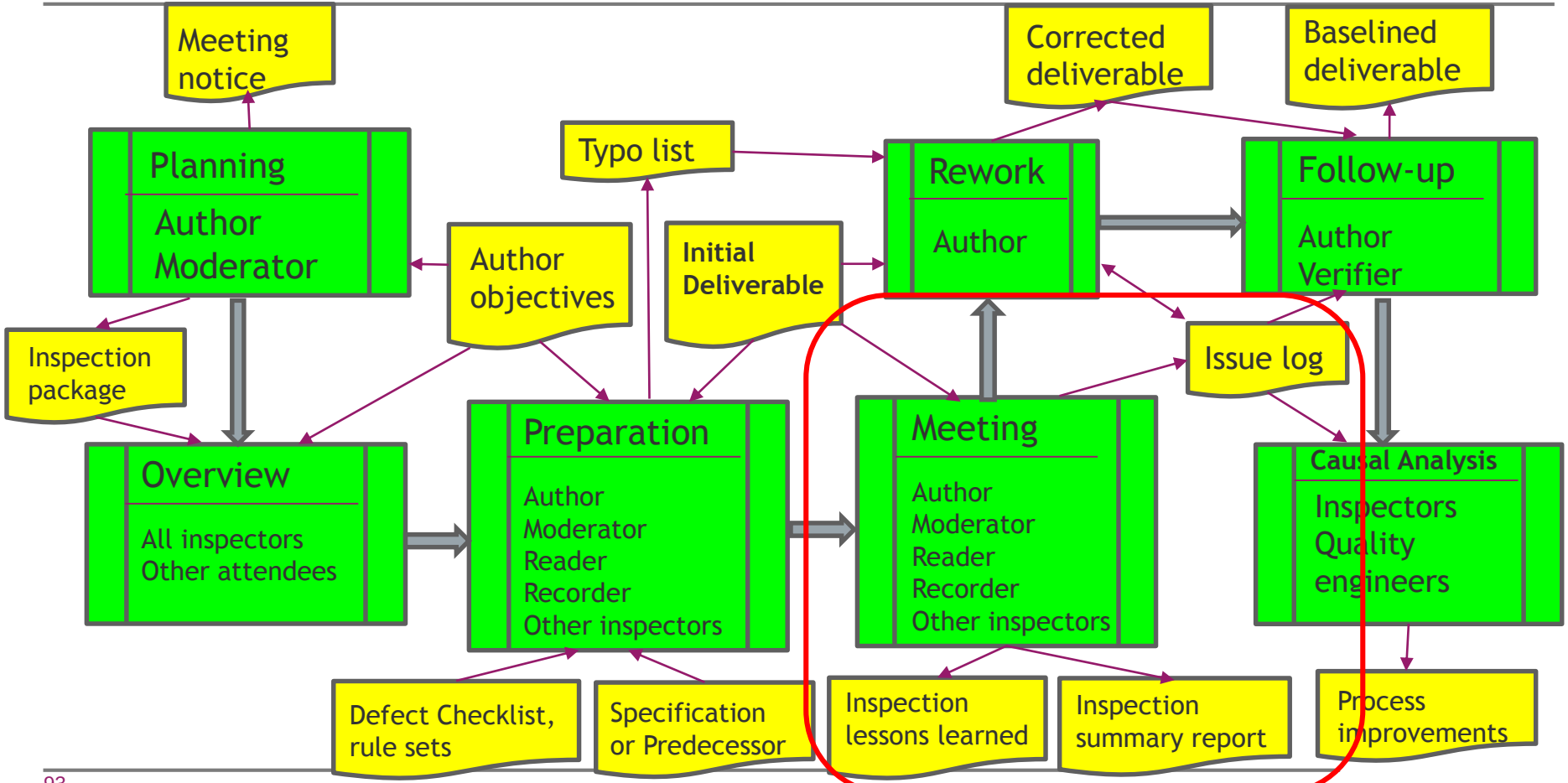


Inspection Process - Preparation

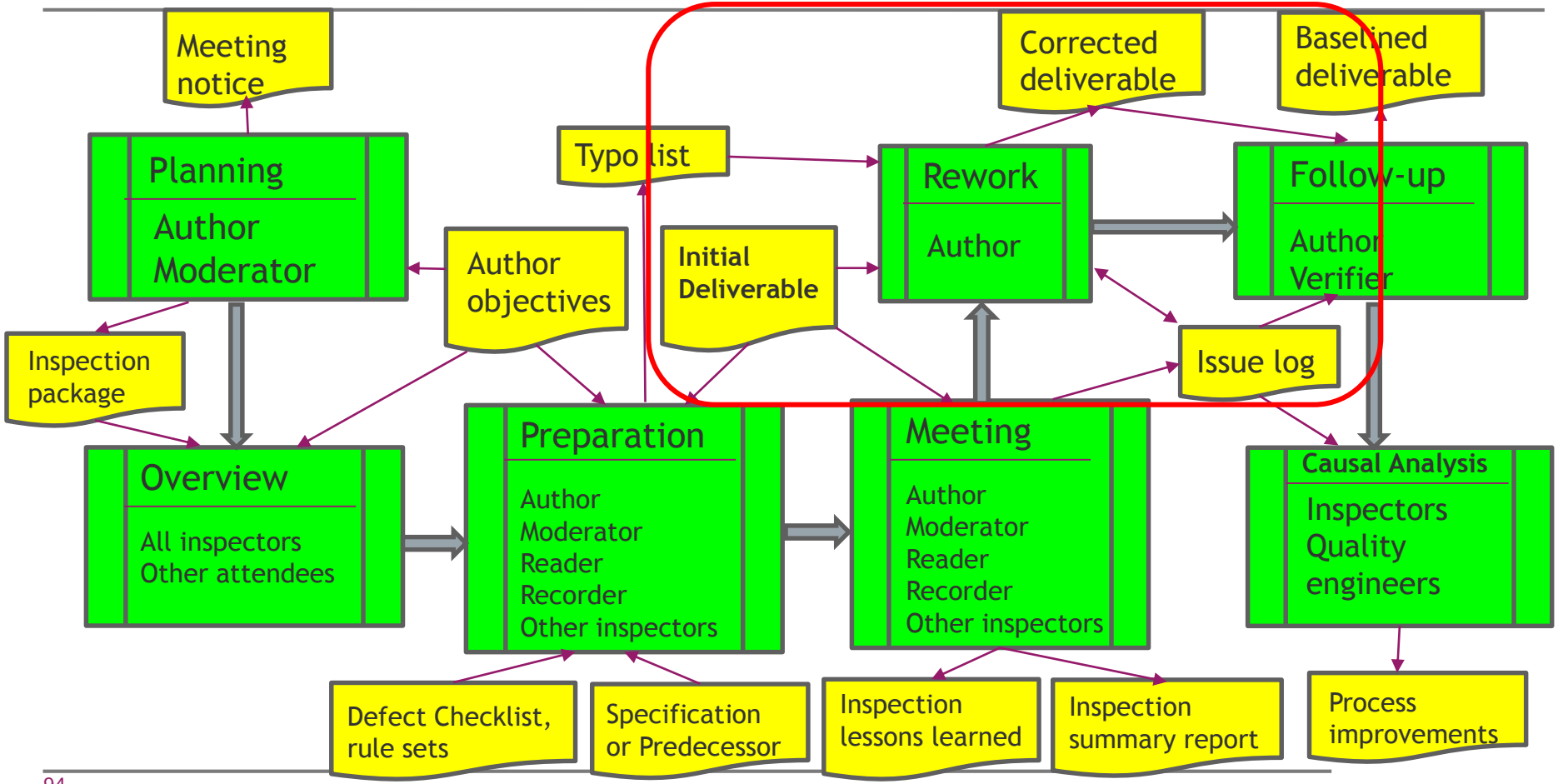
Topics might address

1. The program's operating environment
2. Inputs and outputs parameters
3. Module preconditions and post conditions
4. Exception-handling strategies
5. Significant data structures
6. Interfaces to users or to other SW/HW components.
7. Usage scenarios
8. Architectural context

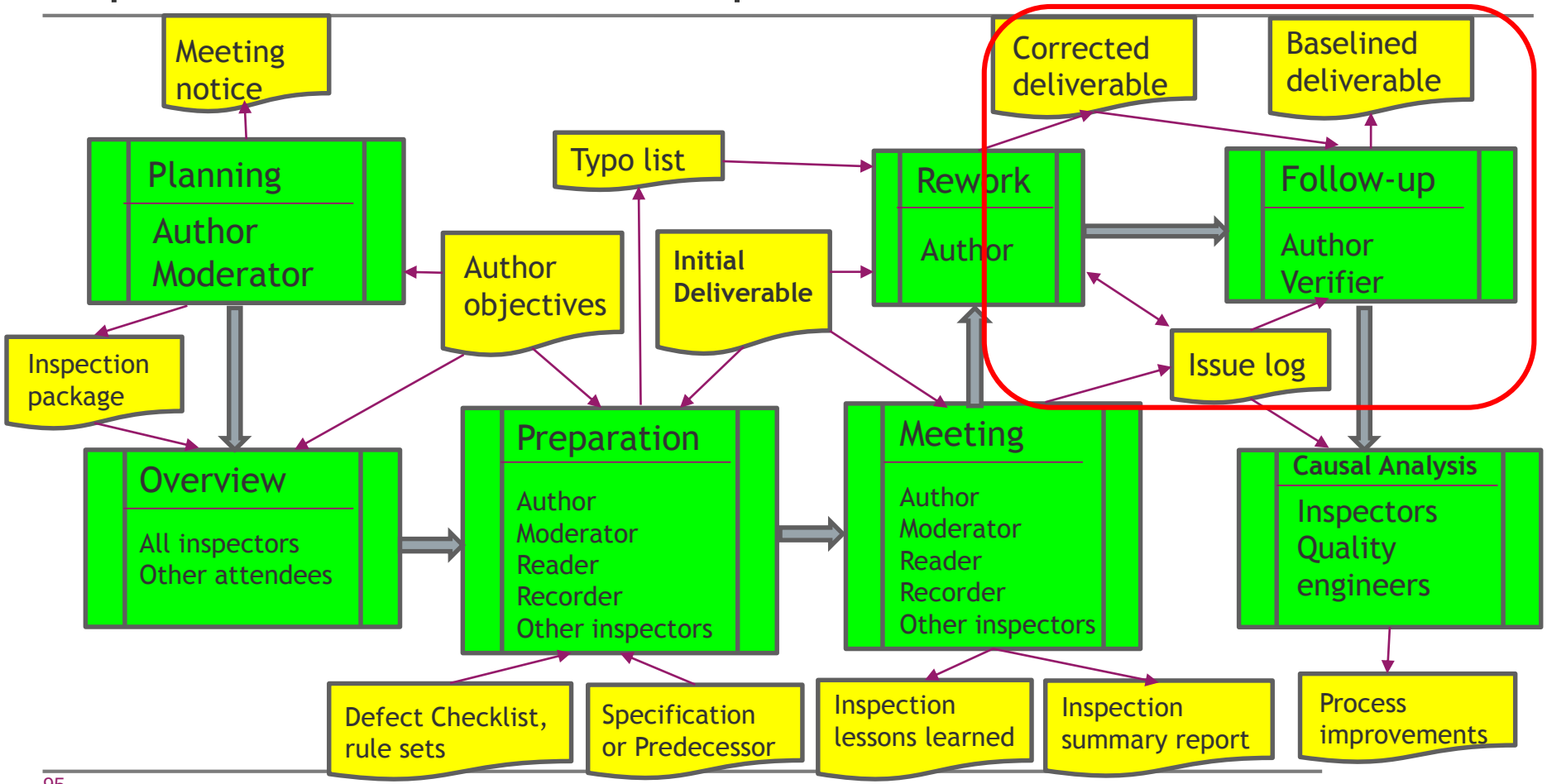
Inspection Process - Meeting



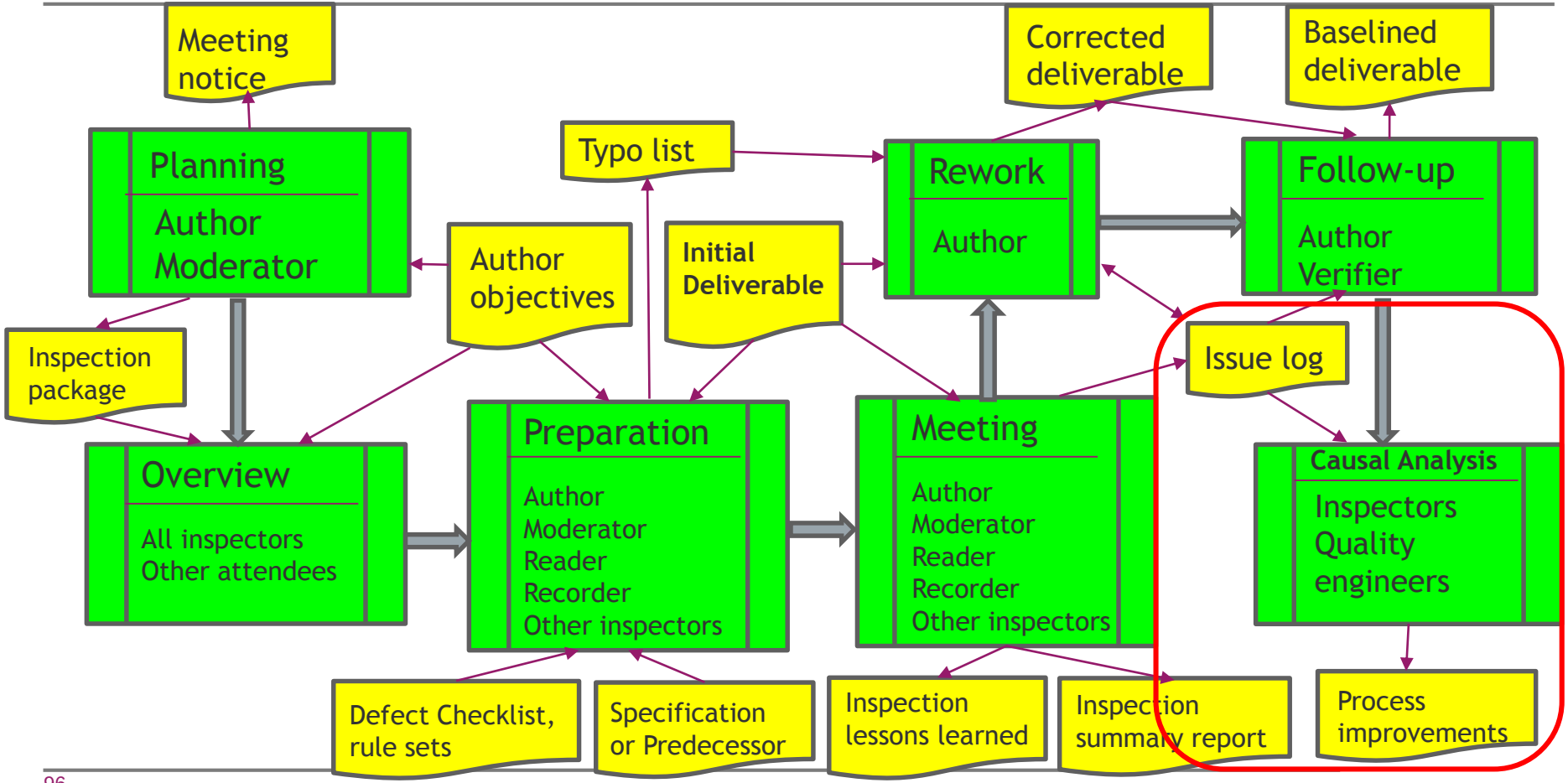
Inspection Process - Rework



Inspection Process – Follow up



Inspection Process – Causal Analysis



Exercise 2-4 – Review Practice

We are going to develop an APP (Name: 臭美). It contains below functions:

- Store pictures of my clothes, shoes, bags and accessories.
- Can search according to name, type, color
- Can select collection
- Remind on a special day and recommend suitable collection

Requirements:

- Create a requirement specification document according to above APP functions
- Organize review team (6-7 person each team) and define roles.
- Choose a proper review method
- Create a process map of Inspection method
- Create outputs of each processes
- Identify the contribution of each role in each process

Presentation:

- Each team have 10 minutes to present the review result and lesson & learn

Analyzing Inspection Data

- Why Collect Data?
- Some Measurement Caveats
- Basic Data Items and Metrics
- The Inspection Database
- Data Analysis
- Measuring the Impact of Inspections
  - Effectiveness
  - Efficiency
  - ROI

Analyzing Inspection Data

- Why Collect Data?
  - Data answers important questions
  - Data provide quantifiable insights and historical perspectives
  - Lets you base decisions on facts instead of perceptions or memories.

- How to choose a right metrics?
  - Goal-Question-Metric (GQM)
    - Set goal – E.g. to reduce your rework costs through inspection.
    - Identify questions.
    - Select metrics that will let you answer those questions.
  - Mistakes need to avoid - Too elaborate

Analyzing Inspection Data

- Some Measurement Caveats

  - It’s important to be honest and non-judgemental about metrics

  - Measurement dysfunctions. E.g.
    - defect severity inflation or deflation
    - Artificial defect closure
    - Distorted defect densities, preparation time and defect discovery.

  - Avoid to draw conclusion from data collected shortly.

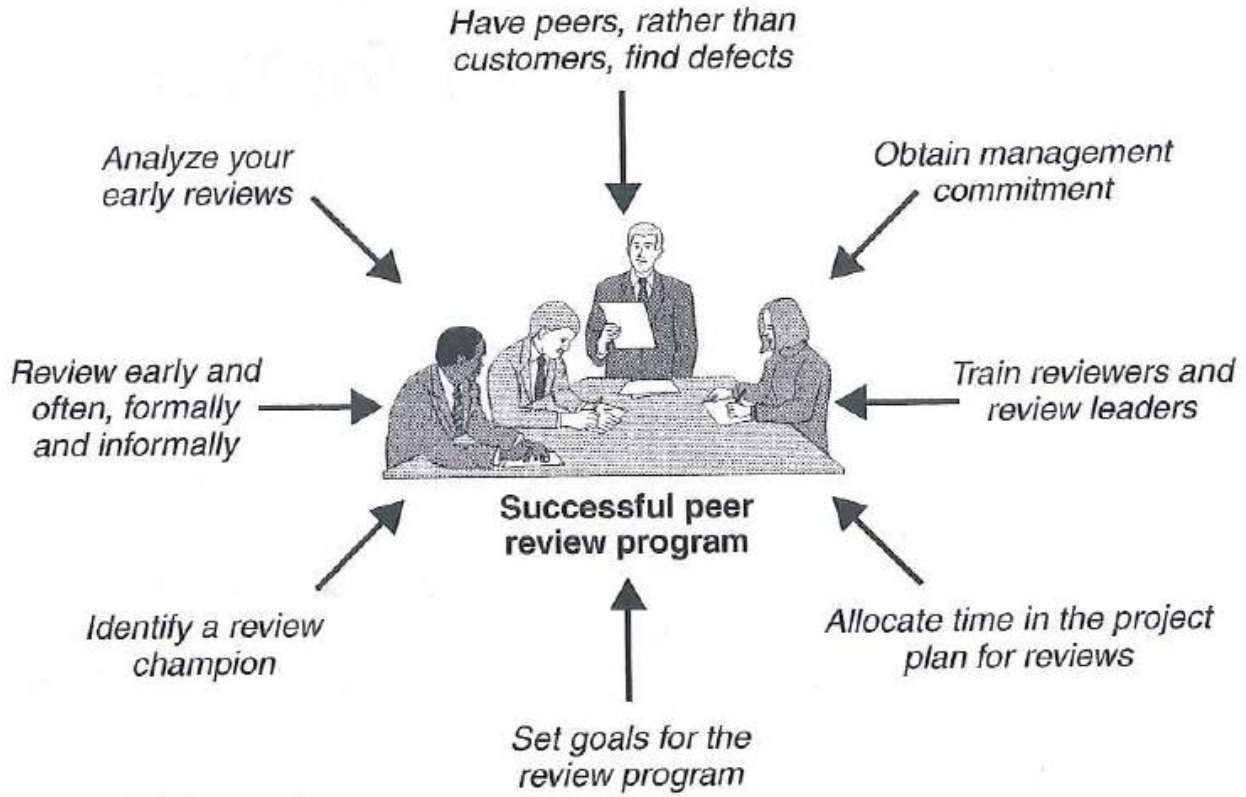
Analyzing Inspection Data

- Basic Data Items and Metrics

Category	Data Item	Description
Size	Size. Planned	Lines of code or document pages that you planned to inspect
	Size. Actual	Lines of code or document pages that were actually inspected
Time	Time. Meeting	Duration of the inspection meeting in hours, if several meetings were needed to complete the inspection, include them all in this total
Effort	Effort. Planning	Total labor hours the moderator and author spend on planning, scheduling meetings, assembling and distributing the inspection package.
	Effort. Overview	Total labor hours spent on the overview stage.
	Effort. Preparation	Total labor hours spent on individual preparation.
	Effort. Meeting	Total labor hours spent in the inspection meeting
	Effort. Rework	Total labor hours of author spent correcting defects, including verification time.
Defects	Defects. Found	
	Defects. Corrected	
Others	Numbers of inspectors	
	Product. Appraisal	Accepted? Conditionally accepted? Rejected? Inspection not completed?

Improve Effectiveness and efficiency of Peer Review

Critical Success Factors



Traps to Avoid

- Participants don’t understand the review process
- The review process isn’t followed
- The right people do not participate
- Review meetings drift into problem-solving
- Reviewers focus on style, not substance.

Discussion: How to solve below issues in inspection?

Cultural Issues

- Some team members refuse to have their work reviewed
- Some team members refuse to review other people's work
- Reviews are unnecessarily brutal, Personal attacks and innuendo are common. Authors become defensive.
- The moderator does not control the review meeting effectively

Discussion: How to solve below issues in inspection?

Planning issues

- Reviews do not appear in the project plan
- Reviews are perceived to slow down the project
- Reviews are skipped in a crisis or time crunch
- People do not review the appropriate portions of their work products
- Roles are not clear among the review participants

Discussion: How to solve below issues in inspection?

Management Issues

- Management does not support peer review or actively opposes them
- Management does not set clear expectation for reviews
- Management want to participate in reviews when they should not
- Managers use metrics date inappropriately or ask to view data they are not entitled to see
- Data collected during the review is not used for anything
-

Discussion: How to solve below issues in inspection?

Effectiveness Issues

- Review participants choose inappropriate preparation methods or analysis techniques
- Discussions during review meetings to revisit decisions made long ago or question the work product's background
- Participants are not adequately prepared for review meetings
- Review find certain kinds of defects but frequently miss others. Many defects are not caught by the reviews
- Reviews keep finding the same kinds of defects
- Reviews find to many defects
- Reviews find many minor, but few major defects
- Reviews are held too late to be effective. Excessive rework is required to correct all the review findings
- Rework is not done correctly. Bad fixes are common.

Special Review Challenges

Large work products
Geographical or time separation
Generated and nonprocedural code
Too many participants
No qualified reviewers available

Workshop & Practise

Choose one of below items to do peer review

- Codes
- Document
- Product

Requirements:

- Organize review team (6-7 person each team) and define roles.
- Choose a proper review method
- Create a process map of chosen review method

Presentation:

- Each team have 5 minutes to present the review result and lesson & learn

Chapter No. 3 -- Quality Assurance

Self - introduction

Zhou Hui

- Product Quality Assurance of APAC Region
  - PQA based at Beijing site
- Connected Home- APAC R&D Support Program
  - Key customers covered are DIRECTV (satellite) - USA, Telstra -Australia , StarSHB - Singapore, SKY Mexico/Brazil, COMCAST (cable) – USA, TATA Sky – India, Airtel - India and Great China

Working Experience

- Joined Technicolor in 2008
  - R&D Process Management
  - SW Outsourcing process
  - Product Quality Assurance

Education & Credentials

- Bachelor of Electronics Engineering – National University of Defense Technology
- Master of Computer Engineering- Beijing University of Aeronautics and Astronautics
- Certified Quality Engineer- ASQ(American)

Email Address: 2216457976@qq.com

Class Objectives

Unit	Knowledge Points	Requirements	Suggested Hours
Qulality Assurance	International quality management model: CMMI	master	4
	International quality management model: Agile	master	5
	Quality assurance process	understand	0.5
	Personal software process	understand	0.5
Senior Quality Management	Defect management	master	2
	Defect analysis	understand	1
	Software quality model and quality analysis	know	1

International quality management model: CMMI

CMMI History

1993 – CMM (Capability Maturity Model) version 1.1

- Software Engineering Institute, sponsored by the U.S. Department of Defense (DoD) and based at Carnegie Mellon University
- http://www.sei.cmu.edu/

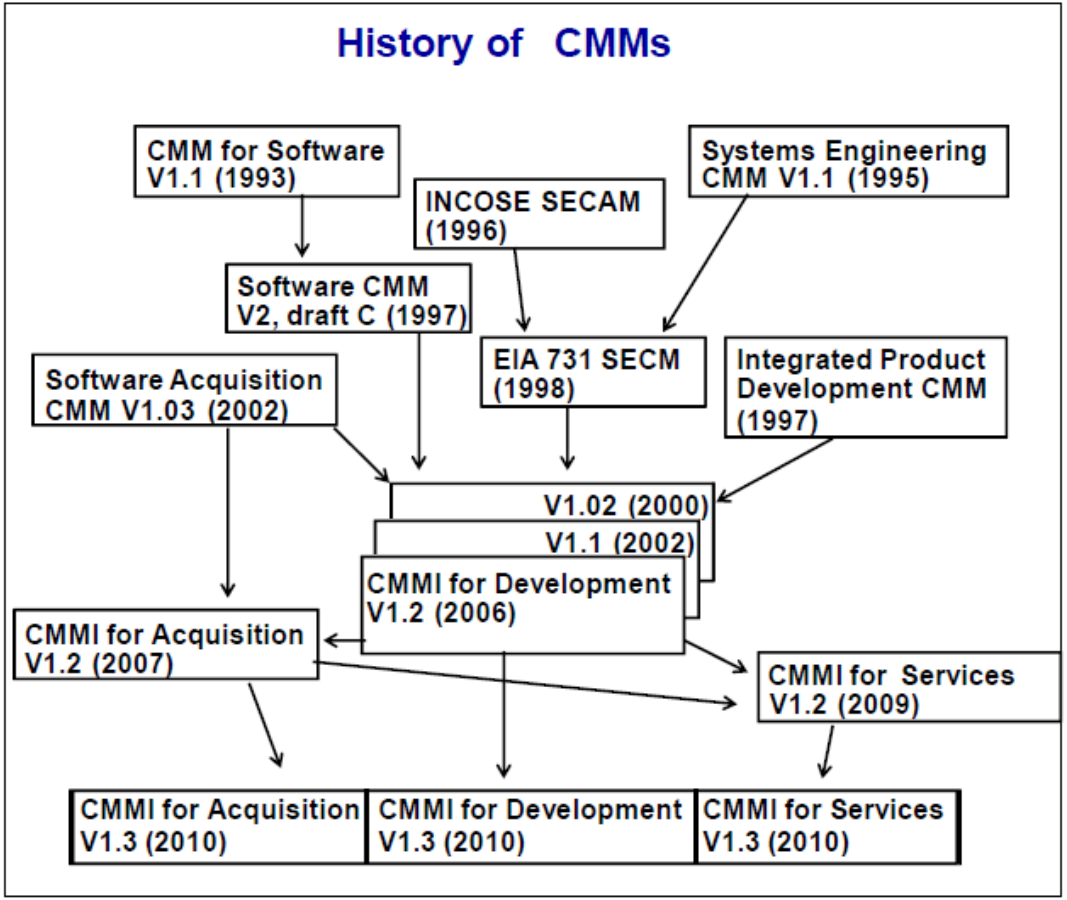
2000 – CMMI (Capability Maturity Model Integration) version 1.0

31 December 2005 – SEI sunsets CMM

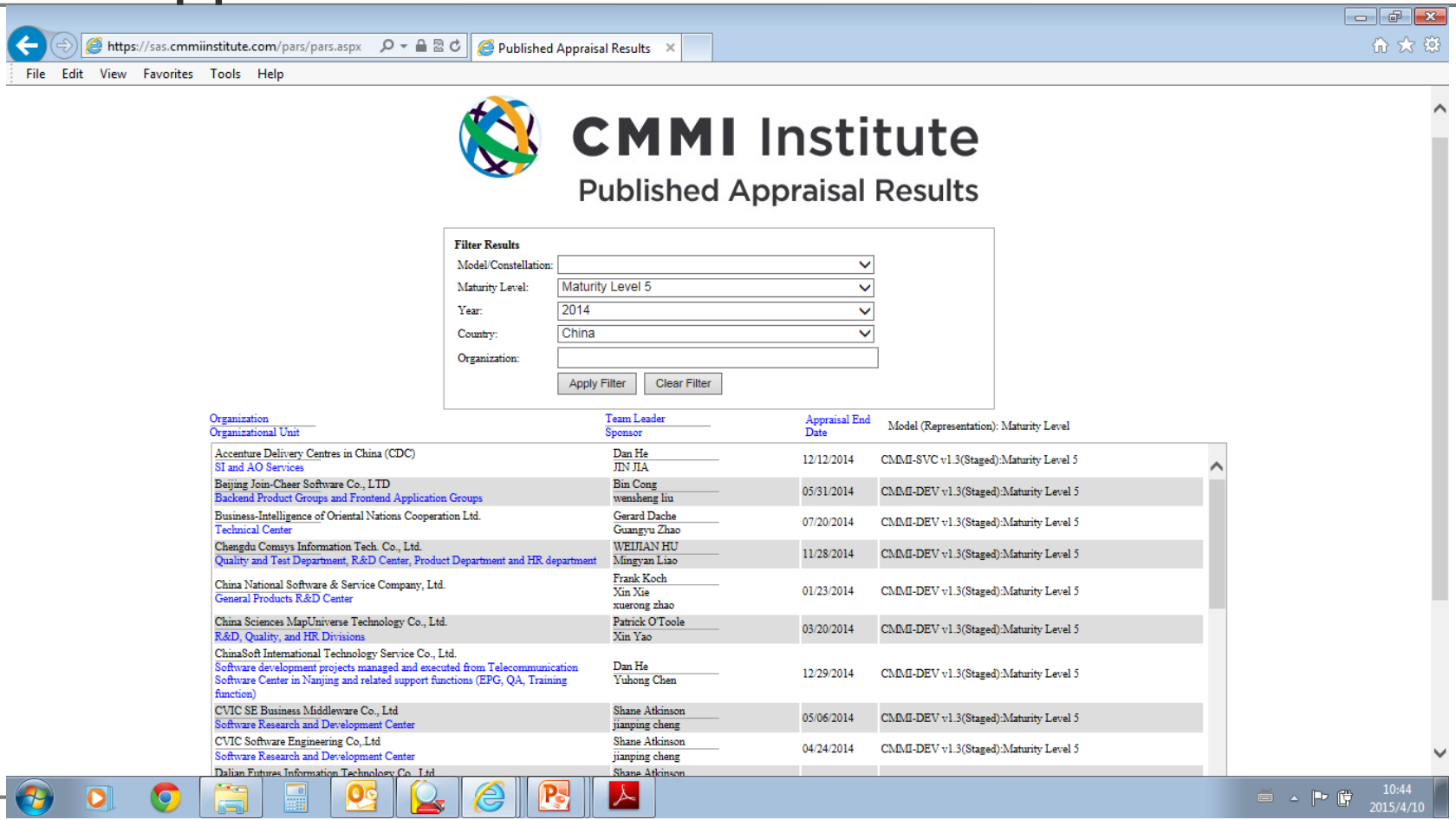
August 2006 – CMMI version 1.2

November 2010 – CMMI version 1.3

- CMMI Institute is part of Carnegie Innovations, a technology commercialization enterprise and part of Carnegie Mellon University
- http://www.cmmiinstitute.com/



CMMI Appraisal Result



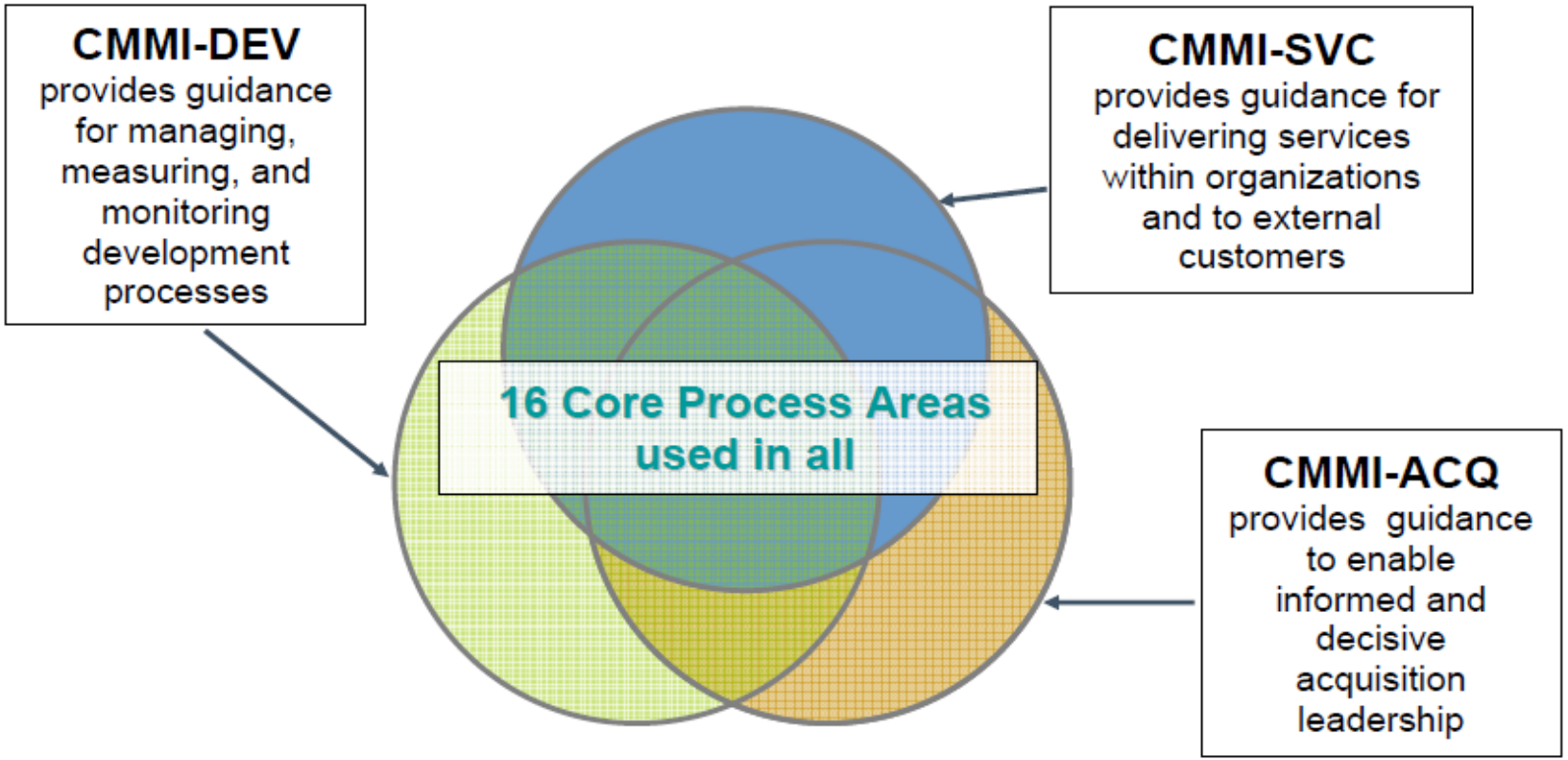
CMMI Concept

A CMMI model is not a process.

A CMMI model describes the characteristics of effective processes.

The Capability Maturity Model Integration (CMMI®) is a **process improvement model** that can be adapted to solve any performance issue at any level of the organization in any industry. The Model provides guidelines and recommendations for helping your organization diagnose problems and improve performance. Used by over 5000 organizations from more than 70 countries all over the world, CMMI helps you identify and achieve measurable business goals.

Three Complementary Constellations

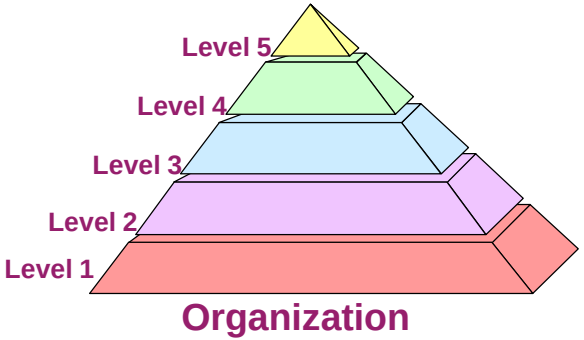


Continuous and Staged Representation

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

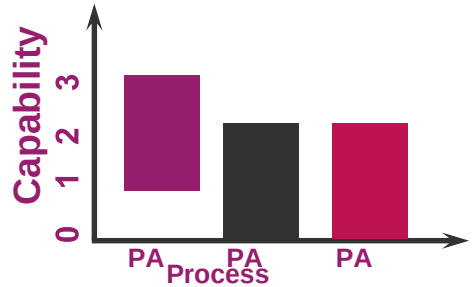
CMMI-DEV IN A NUTSHELL

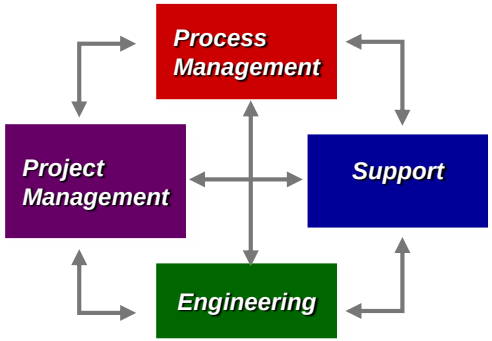
Staged



Name	Area	Level
Requirements Management	Project Management	2
Project Monitoring and Control	Project Management	2
Project Planning	Project Management	2
Supplier Agreement Management	Project Management	2
Configuration Management	Support	2
Measurement and Analysis	Support	2
Process and Product Quality Assurance	Support	2
Product Integration	Engineering	3
Requirements Development	Engineering	3
Technical Solution	Engineering	3
Validation	Engineering	3
Verification	Engineering	3
Organizational Process Definition	Process Management	3
Organizational Process Focus	Process Management	3
Organizational Training	Process Management	3
Integrated Project Management	Project Management	3
Risk Management	Project Management	3
Decision Analysis and Resolution	Support	3
Organizational Process Performance	Process Management	4
Quantitative Project Management	Project Management	4
Organizational Performance Management	Process Management	5
Causal Analysis and Resolution	Support	5

Continuous





CMMI Appraisal

The SCAMPI A appraisal method is the generally accepted method used for conducting ARC Class A appraisals using CMMI models.

The SCAMPI family of appraisals includes Class A, B, and C appraisal methods.

- The SCAMPI A appraisal method is the officially recognized and most rigorous method. It is the only method that can result in benchmark quality ratings.
- SCAMPI B and C appraisal methods provide organizations with improvement information that is less formal than the results of a SCAMPI A appraisal, but nonetheless helps the organization to identify improvement opportunities.

CMMI Investigation

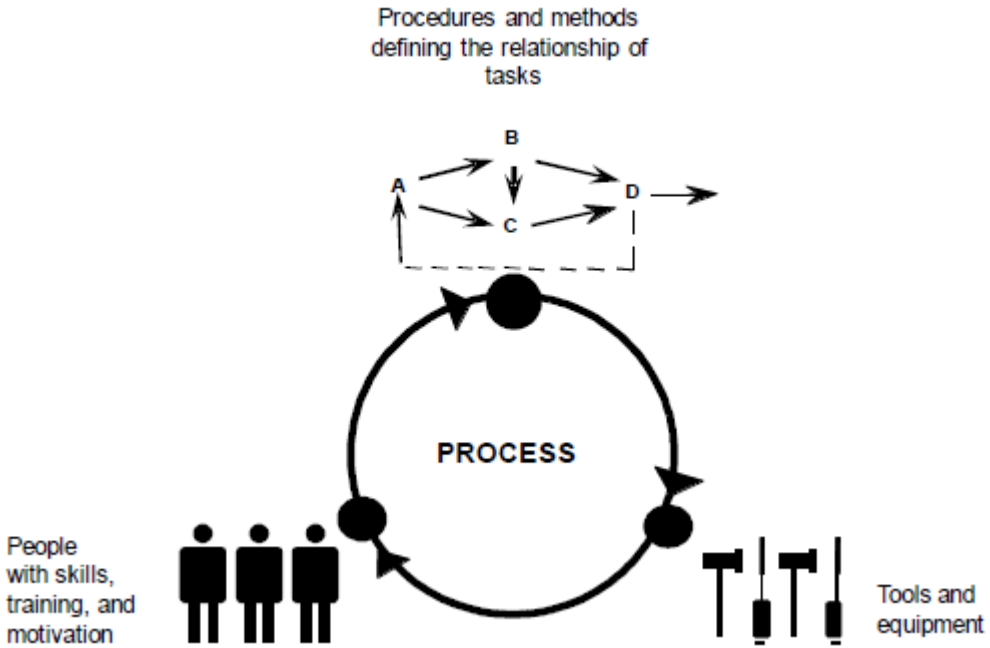
- https://sas.cmmiinstitute.com/pars/pars.aspx

Enumerate 2 Chinese Software Companies which received CMMI Maturity Level 5

- List Organization name and Organization Unit both in Chinese and English
- List Model/Constellation : CMMI-DEV, CMMI-SVC or CMMI-ACQ
- List Year of received
- List Model (Representation): Staged or Continuous
- List Appraisal Team Leader and Sponsor

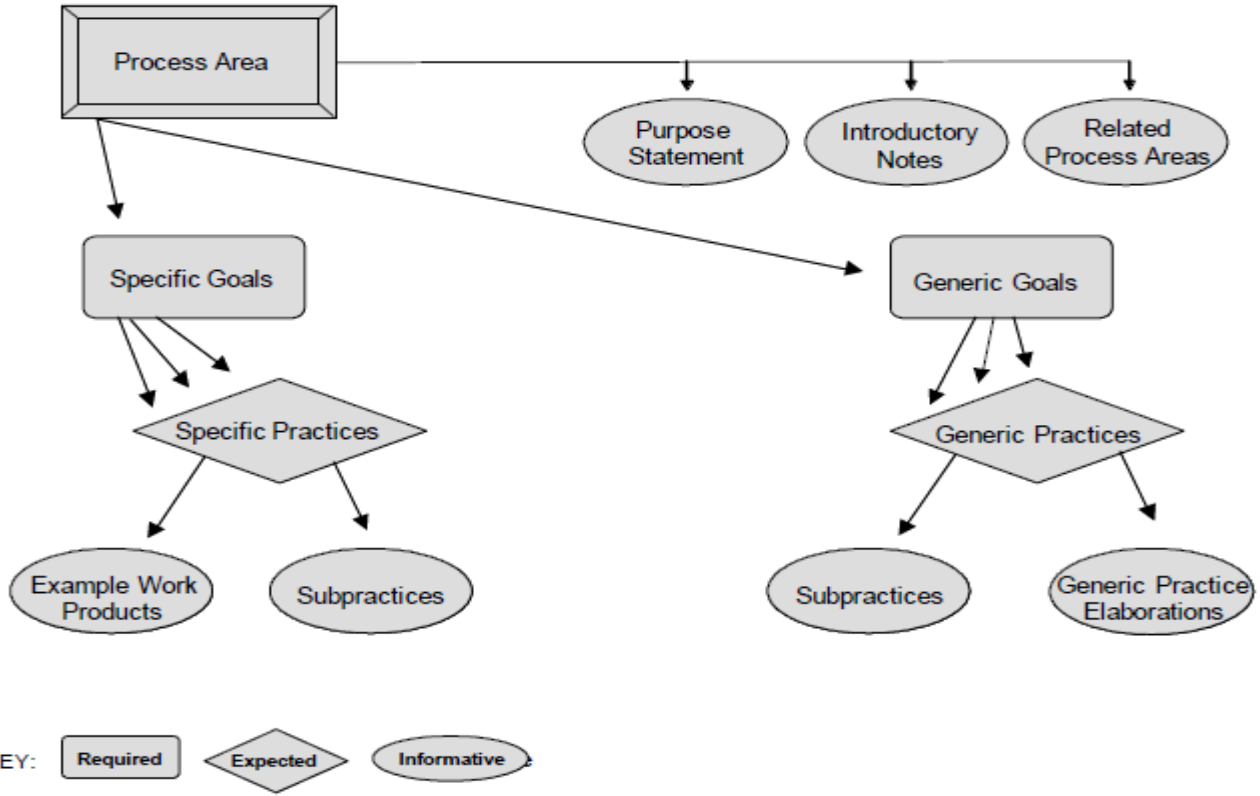
Process

A set of interrelated activities, which transform inputs into outputs, to achieve a given purpose.



Process Area

A cluster of related practices in an area that, when implemented collectively, satisfies a set of goals considered important for making improvement in that area.



Generic Goal and Practice - 1

Generic goals are called “generic” because the same goal statement applies to multiple process areas. A generic goal describes the characteristics that must be present to institutionalize processes that implement a process area. A generic goal is a required model component and is used in appraisals to determine whether a process area is satisfied.

Generic practices are called “generic” because the same practice applies to multiple process areas. The generic practices associated with a generic goal describe the activities that are considered important in achieving the generic goal and contribute to the institutionalization of the processes associated with a process area. A generic practice is an expected model component.

Generic Goal and Practice - 2

GG1 Achieve Specific Goals

- GP 1.1 Perform Specific Practices

GG 2 Institutionalize a Managed Process

- GP 2.1 Establish an 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
- 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

GG 3 Institutionalize a Defined Process

- GP 3.1 Establish a Defined Process
- GP 3.2 Collect Process Related Experiences

Specific Goal and Practice

A **specific goal** describes the unique characteristics that must be present to satisfy the process area. A specific goal is a required model component and is used in appraisals to help determine whether a process area is satisfied.

A **specific practice** is the description of an activity that is considered important in achieving the associated specific goal. The specific practices describe the activities that are expected to result in achievement of the specific goals of a process area. A specific practice is an expected model component.

Specific Goal and Practice – Project Planning

SG 1 Establish Estimates

- SP 1.1 Estimate the Scope of the Project
- SP 1.2 Establish Estimates of Work Product and Task Attributes
- SP 1.3 Define Project Lifecycle Phases
- SP 1.4 Estimate Effort and Cost

SG 2 Develop a Project Plan

- SP 2.1 Establish the Budget and Schedule
- SP 2.2 Identify Project Risks
- SP 2.3 Plan Data Management
- SP 2.4 Plan the Project's Resources
- SP 2.5 Plan Needed Knowledge and Skills
- SP 2.6 Plan Stakeholder Involvement
- SP 2.7 Establish the Project Plan

SG 3 Obtain Commitment to the Plan

- SP 3.1 Review Plans That Affect the Project
- SP 3.2 Reconcile Work and Resource Levels
- SP 3.3 Obtain Plan Commitment

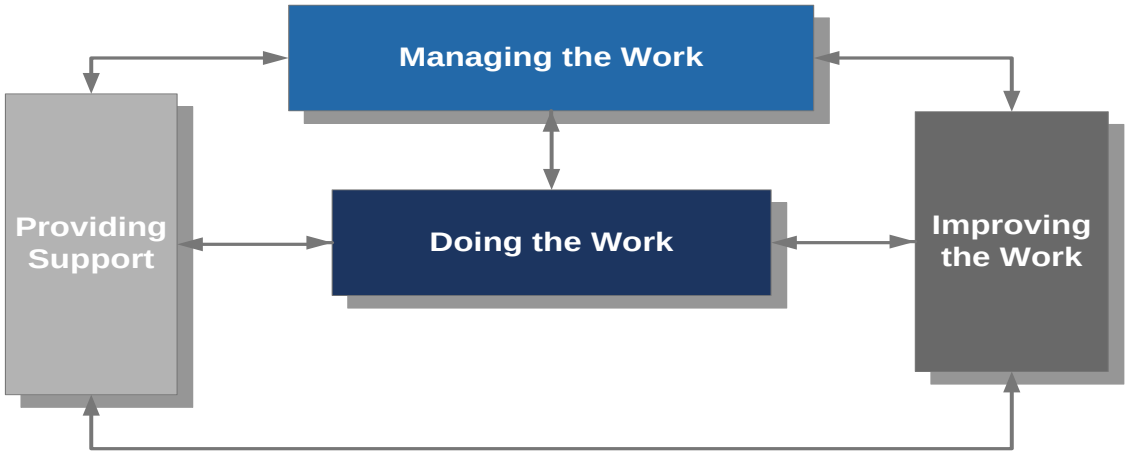
Categories of Process Areas

Project Management: Managing the work

Engineering: Doing the work

Process Management: Improving the work

Support



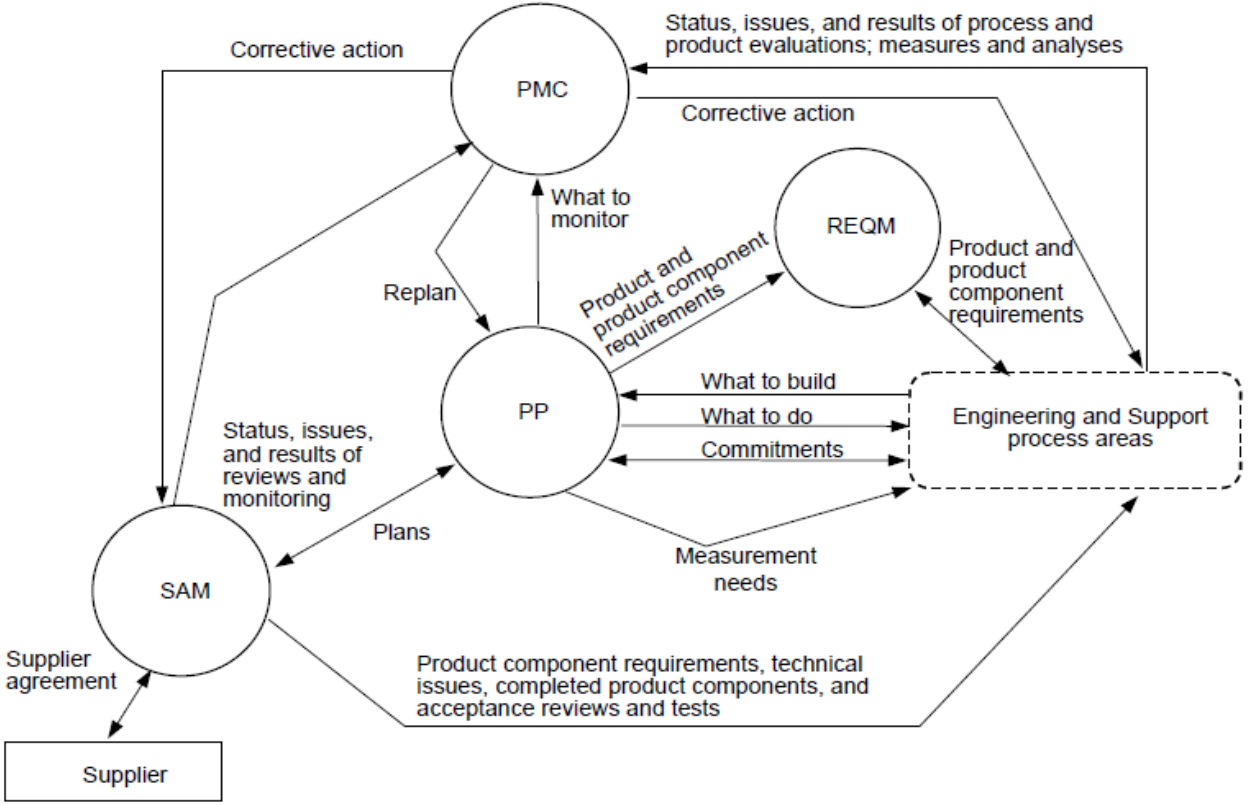
Project Management Process Areas

Project Management process areas cover the project management activities related to planning, monitoring, and controlling the project.

Include

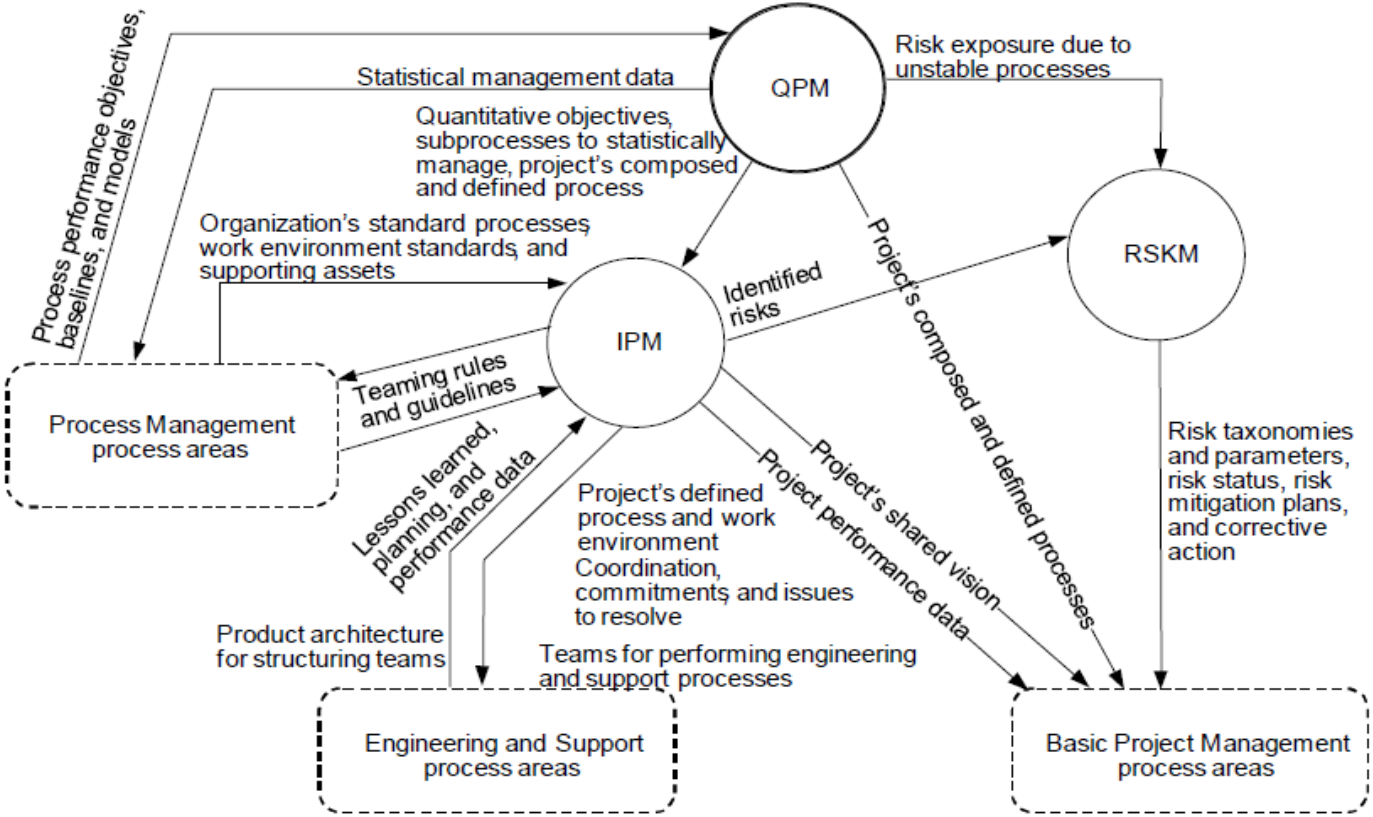
- Integrated Project Management (IPM)
- Project Management and Control (PMC)
- Project Planning (PP)
- Quantitative Project Management (QPM)
- Requirements Management (REQM)
- Risk Management (RSKM)
- Supplier Agreement Management (SAM)

Basic Project Management Process Areas



PMC = Project Monitoring and Control
PP = Project Planning
REQM = Requirements Management
SAM = Supplier Agreement Management

Advanced Project Management Process Areas



IPM= Integrated Project Management
QPM = Quantitative Project Management
RSKM = Risk Management

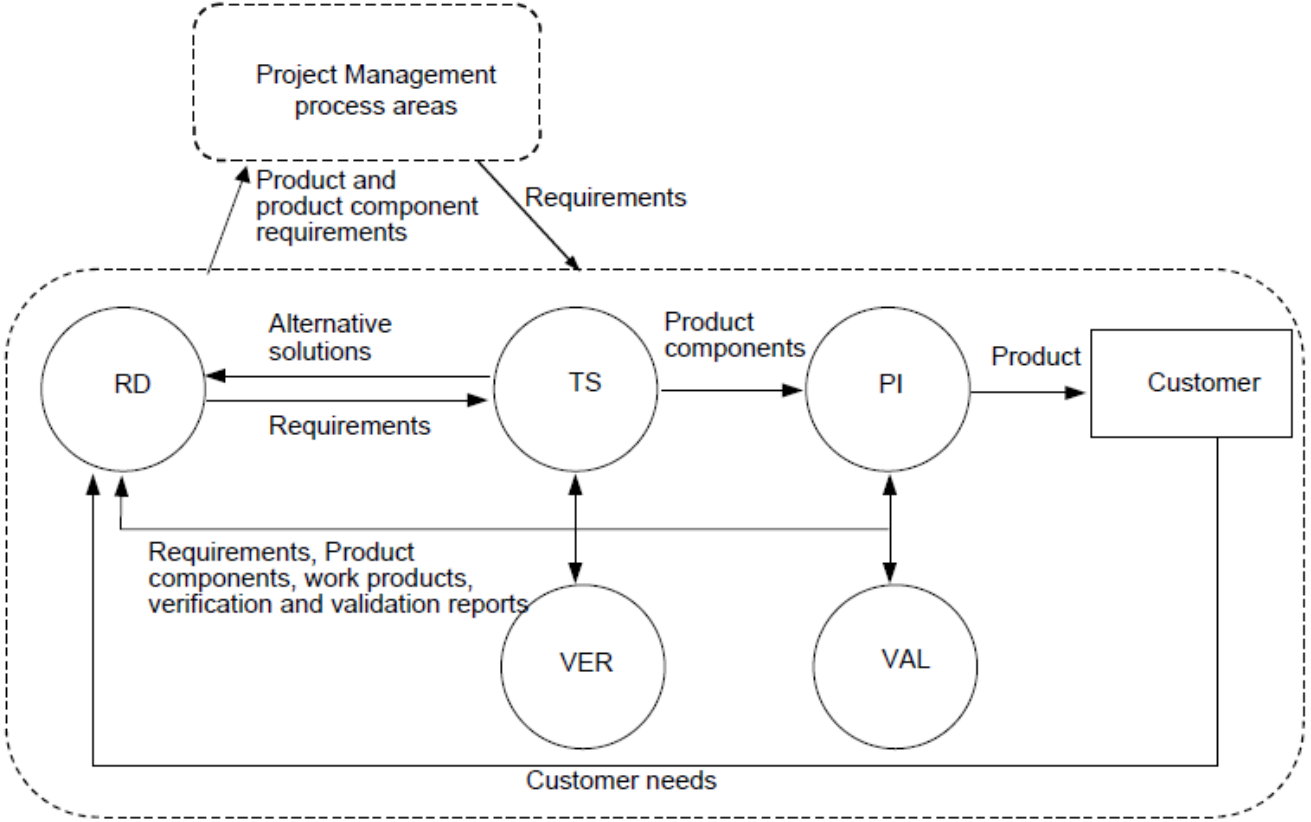
Engineering Process Areas

Addresses development and maintenance activities shared across engineering disciplines

Includes

- Requirements Development (RD)
- Technical Solution (TS)
- Product Integration (PI)
- Verification (VER)
- Validation (VAL)

Engineering Process Areas



PI = Product Integration
RD = Requirements Development
TS = Technical Solution
VAL = Validation
VER = Verification

Support Process Areas

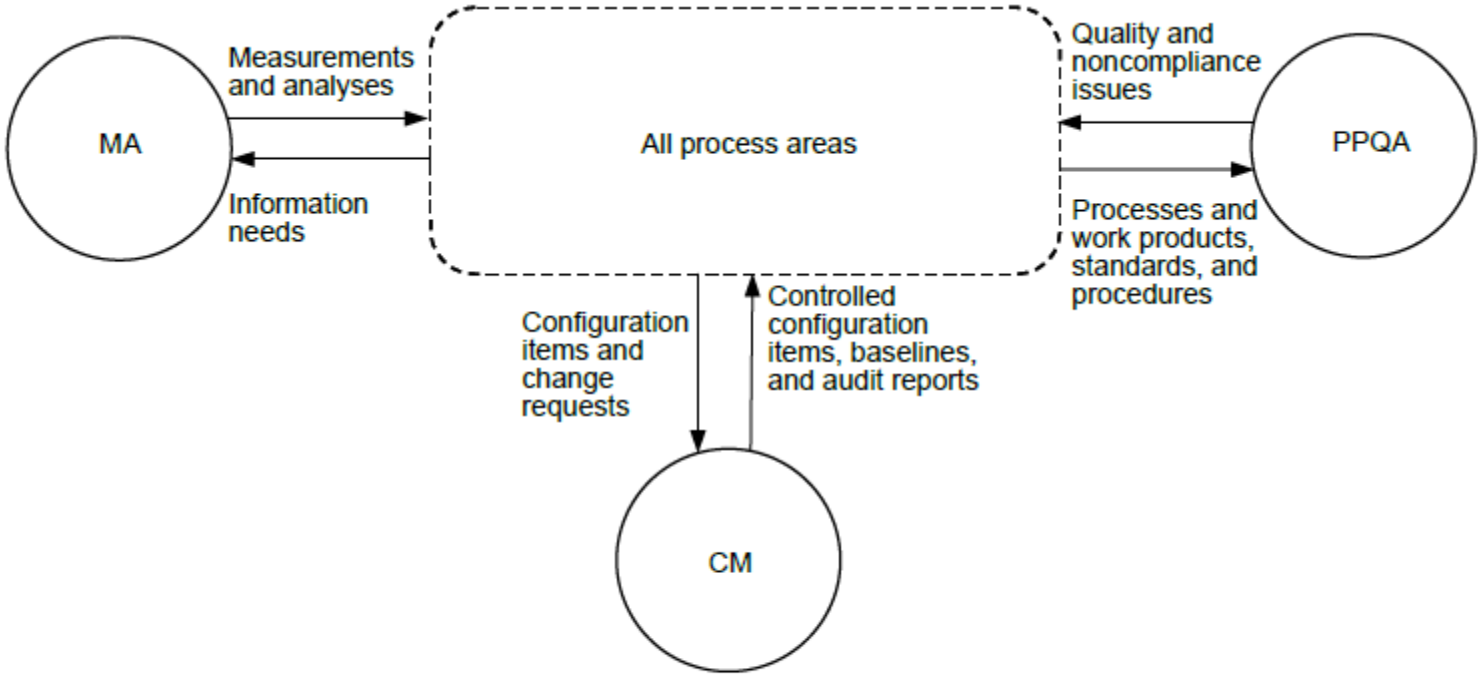
Basic Support Process Areas

- Configuration Management (CM)
- Measurement and Analysis (MA)
- Process and Product Quality Assurance (PPQA)

Advanced Support Process Areas

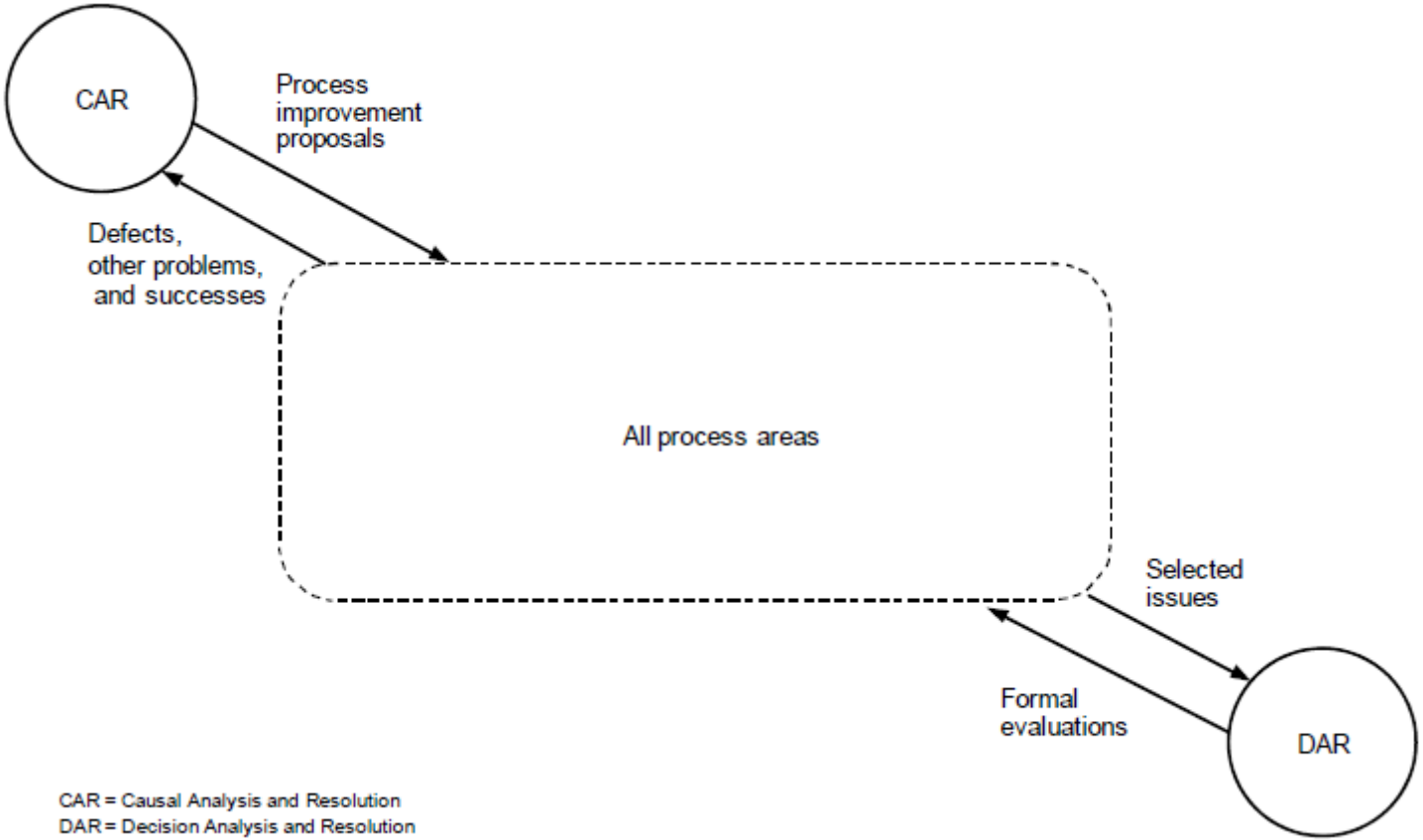
- Causal Analysis and Resolution
  - Discover root causes of defects & inefficiencies
  - Modify process to prevent reoccurrence
- Decision Analysis and Resolution
  - Determine issues subject to formal evaluation

Basic Support Process Area



CM = Configuration Management
MA = Measurement and Analysis
PPQA = Process and Product Quality Assurance

Advanced Support Process Areas



Process Management

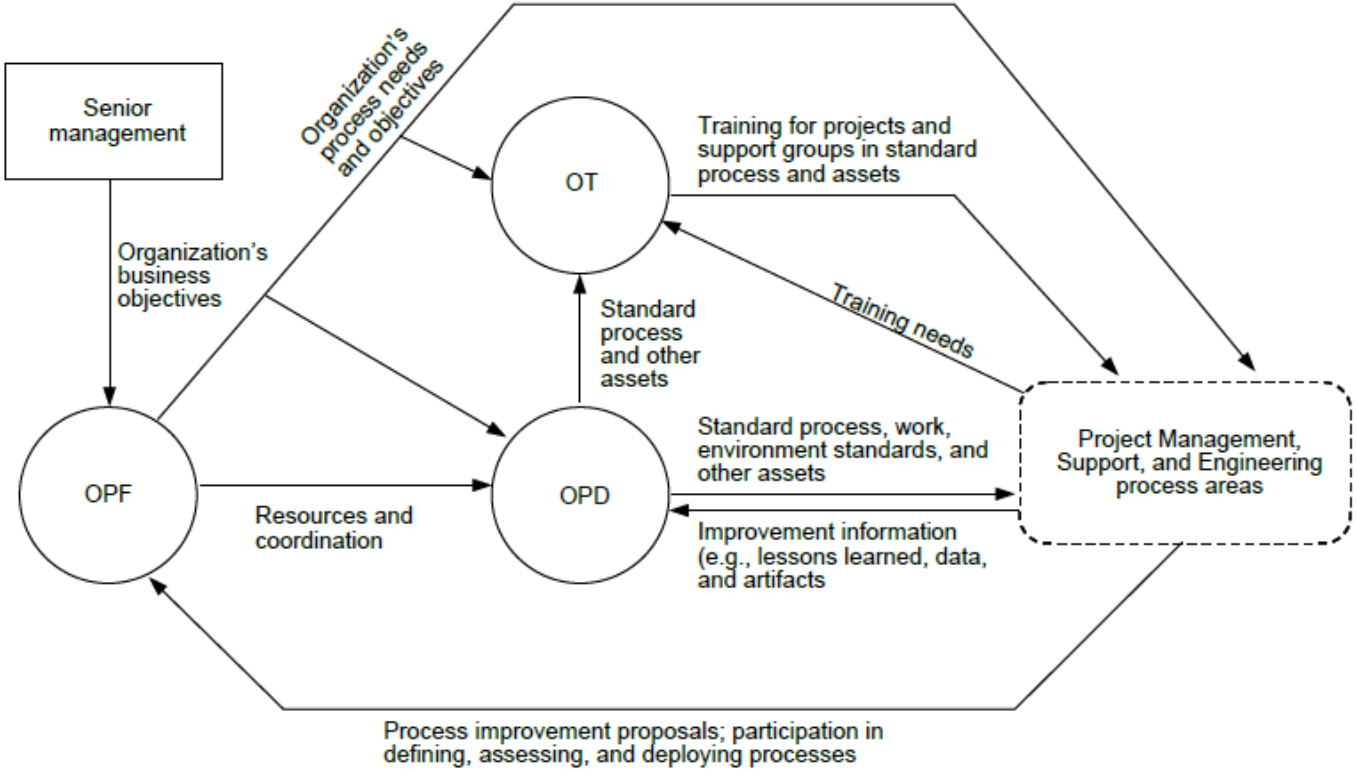
Basic Process Management Areas

- Organizational Process Definition (OPD)
- Organizational Process Focus (OPF)
- Organizational Training (OT)

Advanced Process Management

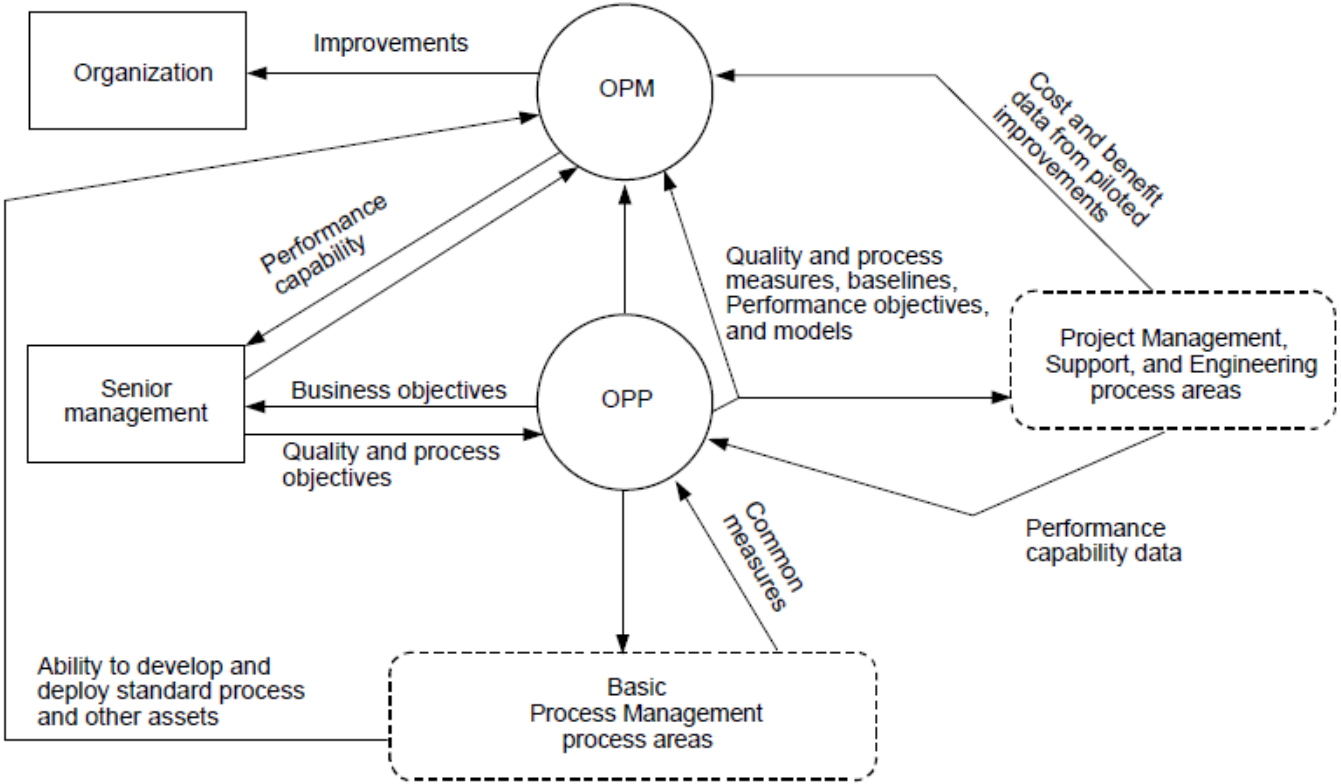
- Organizational Performance Management (OPM)
- Organizational Process Performance (OPP)

Basic Process Management Areas



OPD = Organizational Process Definition
OPF = Organizational Process Focus
OT = Organizational Training

Advanced Process Management



OPM = Organizational Performance Management
OPP = Organizational Process Performance

Process and Product Quality Assurance

- A Support Process Area at Maturity Level 2

Why Quality Assurance?

Focusing on processes

Makes best practices standard

Helps train new people faster

Helps spread the abilities of experts

Eliminates reinventing the wheel

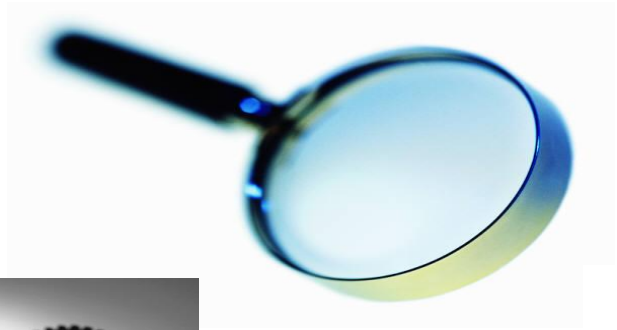
Emphasizes lessons learned

« The quality of a (software) system is largely governed by the quality of the process used to develop and maintain it ». Watts S. Humphrey

Quality Assurance vs Quality Control

	Quality Control	Quality Assurance
Scope	Products	Processes and Products
Objective	Detect problems in the work products	Verify adherence to processes
When	Once the product is made available	While (or even before) the product is being developed

Quality Assurance (QA) in the CMMI®



The purpose of PPQA(Process and Product Quality Assurance) is to provide staff and management with objective insight into process and associated work products.

Goals of PPQA in the CMMI®

Quality assurance activities are planned.

Objectively evaluating performed processes and work products against applicable process descriptions, standards, and procedures

Identifying and documenting noncompliance issues

Providing feedback to project staff and managers on the results of quality assurance activities

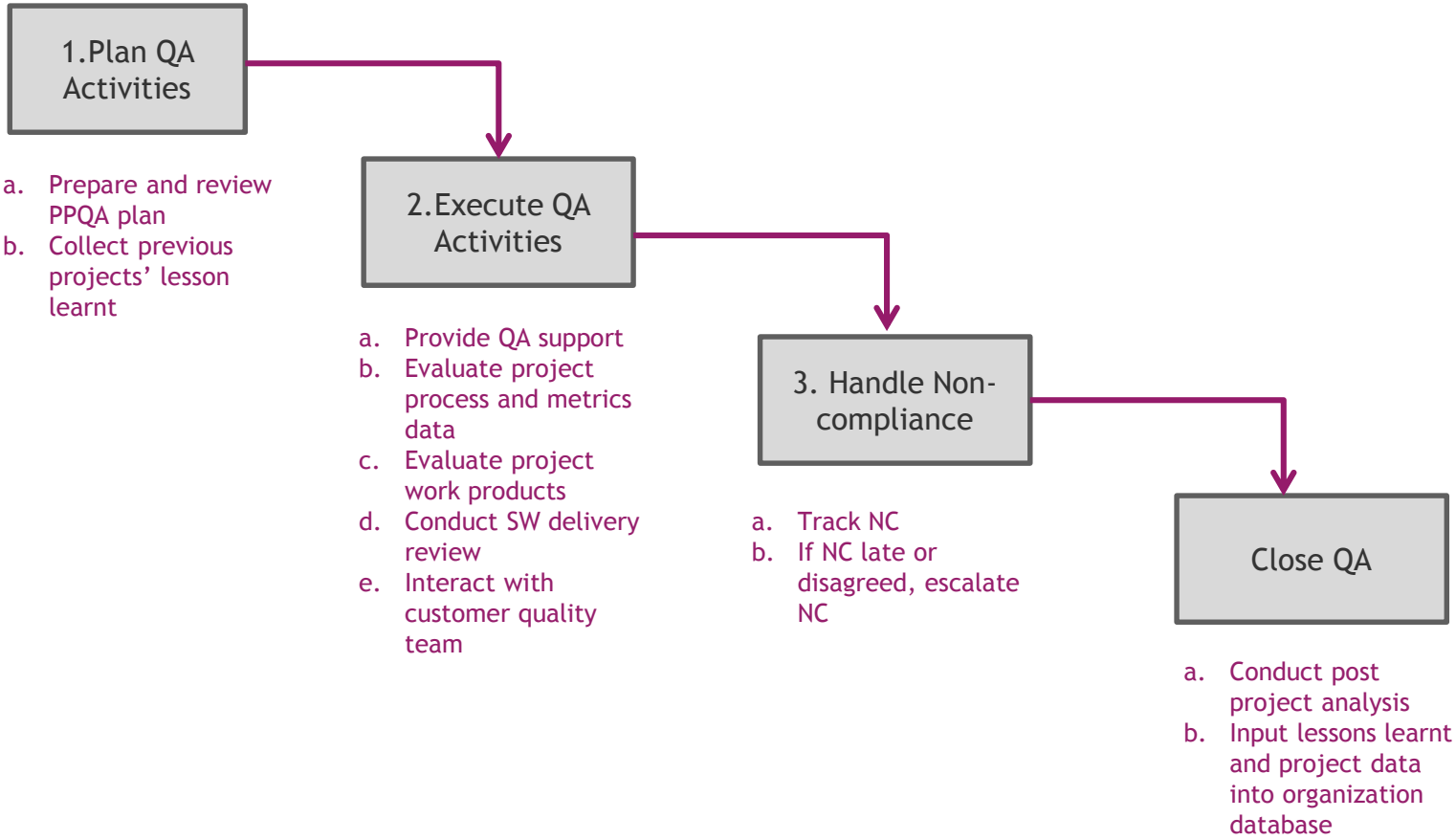
Ensuring that noncompliance issues are addressed

SG and SP –Process and Product Quality Assurance

Specific Goals and Specific Practices

- SG 1 Objectively Evaluate Processes and Work Products
  - SP 1.1 Objectively Evaluate Processes
  - SP 1.2 Objectively Evaluate Work Products
- SG 2 Provide Objective Insight
  - SP 2.1 Communicate and Resolve Noncompliance Issues
  - SP 2.2 Establish Records

Typical PPQA Process for Project



PPQA Plan Example

Identify Customer Expectations

List applicable processes and document deviations and/or adaptations

Identify Key Work Products

Identify Work Product Reviews

Identify Process Reviews and audits

List actions to support to the Project Team

Evaluate Effort estimate

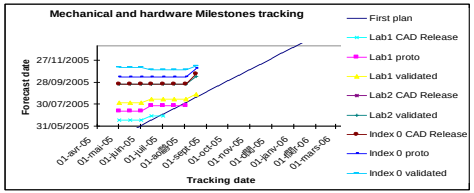
Recap Action Items

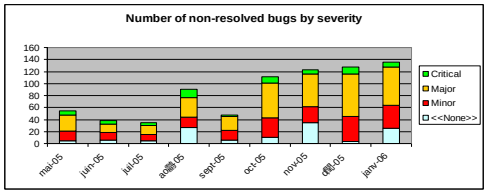
Project Metrics

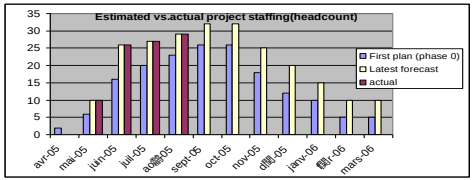
Review Project Data

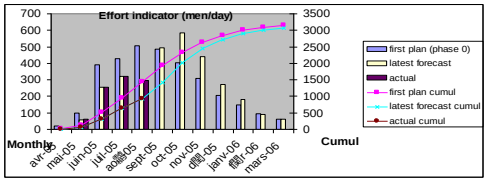
In-depth analysis of project data and trend in terms of:

- Milestones (delay and variance)
- Effort (difference and variance)
- Problem Reports (distribution versus time)
- Change Requests (distribution versus time)
- Risks and Issues









Non-Compliances

Handle Non-Compliances

Noncompliance issues are problems identified in evaluations that reflect a lack of adherence to applicable standards, process descriptions, or procedures.

There are 3 possible ways for resolving a non-compliance issue:

- Adjust the work product to satisfy the process
- Adjust the process to make it usable
- Make an executive decision to not satisfy the process

SW Release Review

SW Release Criteria

- Defect free (0 critical, 0 major)
- Release documents (Release note, test report, etc.) ready

SW Release review

- Code under version control
- Completion of Code review and fixing of review defects
- Completion of testing activities
  - Test Report available
  - Test coverage confirmed
  - Bugs found during the testing to be tracked/logged
  - software shall be validated under a “two pass validation” scheme, at the validation test level:
    - "First pass" is normal feature validation test; it is a full coverage test of the scope of the release. The defects identified for correction are then fixed
    - "Second pass" is after defects fixes have been made in the impacted area, focused on both : Defect correction testing (including regression) in impacted areas to ensure the defect has disappeared; Non-regression testing on other features/area interfacing with impacted areas

SW Delivery checklist (Discussion)

- According to the released SW characteristics, design the release checklist

Requirement Management

- A Project Management Process Area at Maturity Level 2

Specific Goal and Practice – Requirement Mgt.

SG 1 Manage Requirements

- SP 1.1 Understand Requirements
- SP 1.2 Obtain Commitment to Requirements
- SP 1.3 Manage Requirements Changes
- SP 1.4 Maintain Bidirectional Traceability of Requirements
- SP 1.5 Ensure Alignment Between Project Work and Requirements

What is a Requirement ?

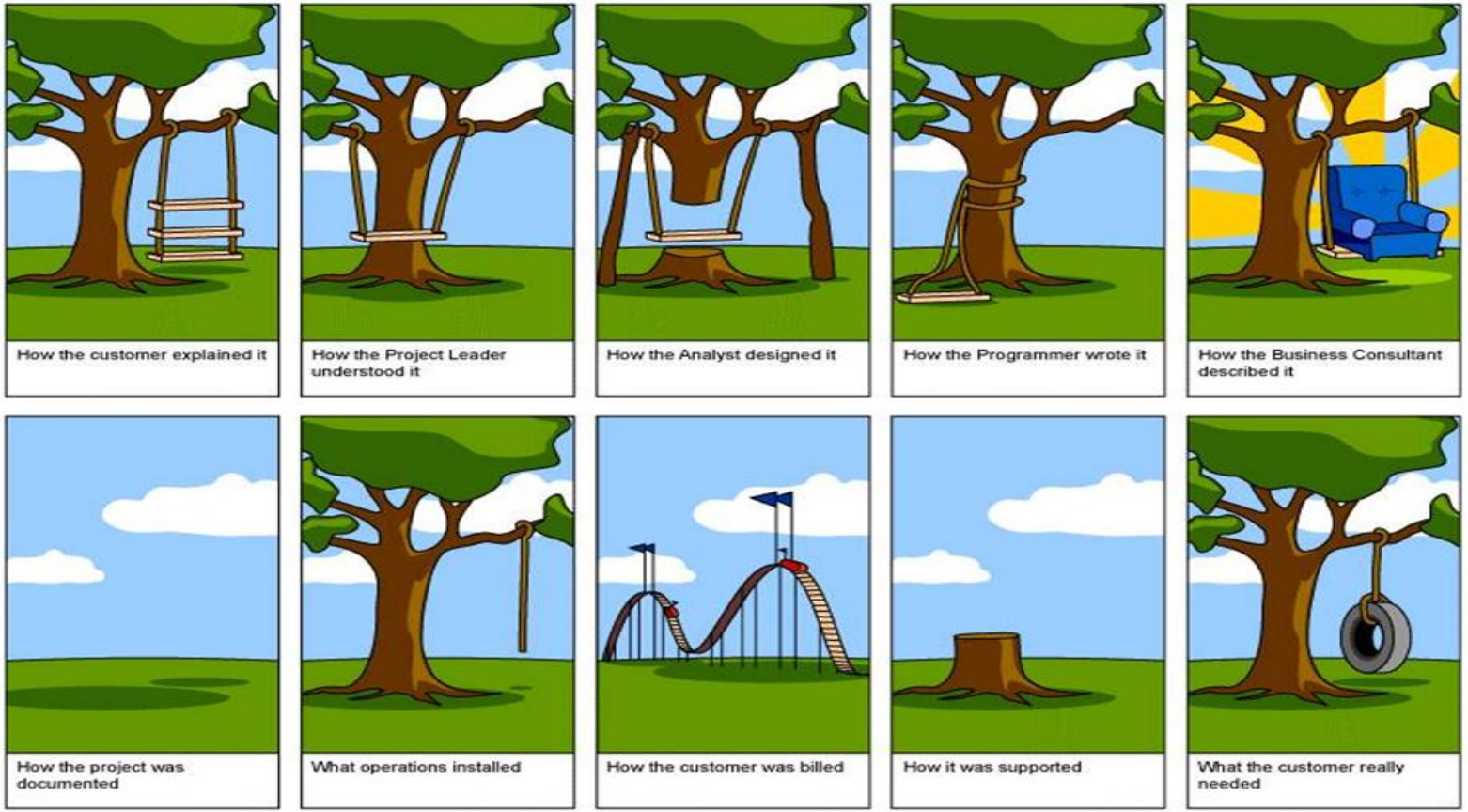
A requirement is a description of a characteristic to be incorporated in the future product

- System requirement - A condition or capability that must be met or possessed by a system or system component to satisfy a condition or capability needed by a user to solve a problem. [IEEE-STD-610]
- Software requirement - A condition or capability that must be met by software needed by a user to solve a problem or achieve an objective. [IEEE-STD-610]

Requirements are

- functional or non-functional (e.g. performance or design constraints)
- identified by the customer or the internal teams
- identified at the beginning of the project or later

Example


How the customer explained it
How the Project Leader understood it
How the Analyst designed it
How the Programmer wrote it
How the Business Consultant described it
How the project was documented
What operations installed
How the customer was billed
How it was supported
What the customer really needed

Understand Requirement

Develop an common understanding with the requirements providers.

Obtain commitment to requirements from project participants.

Requirement Change Management

Once approved by the customer and baselined, requirements shall be placed under Change Management

- Examples of documents which need be updated and re-baselined
  - Requirement Specification
  - Traceability Matrix
  - Design document
  - Test documents

Evaluate the impact of requirement changes from the standpoint of relevant stakeholders

All Change Requests (CR) to requirements shall be managed under Change Management

Requirement Traceability Matrix

This matrix is used to check requirements consistency across the project life cycle.

This matrix is placed under Configuration Management.

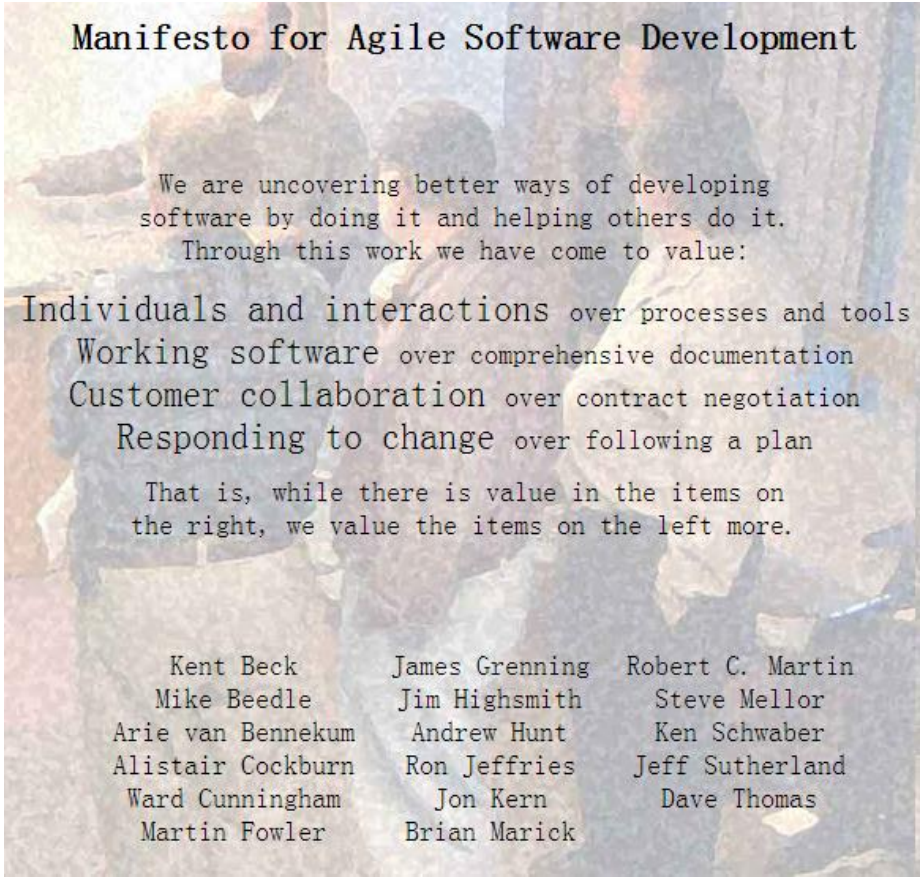
Requirements should be regrouped to be manageable.

Requirement Traceability Matrix Example

Req ID	Features	Requirements	Characteristics	Verify/Review Owner	Where from	Design Reference	Testing Reference	Change Status
SW-01	Security	Bootloader Security Requirement	Req 2.05. The STB shall meet all the requirements specified in the STBD-T-045 Bootloader Security Requirements document.	*Ding Hui*	[Ref-1]	Design-Chap-2.1	*Bootloader-01*	Initial
SW-02	Security	Bootloader component & OTP	Req 2.06. Any bootloader component that is not updateable shall be stored in a permanent locked area in the Flash.	*Ding Hui*	[Ref-1]	Design-Chap-2.1	*Bootloader-02*	Initial
SW-03	Security	Bootloader sign and approval	Req 2.08. Only a bootloader signed by NDS and approved by NDS for use in the field shall be used for production units.	*Ding Hui*	[Ref-1]	Design-Chap-2.1	*Bootloader-03*	Initial
SW-04	HDD Formating	HDD Formatting Feature in MRS	4/14:EPG can format if it is atleast once formatted by factory software. Comments from Panch	Wang Xiao	[Ref-1]	Design-Chap-2.2	*HDD-01*	Deleted
SW-05	HDD Formating	HDD Formatting Feature in Factory Test SW	Factory Test SW need support HDD format for EPG usage	Wang Xiao	[Ref-1]	Design-Chap-2.2	*HDD-02*	Initial
SW-06	Audio	Audio MUTE by HW	Support Audio MUTE/UnMute control by HW PIO.	Sun Lei	[Ref-1]	Design-Chap-2.3	Audio-03	Initial
SW-07	USB2.0	USB2.0 SW	Support USB 2.0(MS) in Uboot and Kernel for Factory Test SW, HW Test SW and MTBF Test SW, and Over current test, 2 ports	Chang Hao	[Ref-1]	Design-Chap-2.4	USB-01	Initial
SW-08	USB3.0	USB3.0 SW	Support USB 3.0(MS) in Kernel for Factory Test SW, HW Test SW and MTBF Test SW, 1 port	Chang Hao	[Ref-1]	Design-Chap-2.4	USB-02	Modification

International quality management model: Agile

Starting point: Agile Manifesto (Feb 2001)



http://agilemanifesto.org/

http://www.agilealliance.org/

Twelve Principles of Agile Software

敏捷宣言遵循的原则

我们遵循以下原则：

我们最重要的目标，是通过持续不断地
及早交付有价值的软件使客户满意。

欣然面对需求变化，即使在开发后期也一样。
为了客户的竞争优势，敏捷过程掌控变化。

经常地交付可工作的软件，
相隔几星期或一两个月，倾向于采取较短的周期。

业务人员和开发人员必须相互合作，
项目中的每一天都不例外。

激发个体的斗志，以他们为核心搭建项目。
提供所需的环境和支援，辅以信任，从而达成目标。

不论团队内外，传递信息效果最好效率也最高的方式是
面对面的交谈。

可工作的软件是进度的首要度量标准。

敏捷过程倡导可持续开发。
责任人、开发人员和用户要能够共同维持其步调稳定延续。

坚持不懈地追求技术卓越和良好设计，敏捷能力由此增强。

以简洁为本，它是极力减少不必要工作量的艺术。

最好的架构、需求和设计出自自组织团队。

团队定期地反思如何能提高成效，
并依此调整自身的举止表现。

Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.

Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.

Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.

Business people and developers must work together daily throughout the project.

Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.

The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.

Working software is the primary measure of progress.

Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.

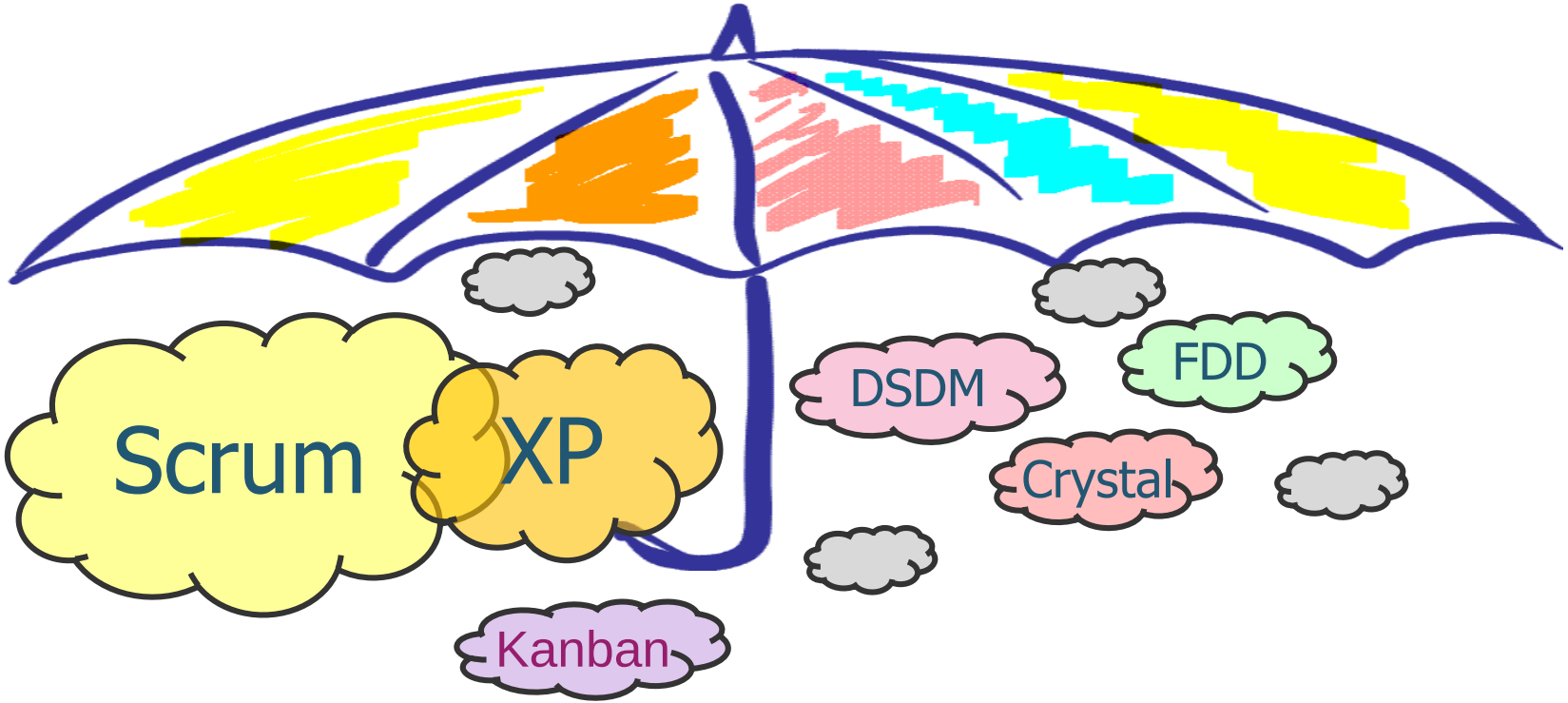
Continuous attention to technical excellence and good design enhances agility.

Simplicity--the art of maximizing the amount of work not done--is essential.

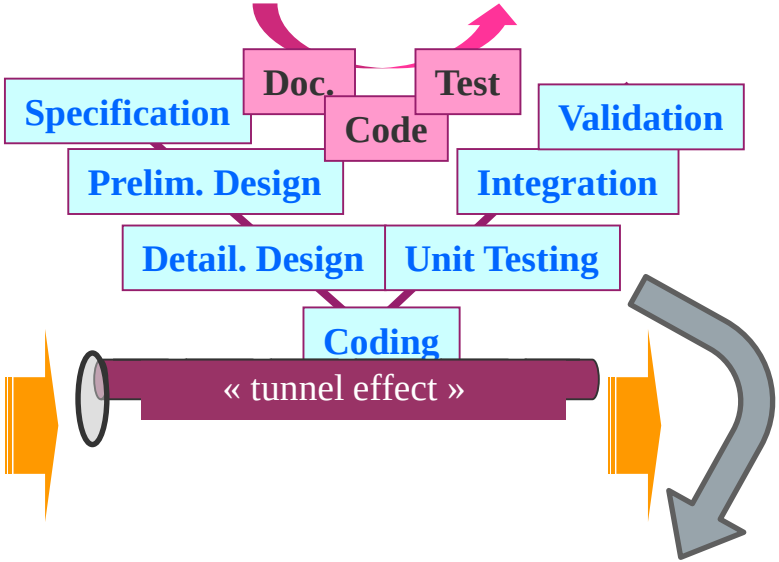
The best architectures, requirements, and designs emerge from self-organizing teams.

At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

Agile is an ”umbrella”of frameworks/methods



Agile - The Big Picture

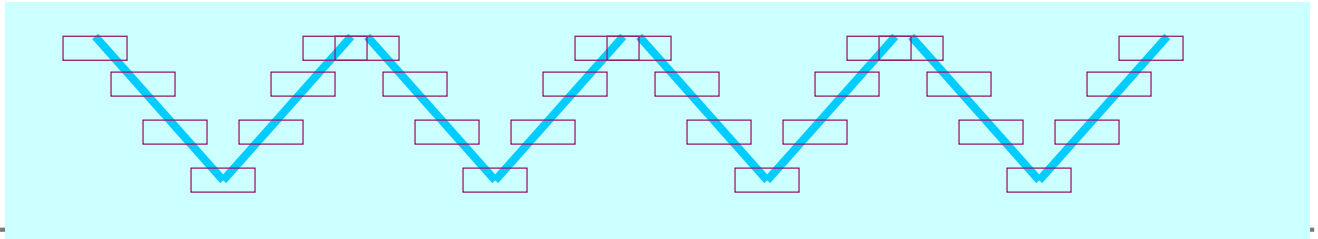


Agile is an iterative process. The iterations are time-boxed that last from 2-4 weeks.

Each iteration involves a full software development cycle from planning, requirements analysis, design, coding, unit testing and acceptance testing.

At the end of each iteration a working product (subset of the overall requirements) is demonstrated to the stakeholders, this minimizes risk and helps adapt to changes quickly if required.

Iterative and incremental life cycle



Agile vs Waterfall method

- Waterfall more suited for :
  - Requirements do not change often
  - Widely distributed development teams
  - Big bang release strategy

- Agile more suited for :
  - Adapt to requirements, change faster
  - Focus on user adoption, experience and feedback
  - Energy is spent on development cycles not documentation

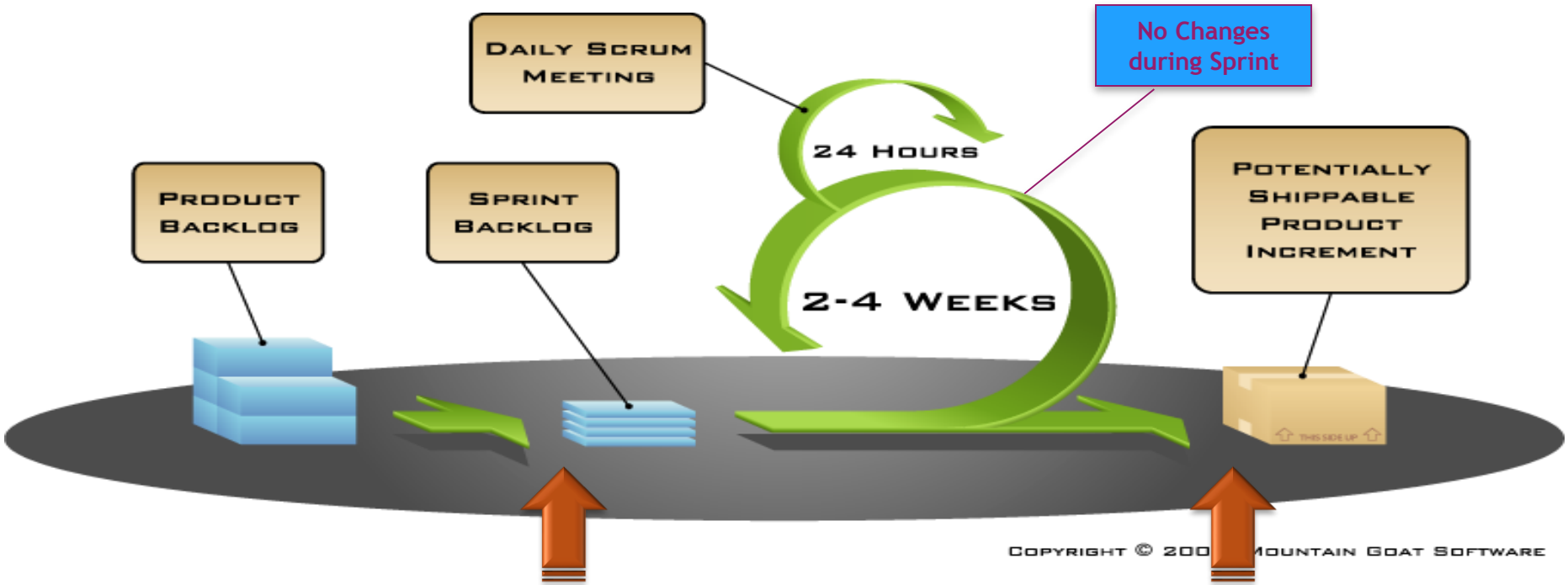
An Agile Project Management Method

Scrum / Rugby

- The Game
- Team
- Individual Players
- Focused
- ONE common goal:

Move the ball FORWARD

Scrum Framework– How Does it work?



Sprint Planning meeting

Sprint Review
Sprint Retrospective

Scrum Framework

Roles

- Product owner
- ScrumMaster
- Team

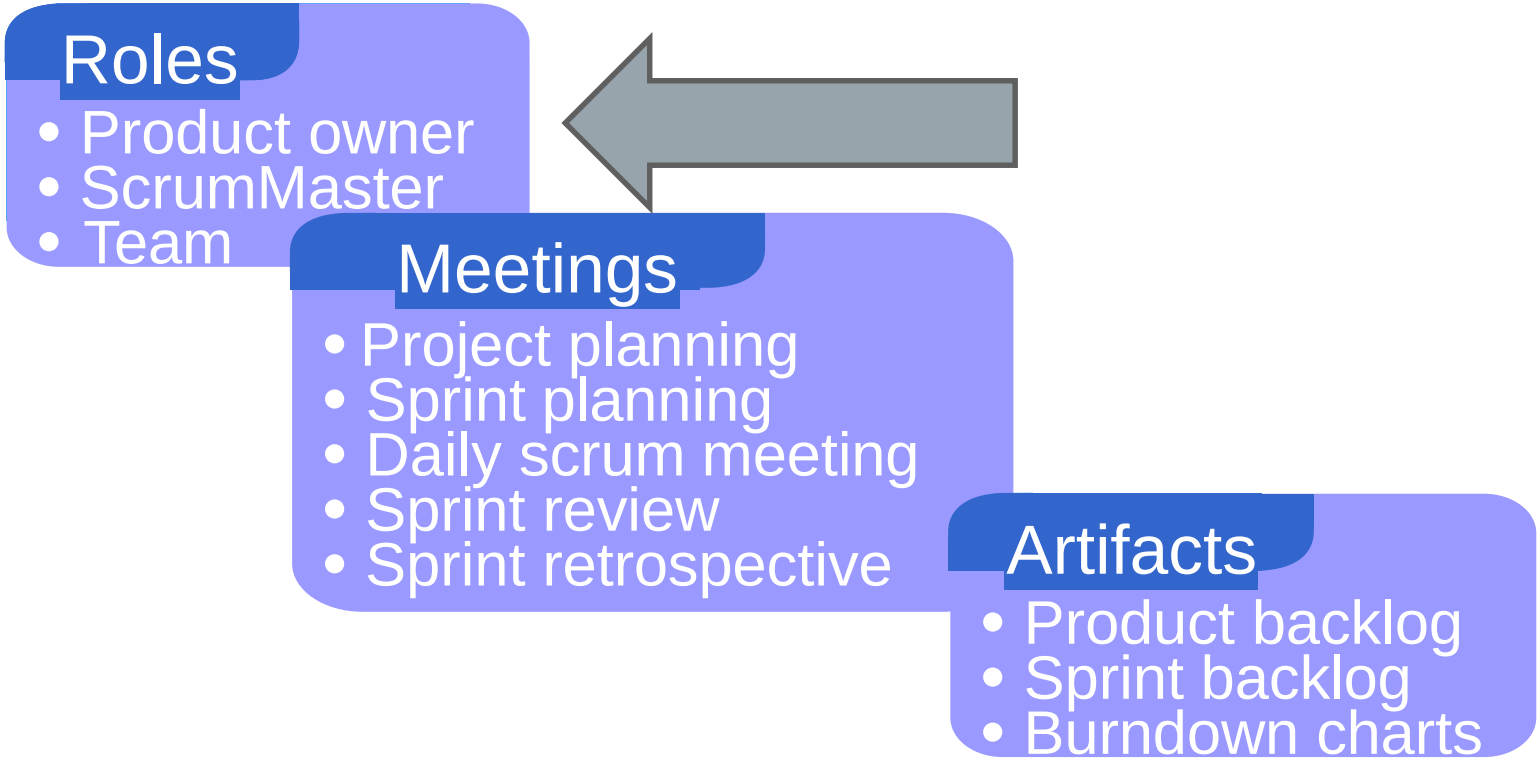
Meetings

- Project planning
- Sprint planning
- Daily scrum meeting
- Sprint review
- Sprint retrospective

Artifacts

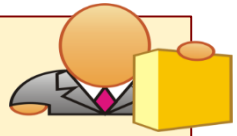
- Product backlog
- Sprint backlog
- Burndown charts

Scrum Framework - Roles



Roles in Scrum

Product Owner (Business)



- Represent the “voice” of the customer
- Define the features of the product
- Decide on release date and content
- Be responsible for the profitability of the product (ROI)
- Prioritized features/backlog
- Adjust features and priority every iteration, as needed
- Accept or reject work results

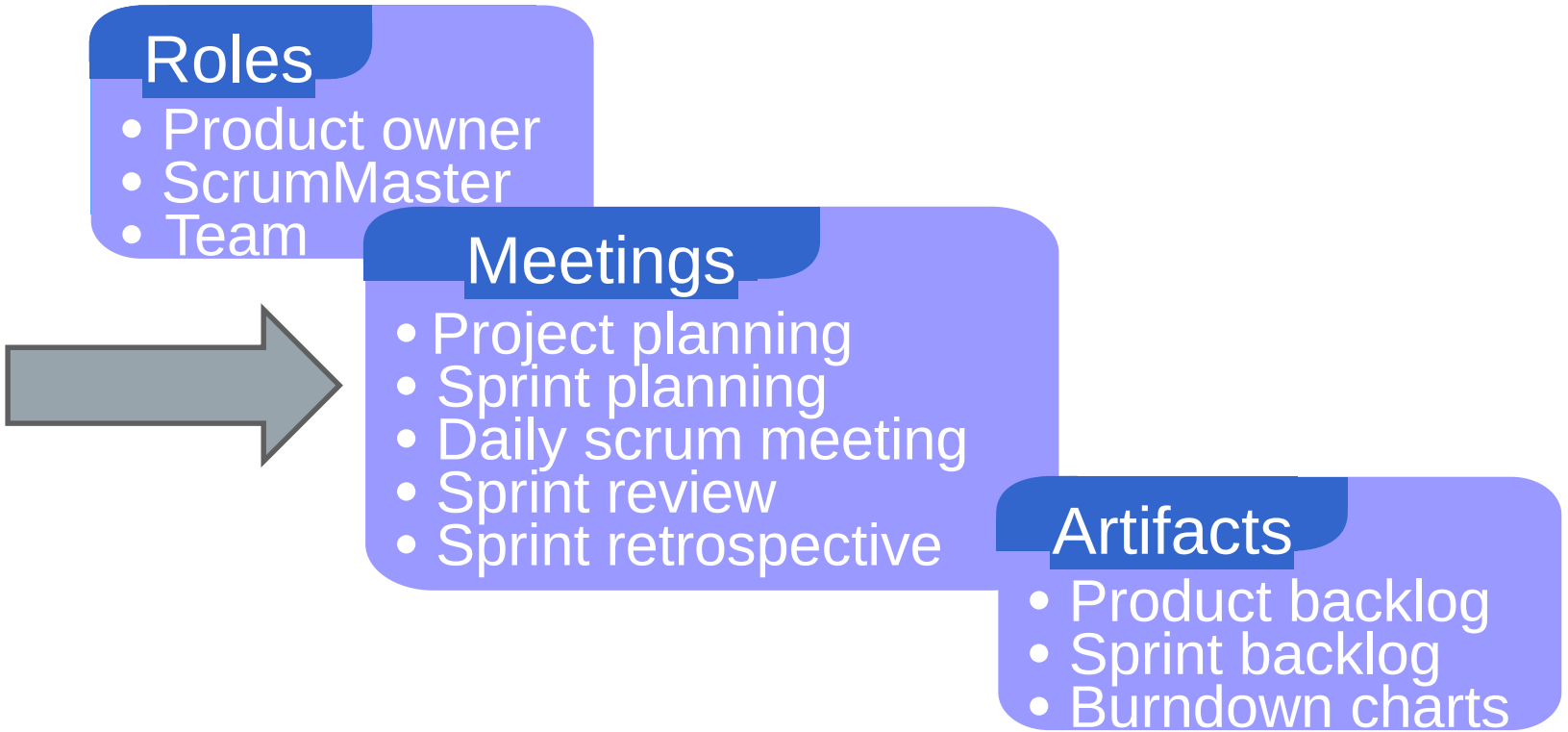
The Team

- Typically 5-9 people
- Cross-functional:
  - Programmers, testers, user experience designers, etc.
  - Members should be full-time
  - May be exceptions (e.g., database administrator)
- Responsible for work Breakdown and task estimation at Sprint Planning Meeting
- **Teams are self-organizing**

Scrum Master (Part of the team)

- Ensure that the process is understood and applied
- Responsible for enacting Scrum values and practices
- Removes impediments
- Ensure that the team is fully functional and productive
- Enable close cooperation across all roles and functions
- Shield the team from external interferences
- Organize and conduct Daily scrum, Sprint Retrospective, Sprint Planning and Sprint Review meetings

Scrum Framework - Meetings



User stories definition

As a user, I want to search for my customers by their first and last names so that I can manage my service.

Estimation – A Key Skill

Estimating how long tasks will take is often difficult

A key value of Agile vs Waterfall comes from the constant striving to improve estimates. Software development being predictable (i.e. having low variance in estimation) has clear business value.

People are best at what they do regularly

History tells us we are not so good at estimating – but what we are good at is comparison

Estimating Product Backlog Work

Use Planning Poker to estimate the work for each backlog item in Ideal Days (or Ideal Weeks) or Story Points. Discuss the size and complexity of each item before estimating (record this information in the backlog notes).

- Ideal Days: One person-work day dedicated entirely to the item - no other work or interruptions.
- Story Points: An agreed-upon unit of effort or size based on one of the shorter items on the backlog. This item should be one which everyone is comfortable estimating.

Once all the estimates are provided, use a formula to convert the estimations to Real Days. Take into account historical information. For example:

- Real Day = Ideal Day * 80%
- 1 Story Point = 3 Real Days

Planning Poker

This activity is coordinated by the ScrumMaster. Each participant gets a deck of estimation cards. Participants agree on the effort unit(Man*Day: one person-work day).

- The Product Owner describes one Product Backlog item at a time and answers any questions the team might have.
- The team may discuss the development approach, key tasks, and estimated size and complexity of the item.
- Each participant privately selects a card representing his/her estimate.
- When everybody is ready with an estimate, all cards are presented simultaneously.
- In the (very likely) event that the estimates differ, the high and low estimators defend their estimates.
- The group briefly debates the arguments, discusses any assumptions, etc.
- A new round of estimation is made. As soon as the estimates appear to be converging on a particular number, the facilitator should ask if everyone is OK with that number. Continue until agreement is reached (generally 2 or 3 rounds).
- The ScrumMaster notes the estimate, along with any assumptions, and the group continues with the next Product Backlog item.

Chickens And Pigs

A pig and a chicken decide to open a restaurant. They discuss the name. The chicken says, "How about 'Ham and Eggs'"? The pig says, "No, I'd be committed, but you would only be involved."

In Scrum, the owner, ScrumMaster and team are the "pigs"; i.e. they are committed. Other stakeholders, including the Product Owner, may only be involved.

At the Daily Scrum, anyone may attend, but only the "pigs" may speak!

Scrum – Key Meetings:Project Planning/Backlog Grooming

- The Product Owner and Team create an initial Product Backlog
- It is often done before each Sprint Planning Meeting
- Re-assessing the relative priority of user stories
- Removing user stories that no longer appear relevant
- Creating new user stories in response to newly discovered needs
- Assigning estimates to user stories which have yet to receive one
- Correcting estimates in light of newly discovered information
- Splitting user stories which are high priority but too coarse grained to fit in an upcoming iteration

Scrum – Key Meetings : Sprint Planning

Part 1

- Product Owner and Team
- **What** should we build in the next sprint?
- Product Owner presents backlog
- Team asks questions
- Collaboration to get common understanding
- Define or Review “Definition of Done”
- Establish Sprint Goals
- Revise estimates
- Team picks from prioritized backlog taking items to fill to team capacity

Part 2

- Scrum Team
- **How** are we going to meet Sprint goals?
- Break sprint backlog into tasks (Estimates no greater than16hrs)
- Team plan how it will work together to deliver
- Review of “Definition of Done”
- Identify Obstacles/Risks
- Identify Dependencies
- Identify conditions of satisfaction (tests, inspection..)
- Make needed adjustments
- Review Sprint Backlog with PO

Scrum – Key Meetings: The Daily Scrum

Scrum Master Coordinates

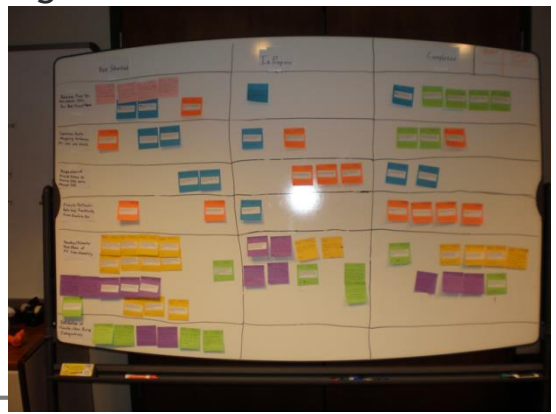
- same time daily, on-time, same place
- 15-minutes (timeboxed), Stand-up

Scrum Team Members

- Team updates the amount of work remaining in the Sprint Backlog before each Scrum for burndown.
- Team synchronizes by answering the questions, One person speaks; others listen
  - What have you done on this project since the last Scrum meeting?
  - What do you plan to do between now and the next Scrum meeting?
  - What obstacles are in your way?
- Not a problem-solving or design session

Scrum Master reports on obstacles

Chickens invited to listen and observe



Scrum – Key Meetings : Sprint Review

Sprint Review

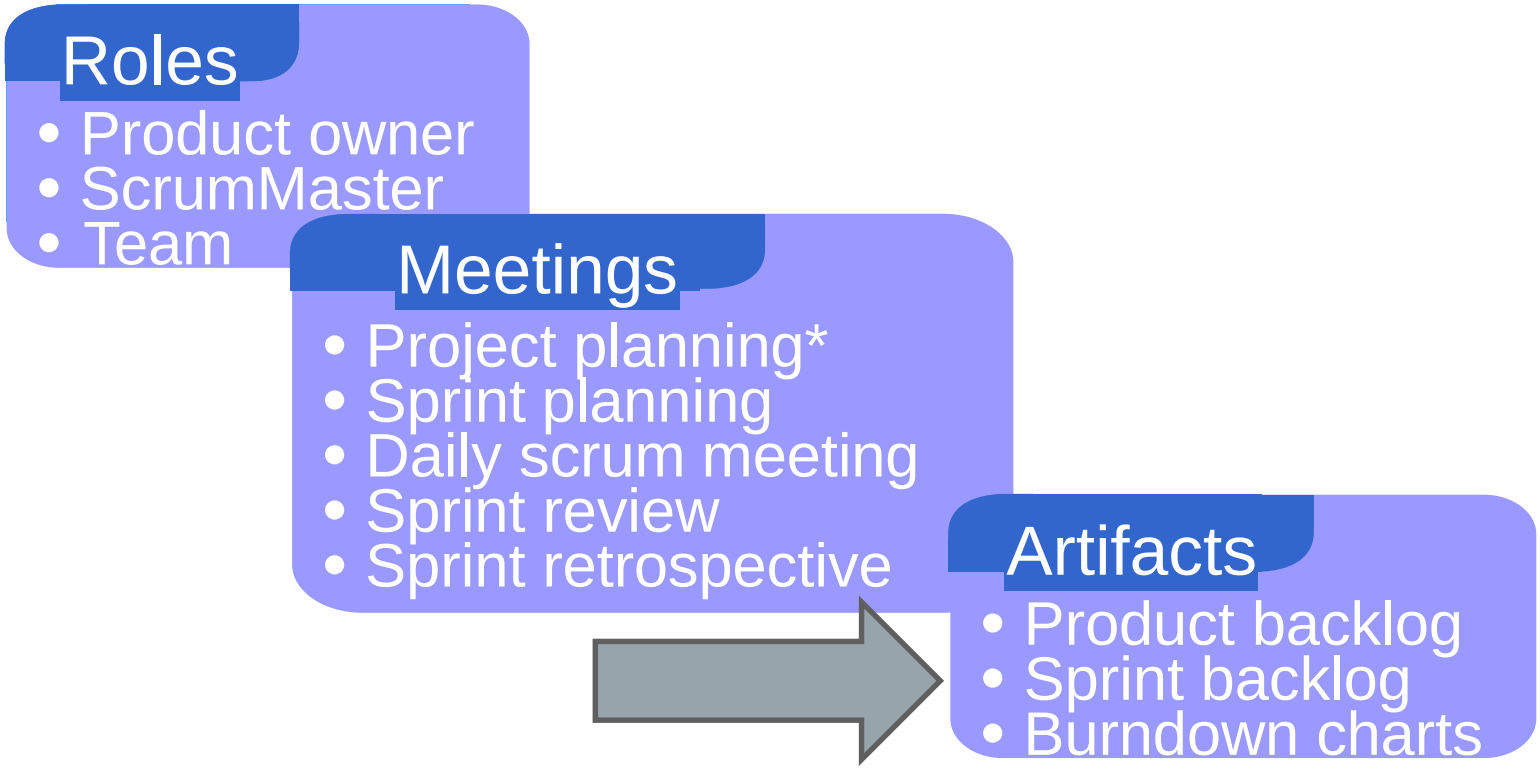
- Purpose is to get feedback and approval
- Team demonstration of what was “done”
- Scrum Team, Product Owner, Stakeholders, Users, others
- Discuss Obstacles / Learning
- Sprint Burndown
- Release Burndown
- Product Owner collects feedback
- Engage your audience

Scrum – Key Meetings : Sprint Retrospective

Sprint Retrospective

- What went well during the last Sprint?
- What could be improved in the next Sprint?
- What adjustments should we make to improve performance?
- Identify 3 specific changes to implement and monitor in next sprint
- Health check for the TEAM
- Celebrate success

Scrum Framework - Artifacts



Product Backlog

Product Backlog: “Requirement” for a system

- High level description of all issues, requirements, bug fixes, performance etc. that require work on a high level.
- Ideally expressed such that each item has value to the users or customers.
- All may add items to the Product Backlog.
- Prioritized (controlled) by the Product Owner, and reprioritized at the start of each sprint.

Product Backlog item	Estimate
Allow a guest to make a reservation	3
As a guest, I want to cancel a reservation.	5
As a guest, I want to change the dates of a reservation.	3
As a hotel employee, I can run RevPAR reports (revenue-per-available-room)	8
Improve exception handling	8
...	30
...	50

Release Backlog

Release Backlog: Subset of the product backlog that you are targeting for that release

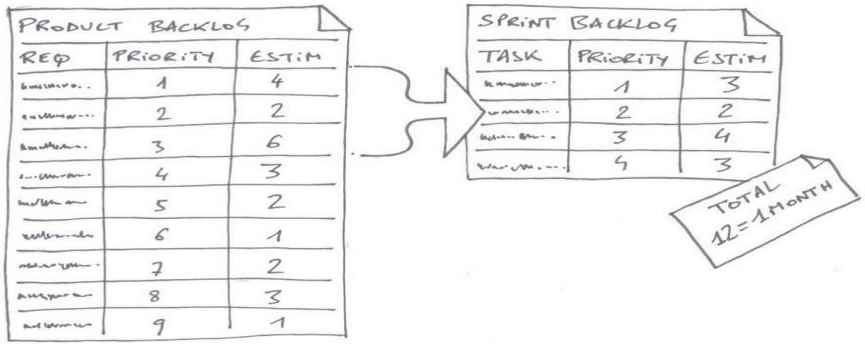
- The release backlog contains more details about the requirement and low level estimate which are usually estimated by the team performing the work.

Sprint Backlog

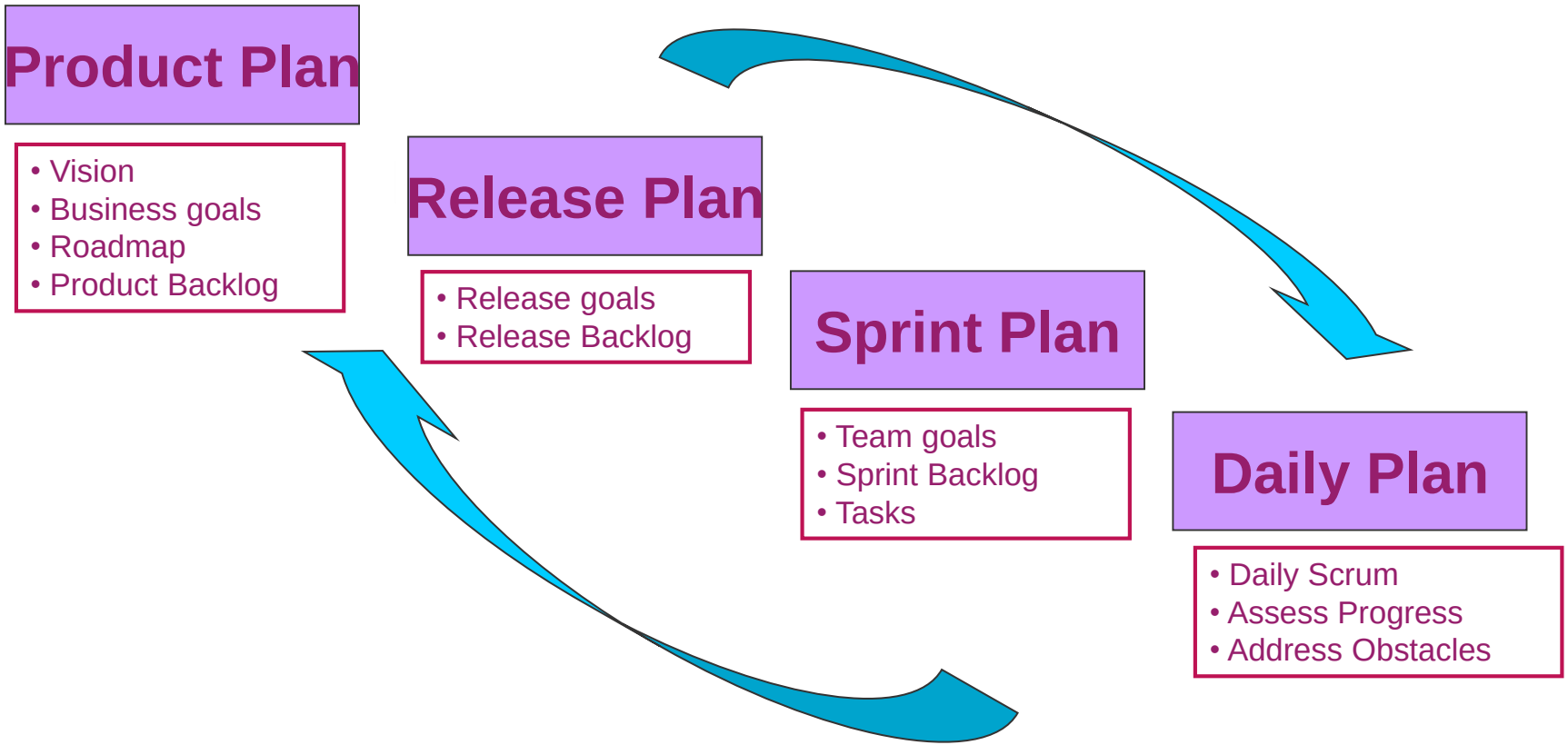
Sprint Backlog

- Subset of the Release Backlog
- Created and controlled by the team.
- Work the team has committed to do.
- “Well-understood” Stories and Tasks.
- When possible, subdivided into tasks less than 16Hours in duration

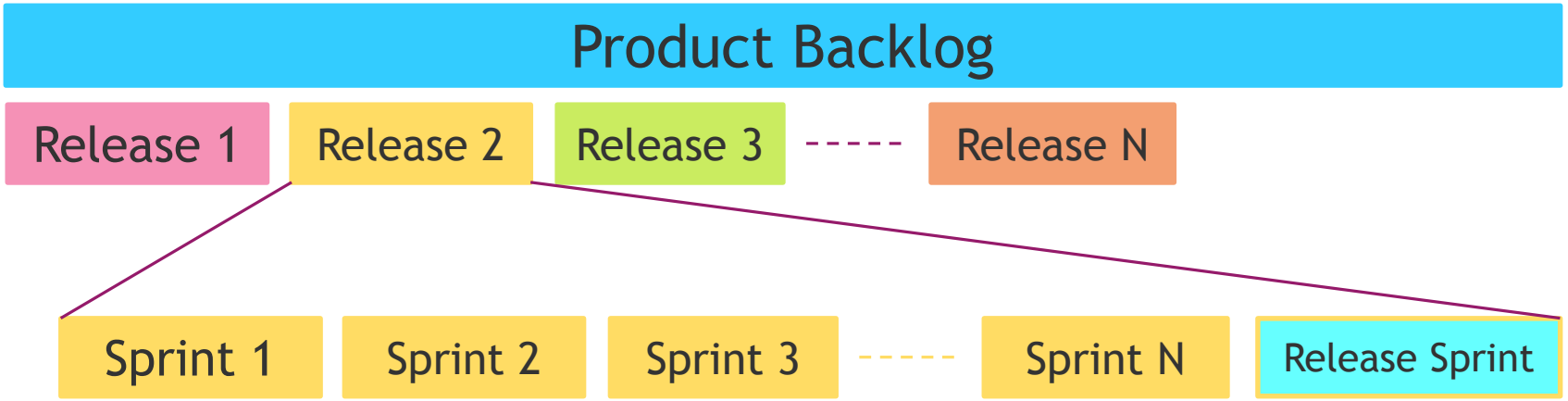
Sprint Backlog Tasks	Mon	Tues	Wed	Thur	Fri
Code the user interface	3	4			
Code the middle tier	4	8	4		
Test the middle tier		2	4	8	
Write online help	8				
Write the foo class	4	4	4	4	
Add error logging			8	4	



Frequent Planning with Scrum



Release Planning



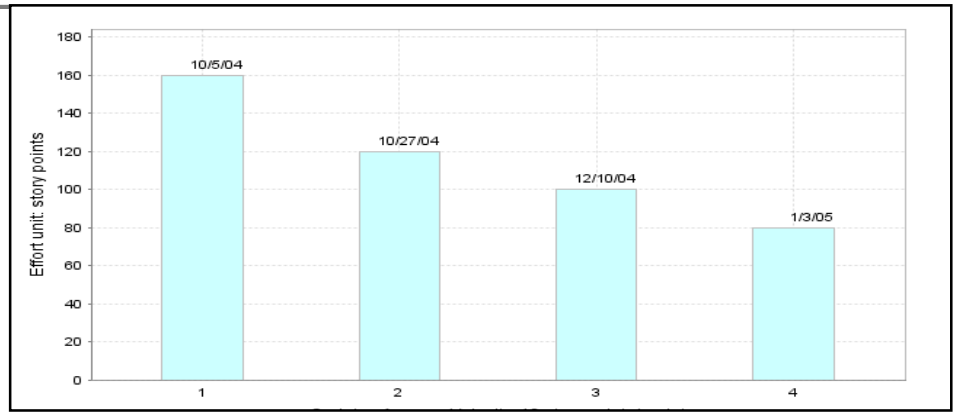
Definition of Done

US	■ Source Code and test script submitted to source control tool ■ CI compilation successfully, CI and regression test pass ■ Acceptance criteria met, test case associated with the US pass. ■ Test cases reviewed at least by one another person, test case execution result updated ■ All critical and major issues detected in code review closed. ■ Static code analysis should be done where necessary, no new Critical and major issue reported ■ Any OpenSource change must be noted, ■ API, design, architecture documentation updated if applicable ■ Any public API change must be documented
Iteration	■ Last CI compilation successfully, CI and regression test pass ■ No Critical bug open in case of customer delivery unless with agreement of PO ■ Issue detected by Opensource check closed in case of customer delivery ■ Sprint test plan executed; Regression, stress, abnormal conditions and system tests should be included when appropriate ■ Enough USs fully groomed for next sprint ■ Sprint review/demo completed ■ Sprint retrospective meeting done and action points documented
Field Release	■ Release Notes published ■ API, guide documented ■ The software is base-lined and released ■ Release build stored in security space (GIT, CC, space with backup etc.) ■ Release test plan executed, customer scenario testing taken into account if relevant ■ Zero Major and Critical bugs open ■ Security check completed, security risk managed and closed

Burndown Charts

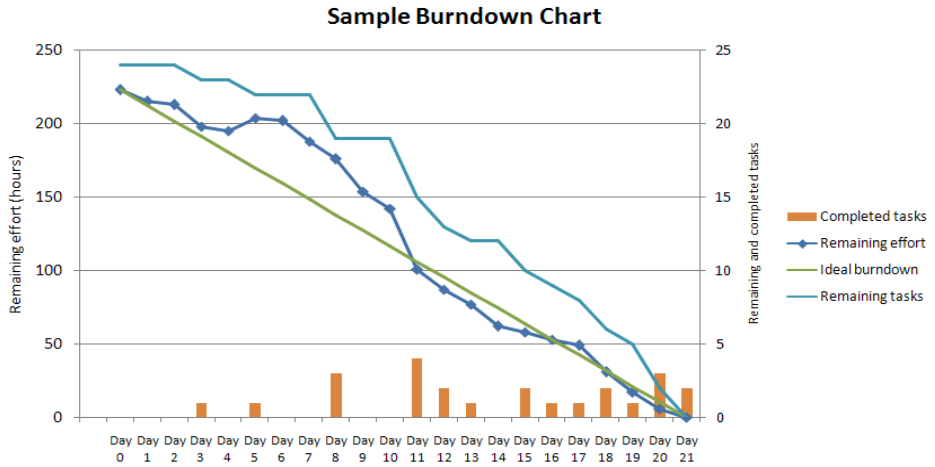
Product Burndown

- Shows the number of Product Backlog items remaining at the end of each Sprint.
- Updated after each Sprint Review.

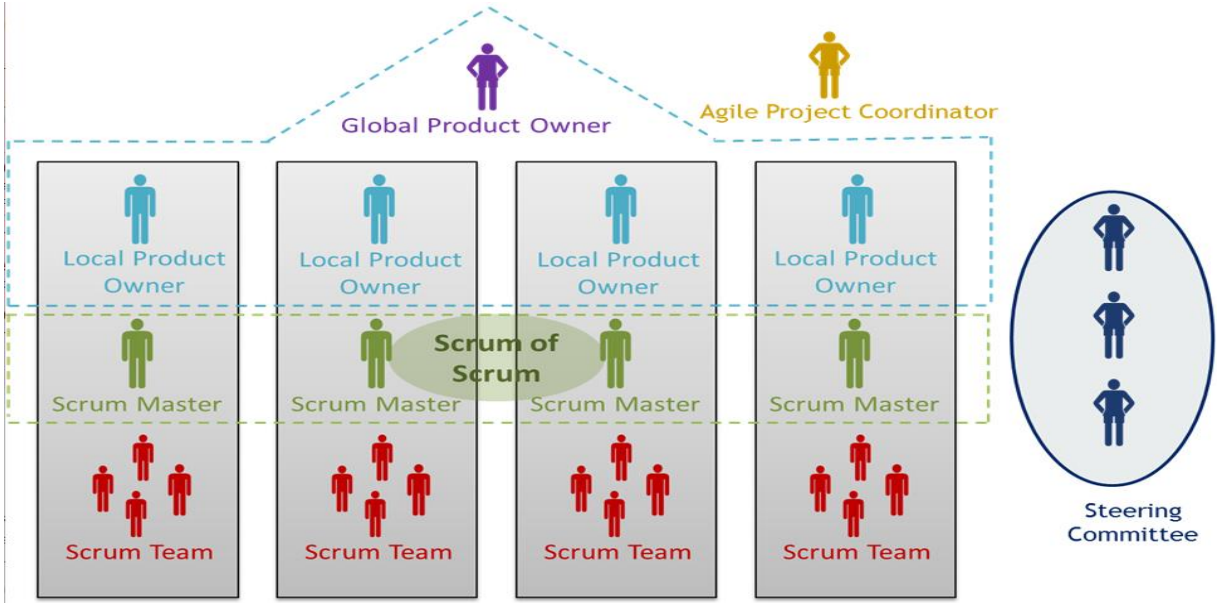


Sprint Burndown

- Shows the amount of work remaining during the Sprint.
- Updated after each Scrum.



Scaled Scrum

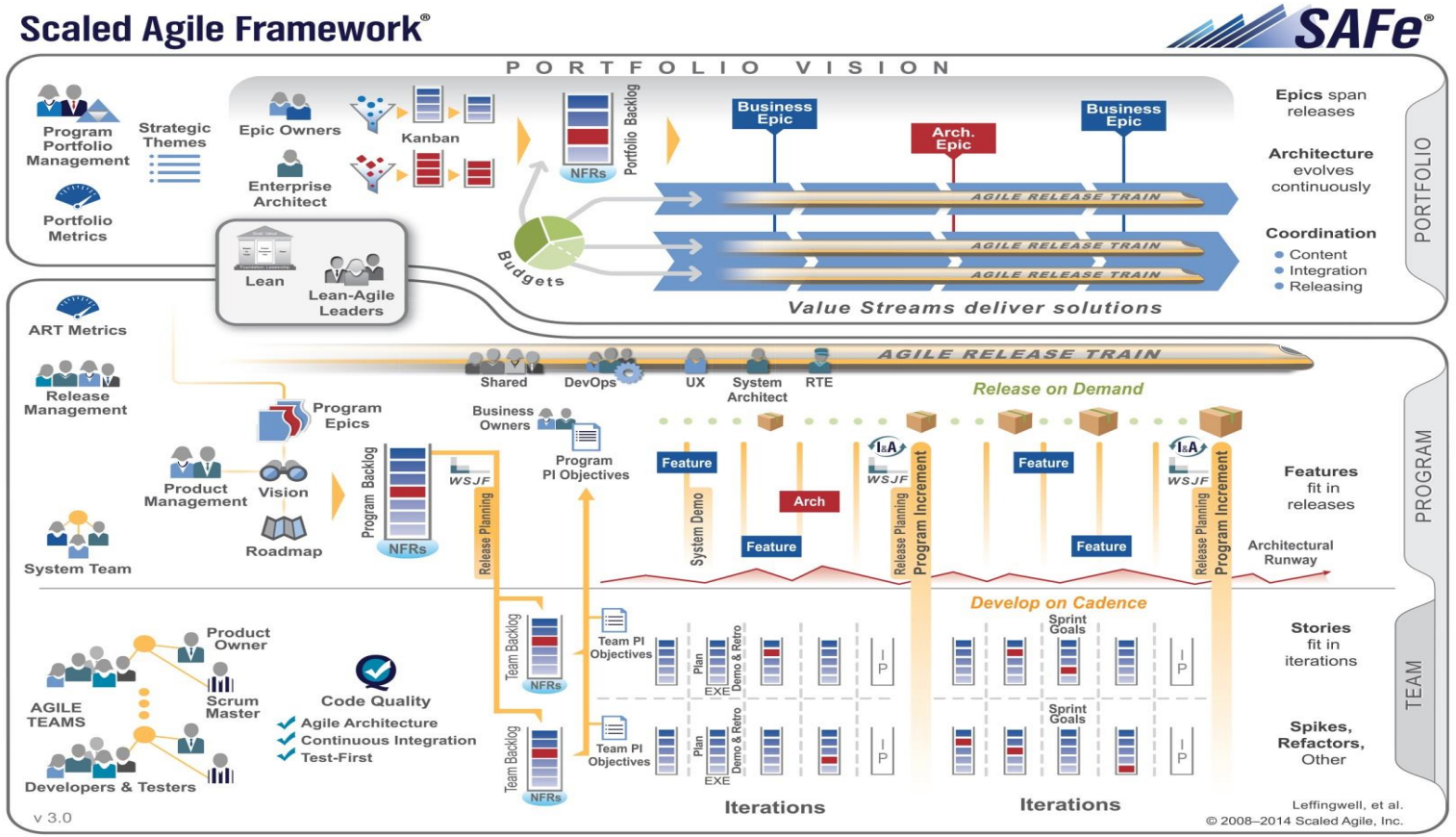


Scaling Scrum is necessary for big projects (more than 2 Scrum teams, i.e. more than 15/20 people)
Scaling is not defined as such in Scrum methodology

The Scrum of Scrums

- The Scrum of Scrums is the group formed by all Scrum Masters of the project
- The Scrum of Scrums meets on a frequent and regular basis (daily to twice a week) after the daily standup meeting
- The Scrum of Scrums works as a team to ensure resolution of issues which cannot be resolved at team level, but at project level
- The Scrum of Scrums is responsible to share successful best practices between teams.

Scaled Agile Framework



Exercise - Scrum Practice

Project Preparation (10 Min)

- Create **a introduction manual for New Opened Restaurant/Hotel** (5 Min)
  - Create Brand and Logo for new Restaurant/Hotel
  - Address location and transportation
  - Design unique features for new Restaurant/Hotel
  - Design target level of service and slogan, and set up Price
- Organize Team (Product Owner, Scrum Master, Team) (5Min)

Project Execution (40 Min)

- Sprint 0: Preparation (5Min)
  - Product Owner discuss product vision and objective with team
  - Create **Backlog**
- Two days Sprint (28 Min)
  - Sprint Planning meeting (8 Min)
  - First Day (10 Min)
    - Stand-up meeting (3 Min)
    - Work (7 Min)
  - Second Day(10 Min)
    - Stand-up meeting (3 Min)
    - Work (7 Min)
- Sprint Review Meeting (3 Min)
- Sprint Retrospective Meeting (4 Min)

CMMI and Agile Comparison

CMMI and Agile Comparison -1

Dimension	CMMI Paradigm	Agile Paradigm
Management	Management plays an important role in ensuring project success. There is much attention to project management	Management is a coaching function (as opposed to traditional command-and-control)that helps to eliminate barriers to progress.
Trust	Some CMMI practices assume the need to compensate for a low-trust environment	Agile methods originated from the recognition that teams work best when they are composed of task-mature individuals operating in high trust groups.
Planning	CMMI promotes macro (project-level) planning with an emphasis on establishing a suitable defined process enabling the project to achieve its objectives.	There are multiple levels of planning, There is a strong emphasis on flexibility and replanning as conditions change.

CMMI and Agile Comparison -2

Dimension	CMMI Paradigm	Agile Paradigm
Market/User Assumption	CMMI is broadly beneficial but particularly so when the target market becomes more mature and process innovation becomes a more important differentiator to organization success.	Agile methods have the most benefit in an emergent and not well-understood target/ market.
Design Presumptions	CMMI presumes the product architecture is selected or created in the early stages of a project and is revisited when it becomes clear the selected architecture is no longer valid or when using an iterative lifecycle.	Projects are most successful when corporate standard architectures are adopted with flexibility applied as the project progresses.
Life-Cycle Emphasis	CMMI has a strong “review-as-you-develop” emphasis. CMMI encourages documentation, analyses, and reviews before product components are integrated into a functional product. CMMI also encourages frequent (early and mid-course) validations to ensure the right product is being built.	Agile methods employ concurrent development, test iterations, and informal peer reviews of work products as necessary
Cost of Failure	Historically, CMMI was developed in a domain of high cost of failure.	Agile methods have flourished in a domain of low cost of failure or linear incremental cost of failure. Examples within this domain include Internet commerce, social networking, and games development.

Configuration Management

- A Support Process Area at Maturity Level 2

Configuration Management in CMMI

Purpose

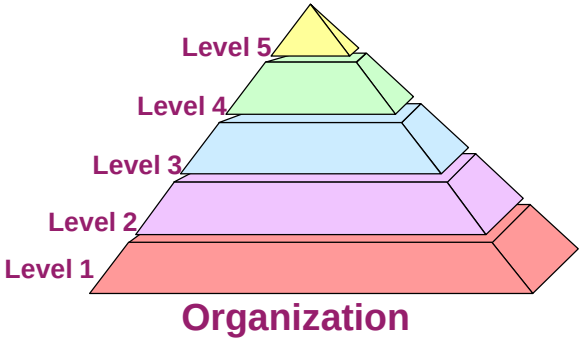
- The purpose of Configuration Management (CM) is to establish and maintain the integrity of work products using configuration identification, configuration control, configuration status accounting, and configuration audits.

Examples of work products that can be placed under configuration management include the following:

- Hardware and equipment /Drawings
- Product specifications / Requirements/ Architecture documentation and design data
- Tool configurations / Compilers / Installation logs
- Code and libraries
- Test tools and test scripts
- Product data files
- Product technical publications
- Plans
- User stories / Iteration backlogs
- Process descriptions / Product line plans, processes, and core assets

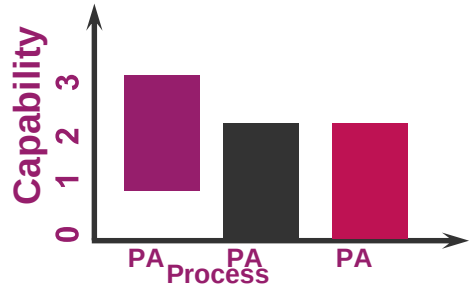
CMMI-DEV IN A NUTSHELL

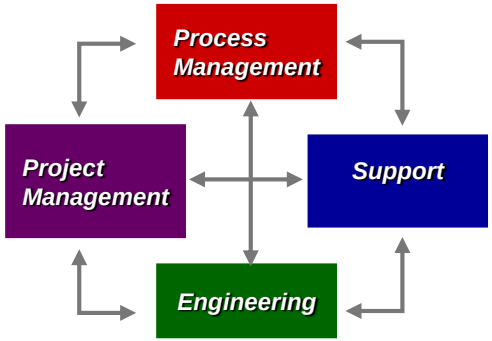
Staged



Name	Area	Level
Requirements Management	Engineering	2
Project Monitoring and Control	Project Management	2
Project Planning	Project Management	2
Supplier Agreement Management	Project Management	2
Configuration Management	Support	2
Measurement and Analysis	Support	2
Process and Product Quality Assurance	Support	2
Product Integration	Engineering	3
Requirements Development	Engineering	3
Technical Solution	Engineering	3
Validation	Engineering	3
Verification	Engineering	3
Organizational Process Definition	Process Management	3
Organizational Process Focus	Process Management	3
Organizational Training	Process Management	3
Integrated Project Management	Project Management	3
Risk Management	Project Management	3
Decision Analysis and Resolution	Support	3
Organizational Process Performance	Process Management	4
Quantitative Project Management	Project Management	4
Organizational Performance Management	Process Management	5
Causal Analysis and Resolution	Support	5

Continuous





Specific Goals and Practices of CM in CMMI

SG 1 Establish Baselines

- SP 1.1 Identify Configuration Items
- SP 1.2 Establish a Configuration Management System
- SP 1.3 Create or Release Baselines

SG 2 Track and Control Changes

- SP 2.1 Track Change Requests
- SP 2.2 Control Configuration Items

SG 3 Establish Integrity

- SP 3.1 Establish Configuration Management Records
- SP 3.2 Perform Configuration Audits

Configuration Identification

The purpose of configuration identification is to ensure that all the products and deliverables to be controlled

- are uniquely named,
- have an identified owner.

Items to control include

- Requirements, Design,
- Source Code,
- Test Cases, Code and Results
- Tools and Scripts
- Binaries and Streams

Configuration items can include hardware, equipment, and tangible assets as well as software and documentation

Baseline

A baseline is the complete status of a configuration item at a given point in time.

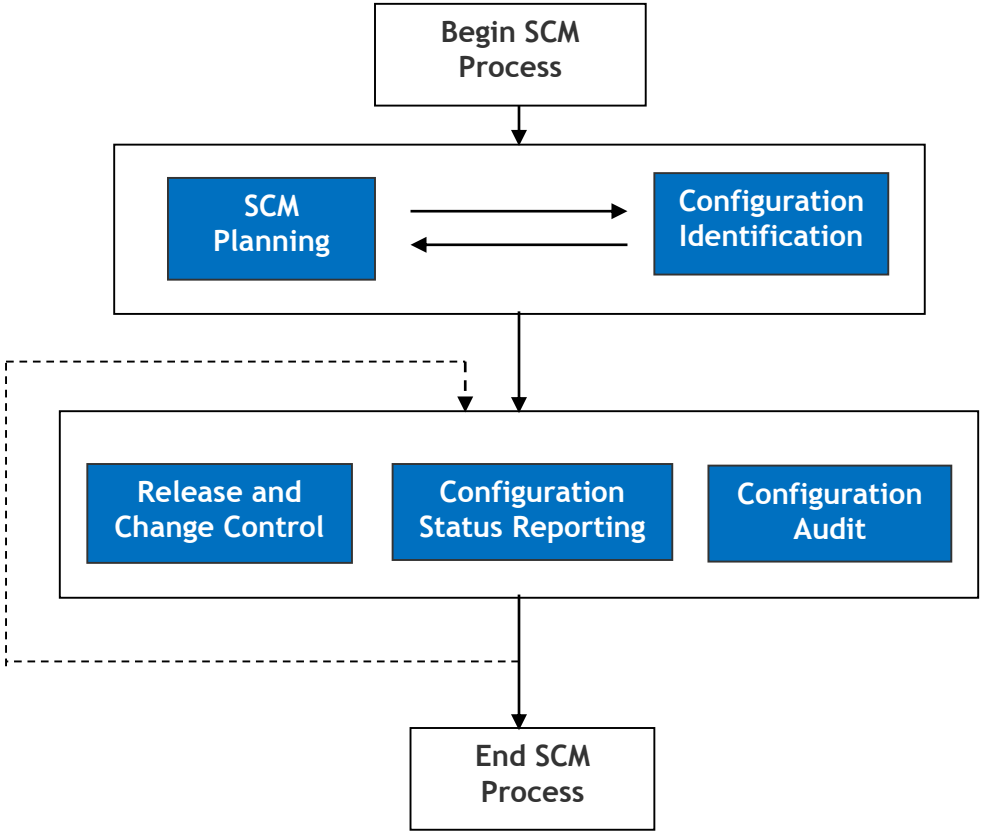
A software baseline can be a set of requirements, design, source code files and the associated executable code, build files, and user documentation (associated entities) that have been assigned a unique identifier.

A baseline should have the following attributes:

- Known Functionality: The features and functions of a particular baseline will be documented and available for reference.
- Known Quality: The quality of a baseline will be well defined in terms of known bugs and tests performed.
- Unchangeable: Any changes may become a new baseline, but the existing baseline does not change.
- Re-creatable: All the CIs comprising the baseline can be recreated at any point in time. This is critical for maintaining development, testing, and multiple release versions

A baseline is taken at each milestone and delivery of the project

SCM Process Example



PR and CR

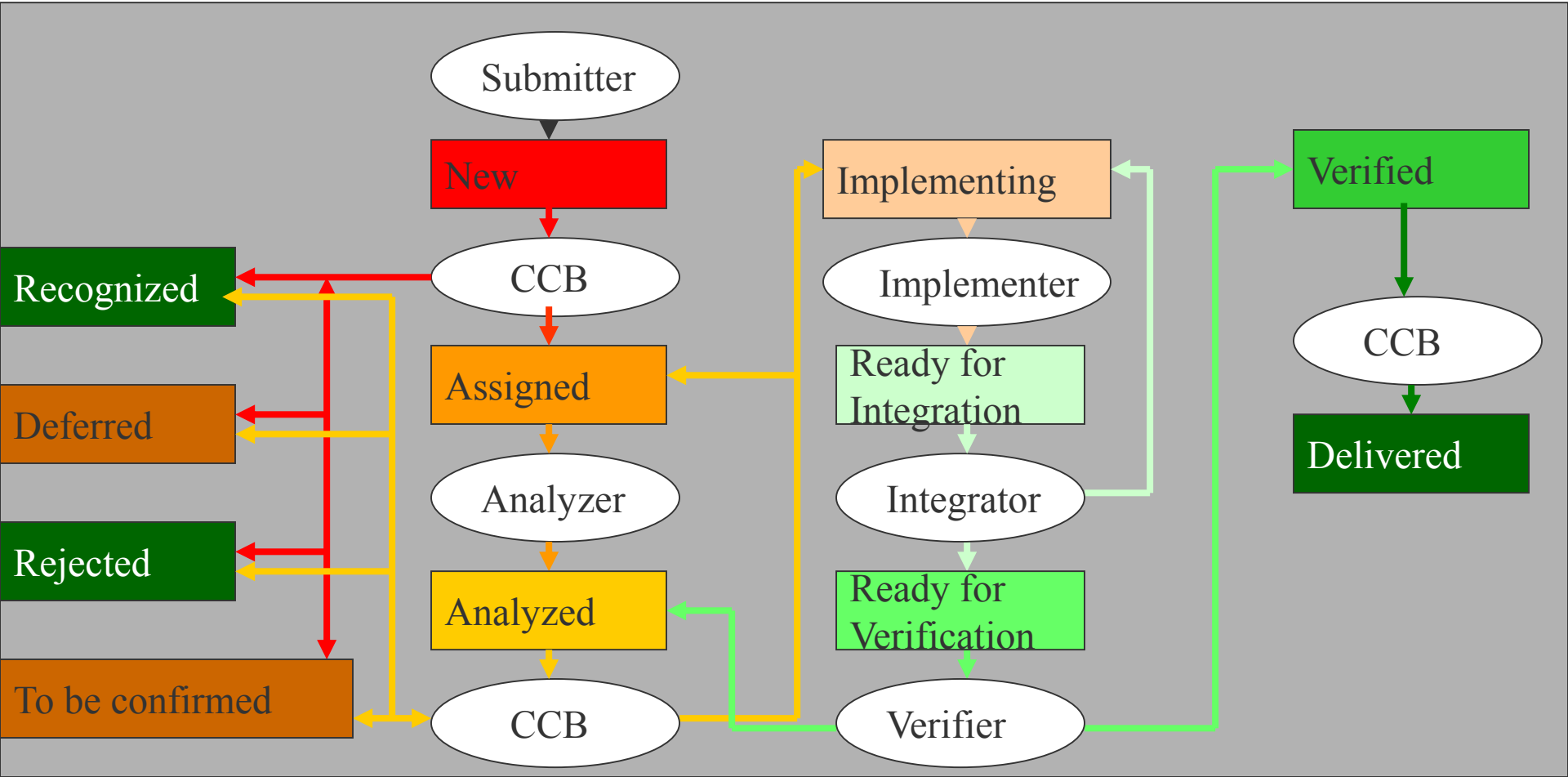
PR(Problem Report)

- Issue found in an identified release
- if not fixed, this issue implies non compliance to the applicable specifications of the product

CR(Change Request)

- Implementation change request (from customer/project/…), which implies a modification of the applicable specifications of the product
- Test cases and Traceability matrix may be updated to reflect changes in the project consistency

Change Control Process



Change Control Board (CCB)

A CCB is created for each new project.

The CCB should involve all the relative key roles in project: Project Manager, Integrator, Test engineer, Technical Leader, Program Manager if needed, and any other roles as options who is necessary on the decision of change control depends on the topics in CCB meeting

CCB's responsibilities are:

- Managing Change Requests and Problem Reports of the project.
- Approving also the baselines of the project, and plan the baselines changes according to the accepted PR and CR.
- Providing a CCB report as output

Configuration audits

Configuration audits are performed to check that SCM activities are effective and efficient for the project and the organization

- Changes made on an existing baseline are implemented as intended
- The SCM plan is being followed

SCM audits are useful during project transitions

- Requirements, design, implementation, test, …

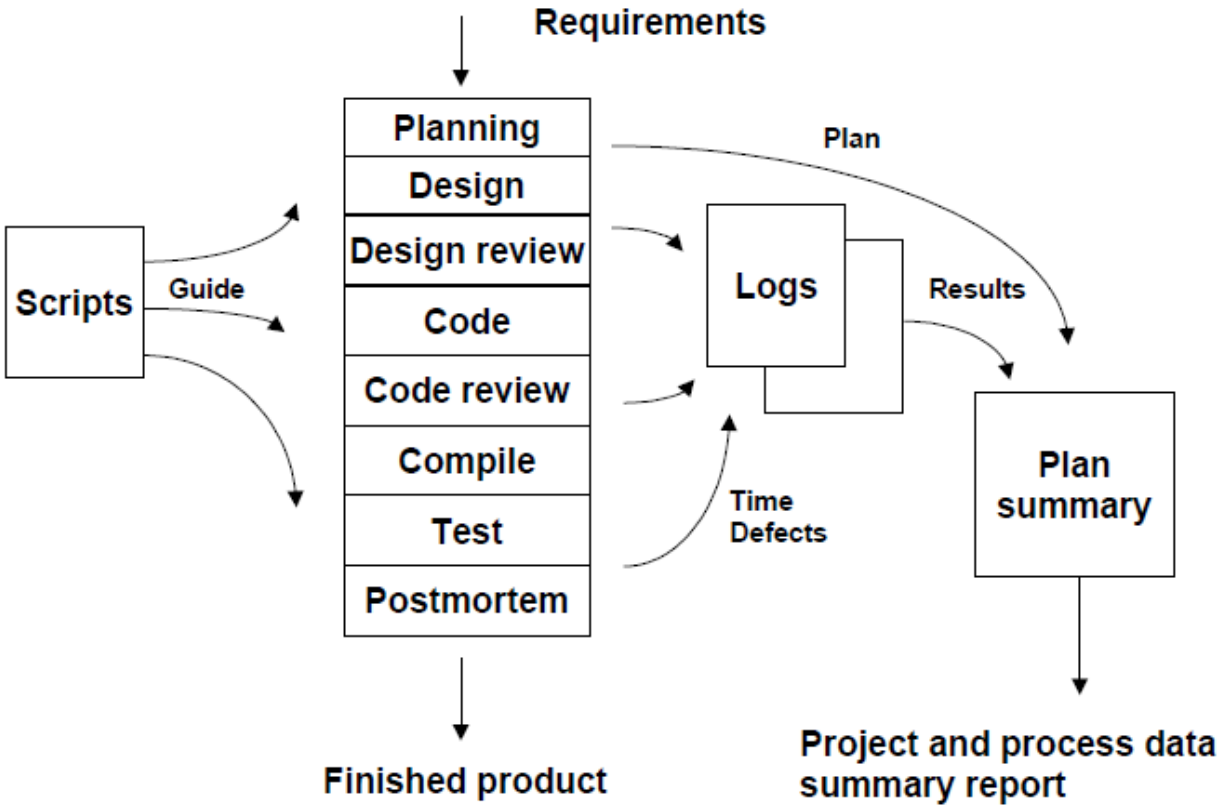
PSP and TSP

Personal Software Process (PSP)

The Personal Software Process provides engineers with a disciplined personal framework for doing software work.

The PSP process consists of a set of methods, forms, and scripts that show software engineers how to plan, measure, and manage their work.

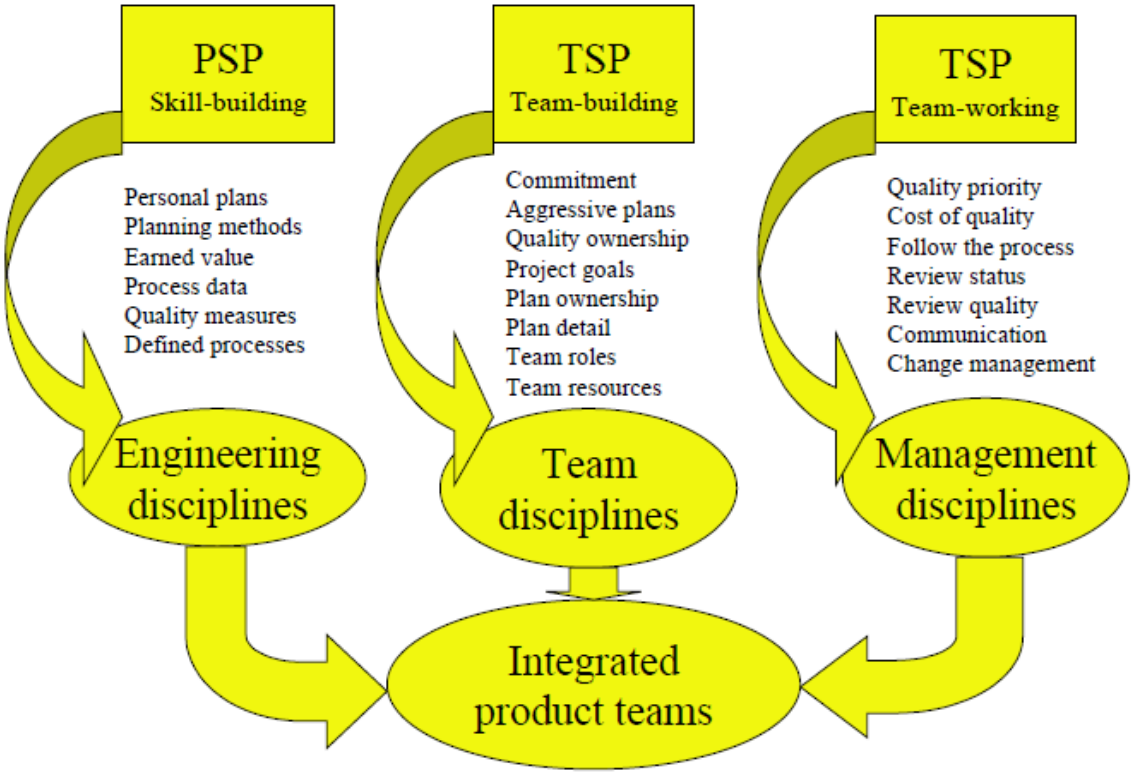
PSP Process Flow



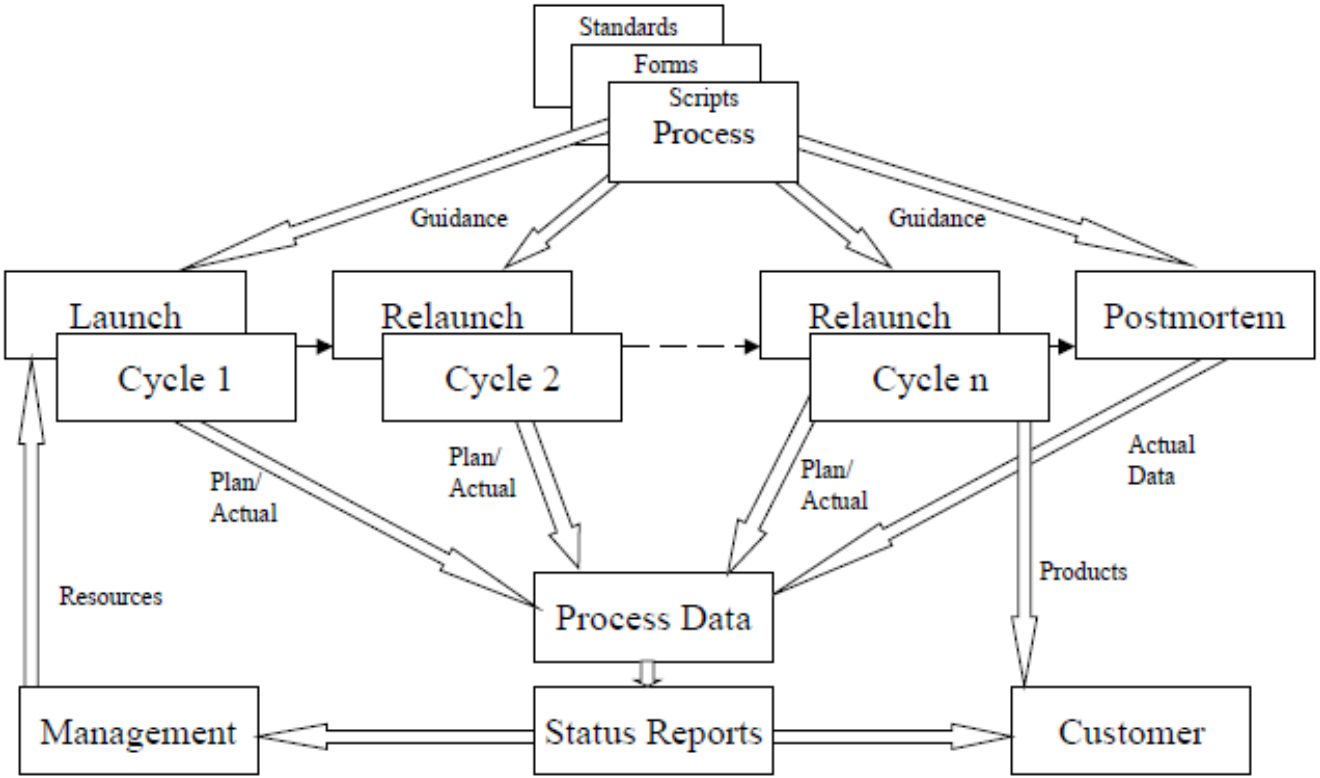
Team Software Process (TSP)

Team Software Process (TSP) guides engineering teams that are developing software-intensive products. Using TSP helps organizations establish a mature and disciplined engineering practice that produces secure, reliable software in less time and at lower costs.

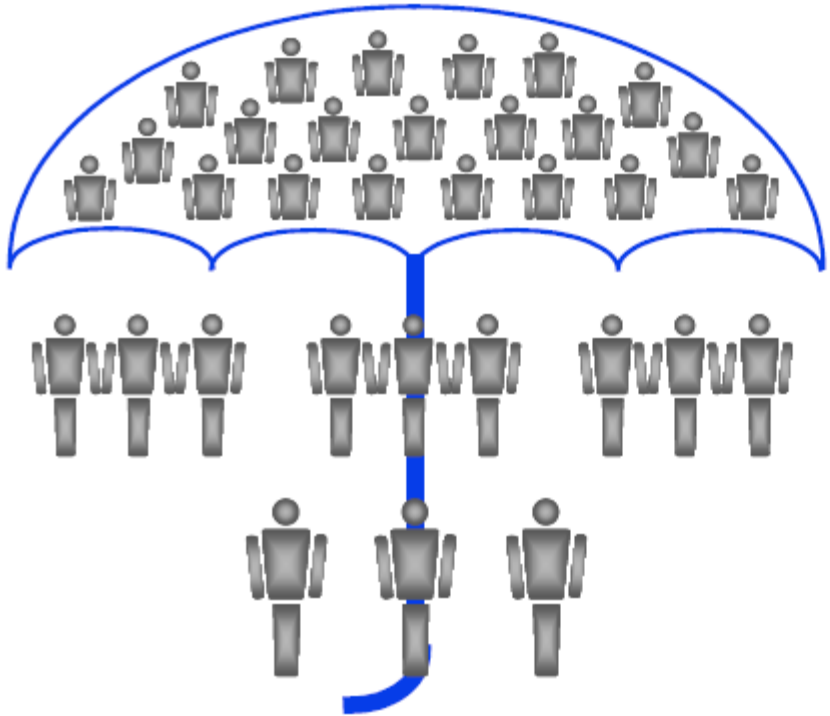
TSP Team Building



TSP Process Flow



CMM, TSP and PSP Relationship



CMM - Improves organization's capability; management focus.

TSP - Improves team performance; team and product focus.

PSP - Improves individual skills and discipline; personal focus.

Defect Management

Defect

A Software Defect is a condition in a software product which does not meet a software requirement (as stated in the requirement specifications) or end-user expectations (which may not be specified but are reasonable).

A defect is an error in coding or logic that causes a program to malfunction or to produce incorrect/unexpected results.

Defects can be in programs, in designs, or even in the requirements, specifications, or other documentation.

Defects can be found in any process phase.

Also called as Bug, Problem Report(PR)

Defect Classification- Severity

Critical

- Some key functionalities don’t work at all.
- The product performance will be degraded
- High Risk associated with release to production

Major

- Some functionalities work in a degraded mode

Minor

- Low risk associated with release to production

Defect Classification- Priority

Defect Priority (Bug Priority) indicates the importance or urgency of fixing a defect:

- High
- Medium
- Low

Defect Classification- Origin

Requirement

Design

Code

Test

Defect Classification- Type

Function

Assignment

Interface

Checking

Build/Package/Merge

Documentation

Algorithm

User Interface

Performance

Norms

Defect Classification- Status

New

Open

Resolved

Rejected

Verified

Closed/Delivered

Deferred

Defect Description

Title: Mandatory

Precondition: Mandatory

Procedure: Mandatory, steps to reproduce the defect, steps should be separated by digital number.

Observed result: Mandatory

Expected result: Mandatory

Test Case ID: Mandatory

Document reference: Optional

Comments: Optional

Attachment: Optional

Defect Description Example

Title:

Parameters are not persistent after software download

Preconditions:

Use XXX.trp

Procedure:

1- Change default value of the memo countdown period in User Settings menu (value:2 for example)
2- Change default value of the picture quality in Settings menu (value: standard for example)
3- Do SW download of SW 3.0.1 to SW 3.0.2

Observed Result:

Values of Memo Countdown period and Picture Quality are changed after the download
Value of memo countdown period : 1
Value of picture quality : high(RGB)

Expected Result:

Parameters are not modified after a download of one new software version.

Test Case ID:

SIT_SSU_003

Comments:

User's settings may be lost while user has no idea about this, then it will cause confusion to end user.

Defect Description Requirement

Condense - Say it clearly but briefly

Accurate - Is it a defect or could it be wrong operation, misunderstanding, etc.?

Neutralize - Just the facts. No zingers. No humor. No emotion.

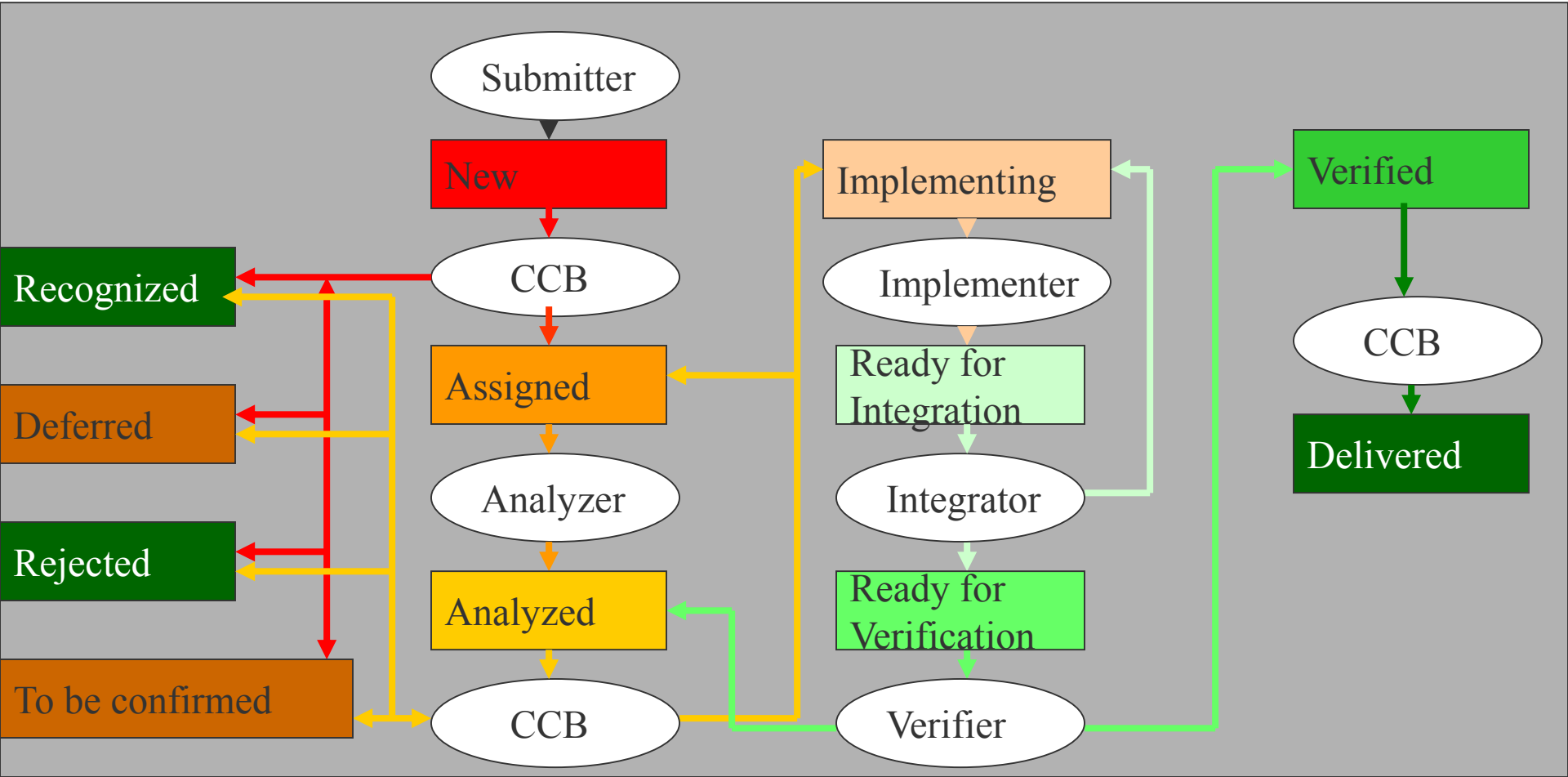
Clear - Explicitly, what is the problem?

Isolate - What has been done to isolate the problem?

Reproduce - What are the essentials in triggering/reproducing this problem? (environment, steps, conditions)

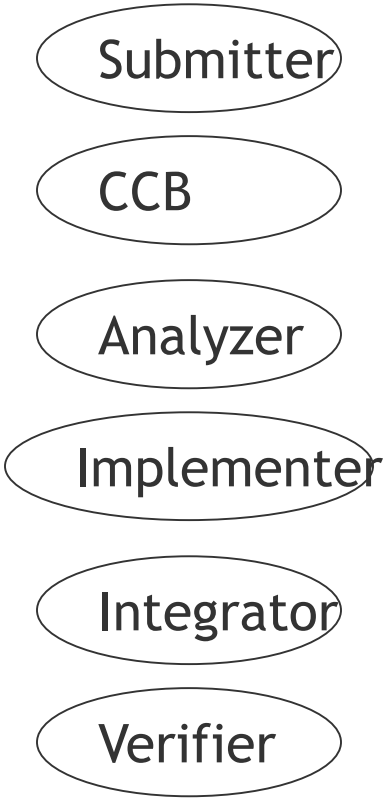
Impact - What is the impact to the customer? What is the impact to test?

Problem Report Workflow Example



Problem Report Workflow - Roles

6 Roles



Change Control Board (CCB)

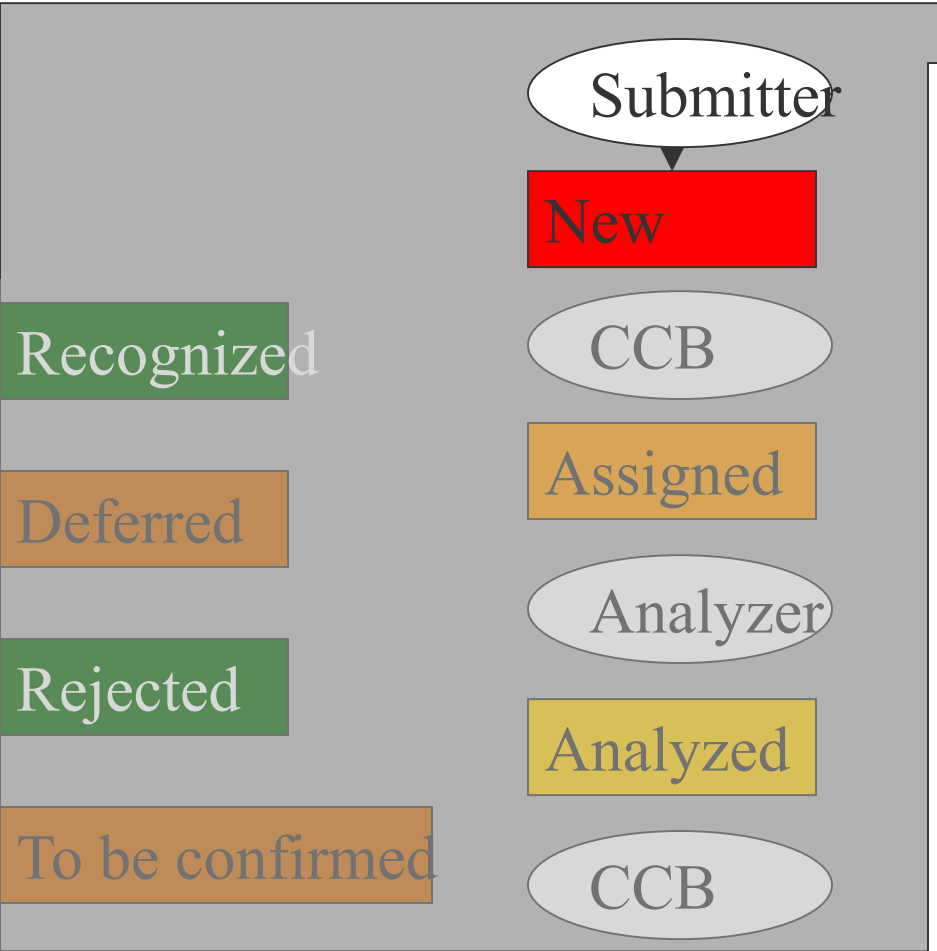
A CCB is created for each new project.

The CCB should involve all the relative key roles in project: Project Manager, Integrator, Test engineer, Technical Leader, Program Manager if needed, and any other roles as options who is necessary on the decision of change control depends on the topics in CCB meeting

CCB's responsibilities are:

- Managing Change Requests and Problem Reports of the project.
- Approving also the baselines of the project, and plan the baselines changes according to the accepted PR and CR.
- Providing a CCB report as output

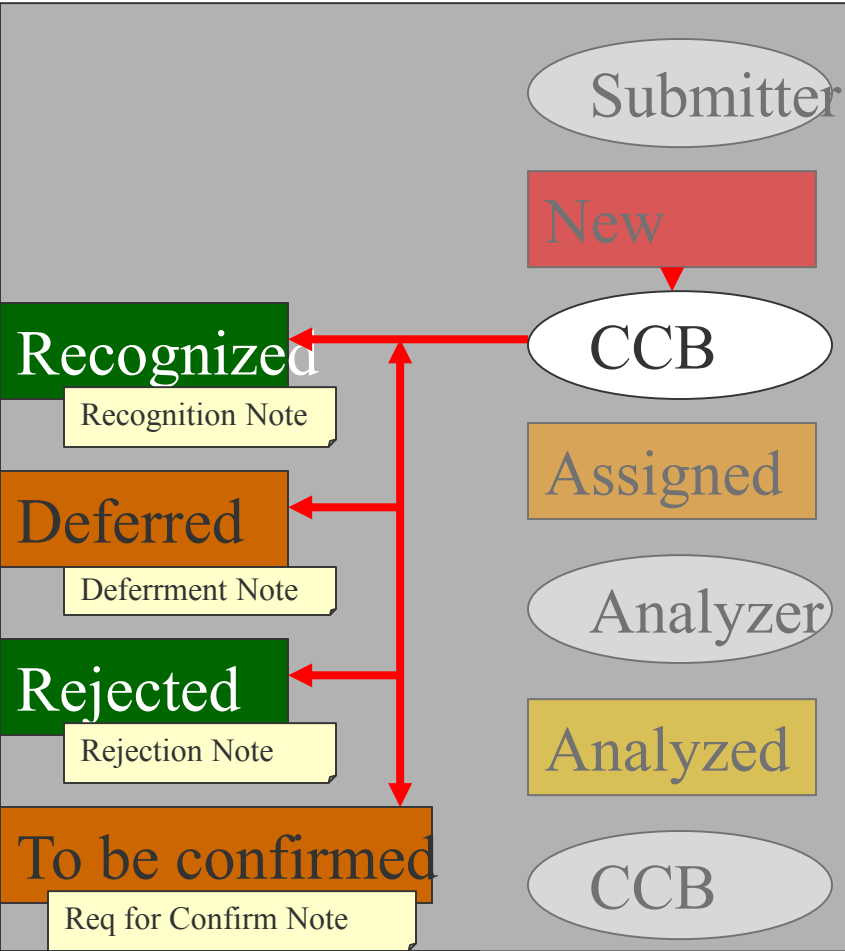
Submission



The Submitter submit the PR/CR in the dedicated project database, giving the following informations :

- Title
- Description
- Record Type
- Scope
- Severity
- Feature
- Found in Sw Release (NA in case of CR)
- Found in Hw Release (NA in case of CR)
- Found in Integration Label
- How Found
- CI impacted
- Repeatibility
- Associated Requirement
- External Reference

Recognize/Deferr/Reject/To be confirm



The CCB checks each **New** PR/CR
and may decide to :

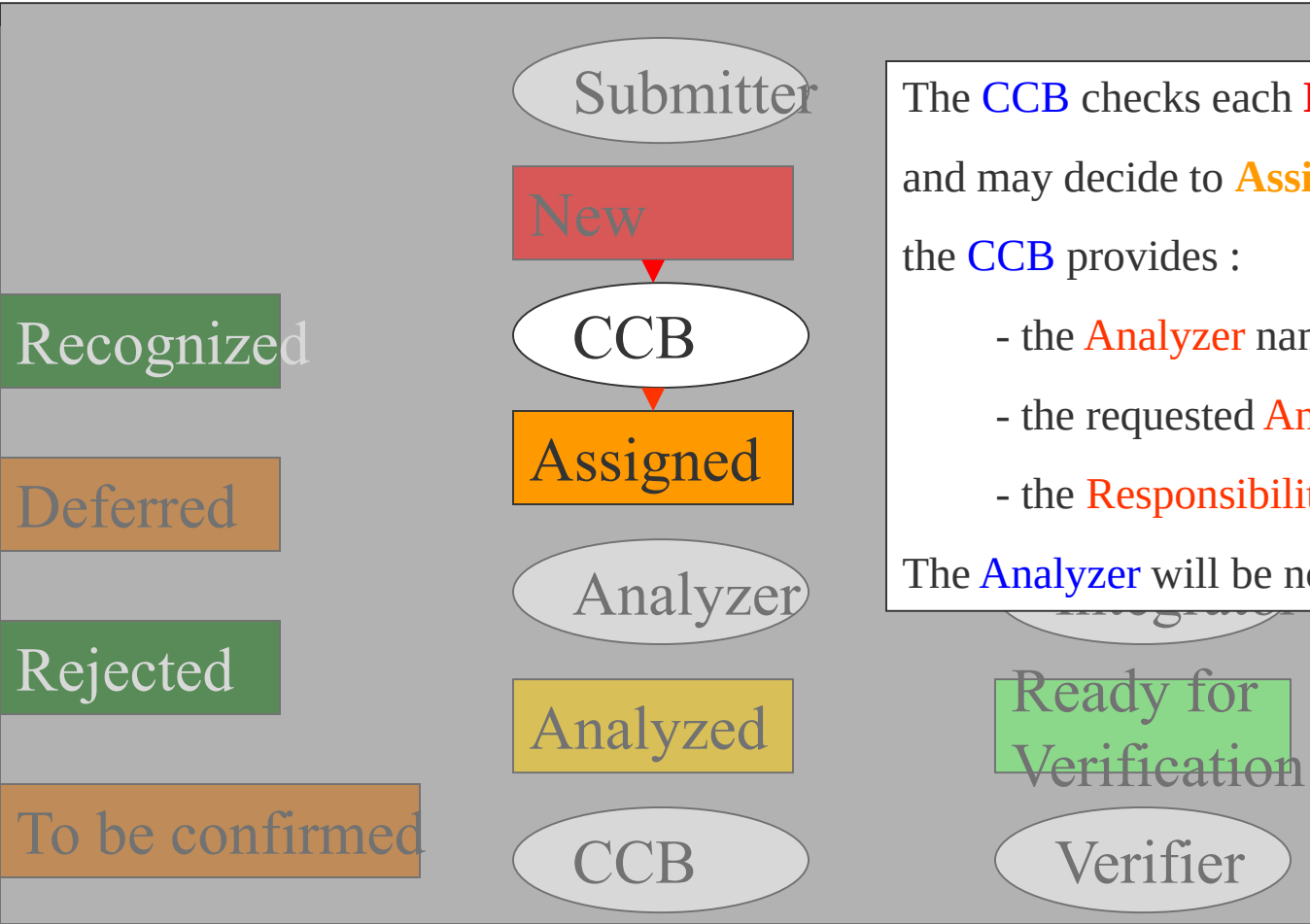
- **Recognize** it :
known limitation of the product :
decision of the Program Manager
(optional member of the CCB)

- **Deferr** it :
decision of the Program Manager
(optional member of the CCB) or SPL

- **Reject** it
decision of the Program Manager
(optional member of the CCB) or SPL

- **To be Confirmed** :
the PR/CR will be re-discussed in the CCB
when it will be confirmed

Assign



The CCB checks each **New** PR/CR

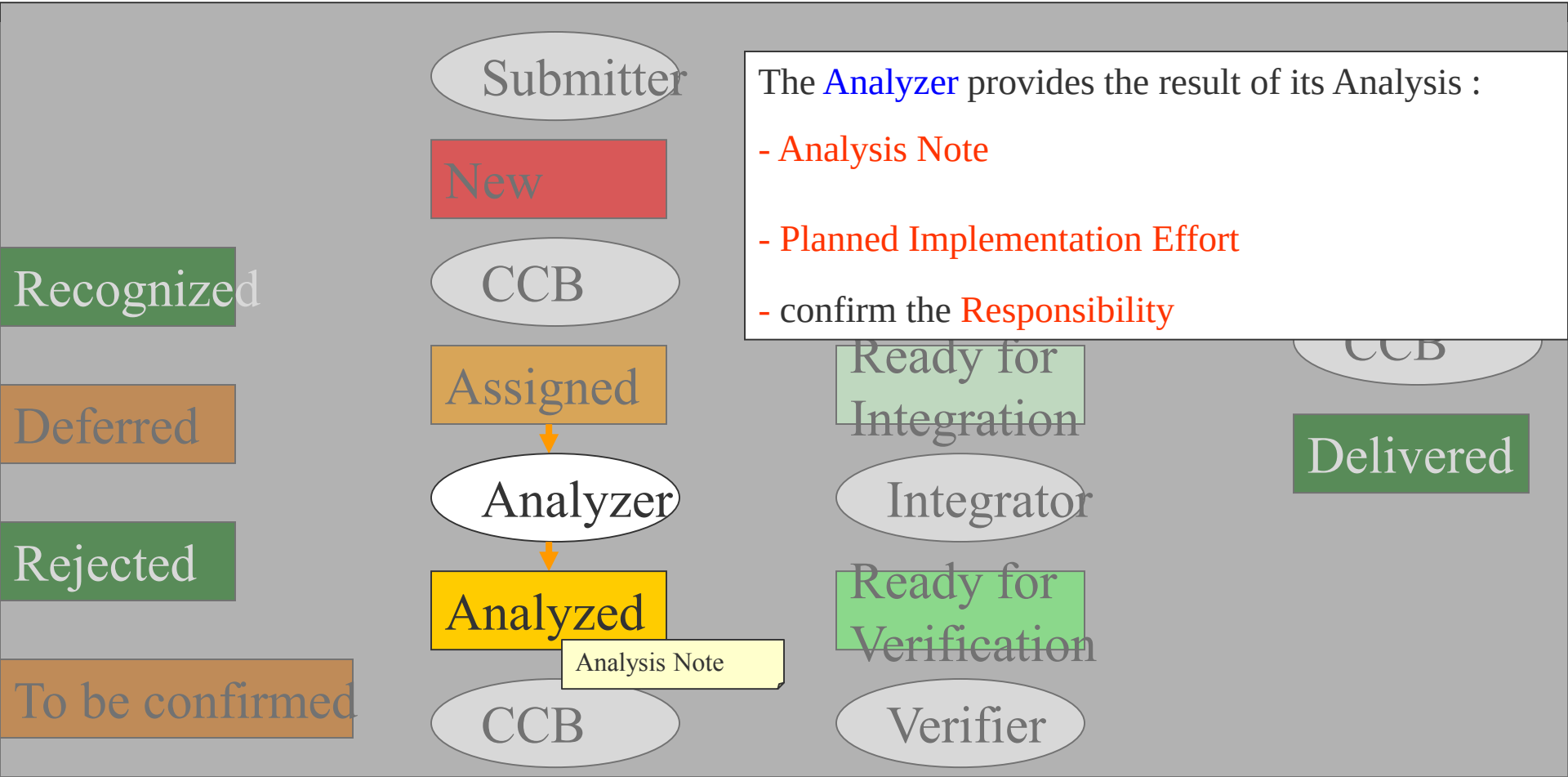
and may decide to **Assign** it :

the CCB provides :

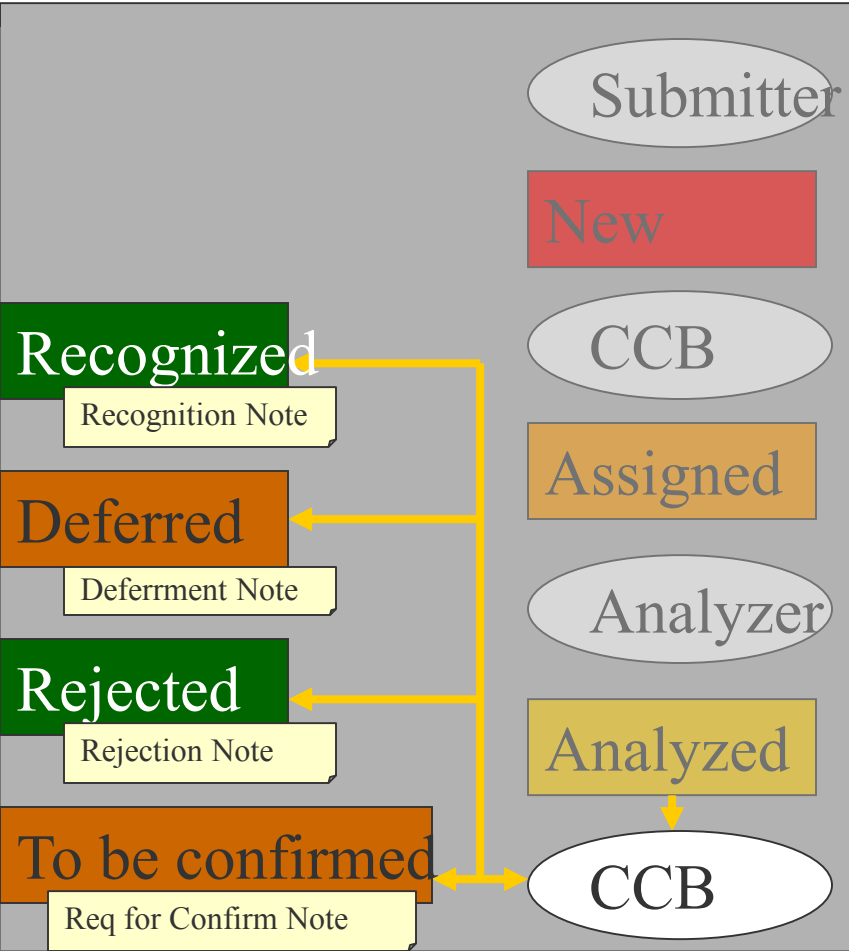
- the Analyzer name
- the requested Analysis Date
- the Responsibility

The Analyzer will be notified of the Analysis request

Analyzed


Submitter
New
CCB
Assigned
Analyzer
Analyzed
Analysis Note
CCB
Recognized
Deferred
Rejected
To be confirmed
Ready for Integration
Integrator
Ready for Verification
Verifier
Delivered
The Analyzer provides the result of its Analysis :
- Analysis Note
- Planned Implementation Effort
- confirm the Responsibility

Recognize/Deferr/Reject/To be confirm



The CCB checks each **Analyzed** PR/CR and may decide,
knowing the Analysis result to

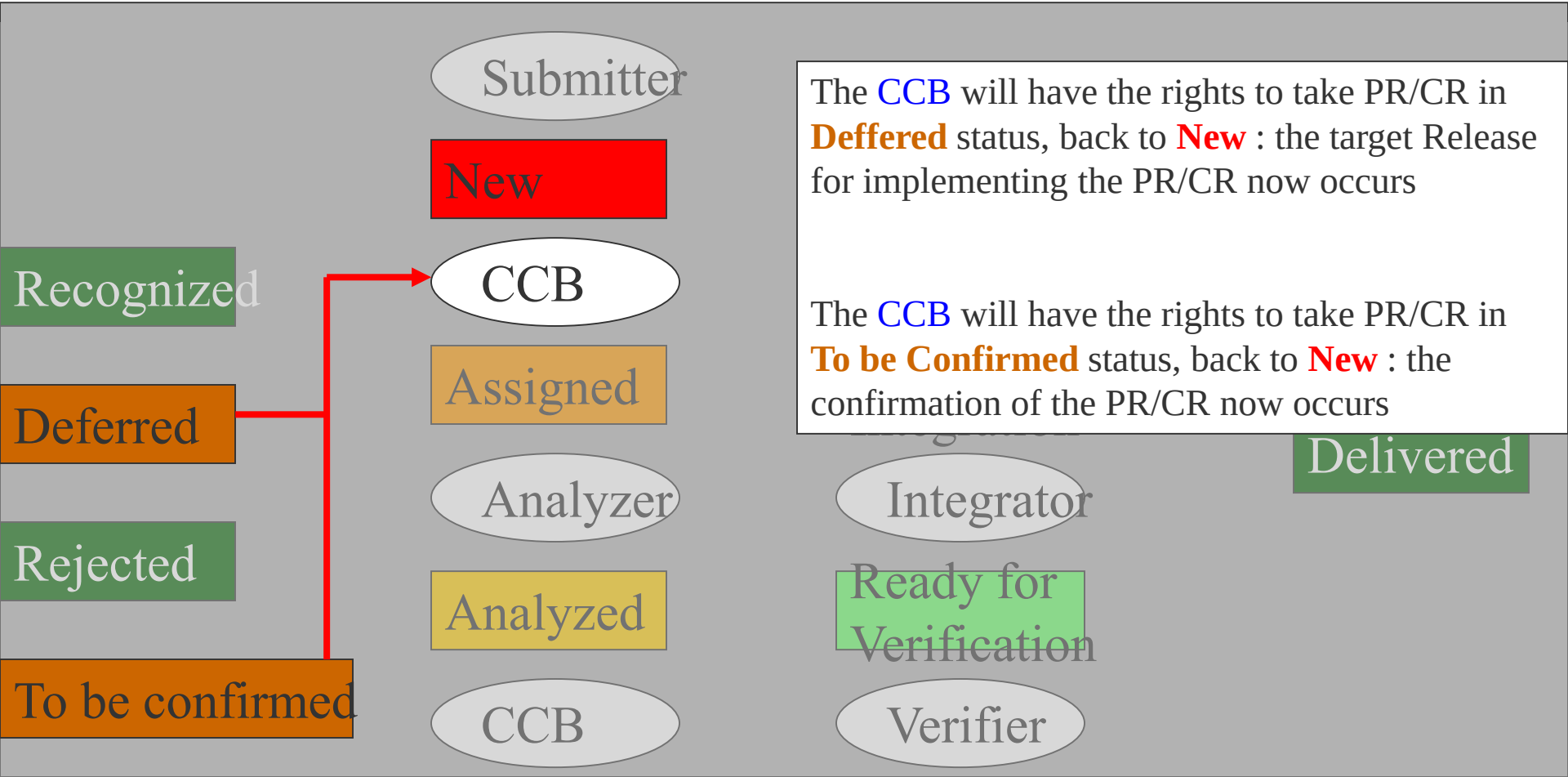
- **Recognize** it :
known limitation of the product :
decision of the Program Manager
(optional member of the CCB)

- **Deferr** it :
decision of the Program Manager
(optional member of the CCB) or SPL

- **Reject** it
decision of the Program Manager
(optional member of the CCB) or SPL

- **To be Confirmed** :
the PR/CR will be re-discussed in the CCB
when it will be confirmed

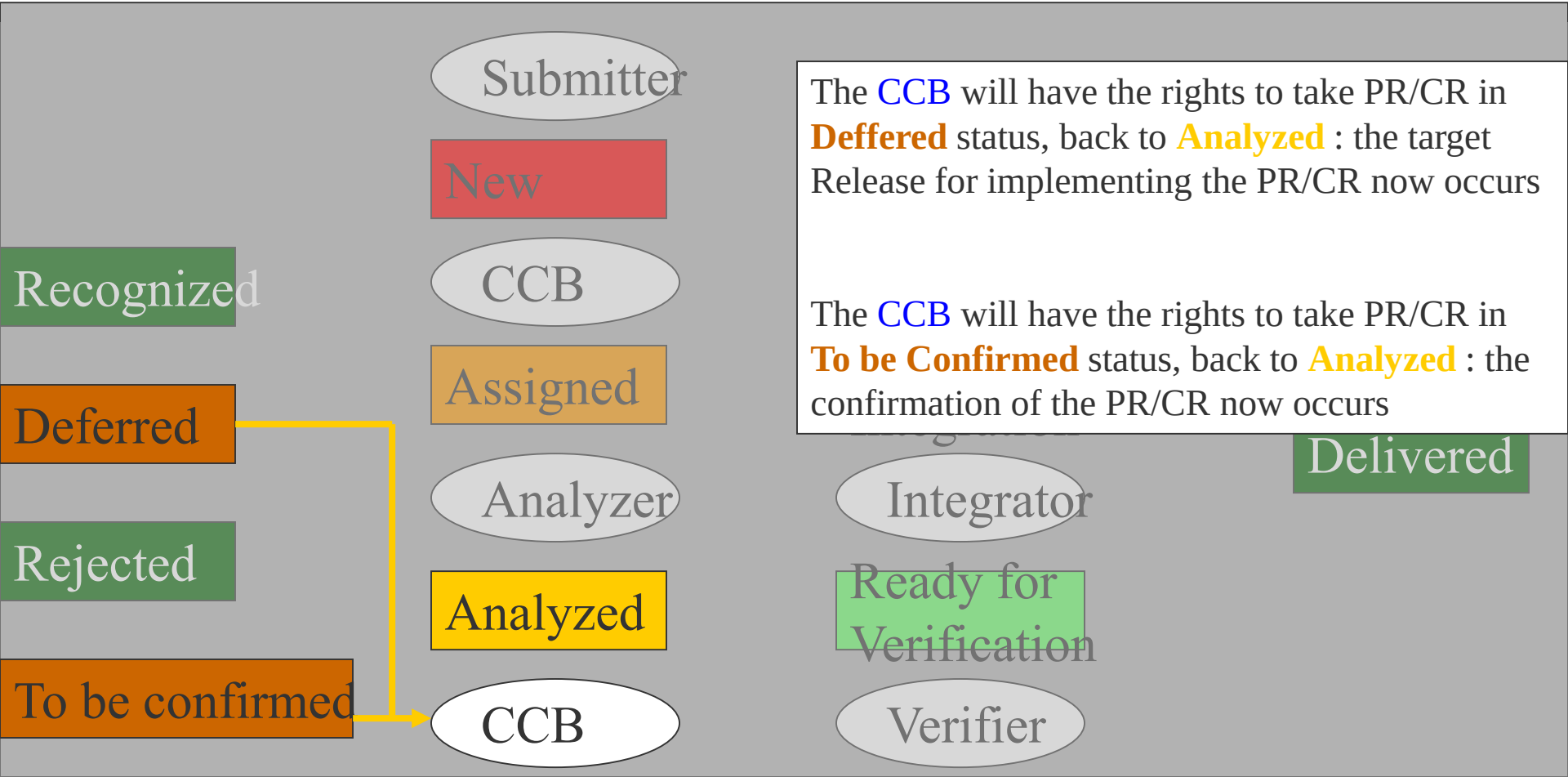
Back to New



The CCB will have the rights to take PR/CR in **Deffered** status, back to **New** : the target Release for implementing the PR/CR now occurs

The CCB will have the rights to take PR/CR in **To be Confirmed** status, back to **New** : the confirmation of the PR/CR now occurs

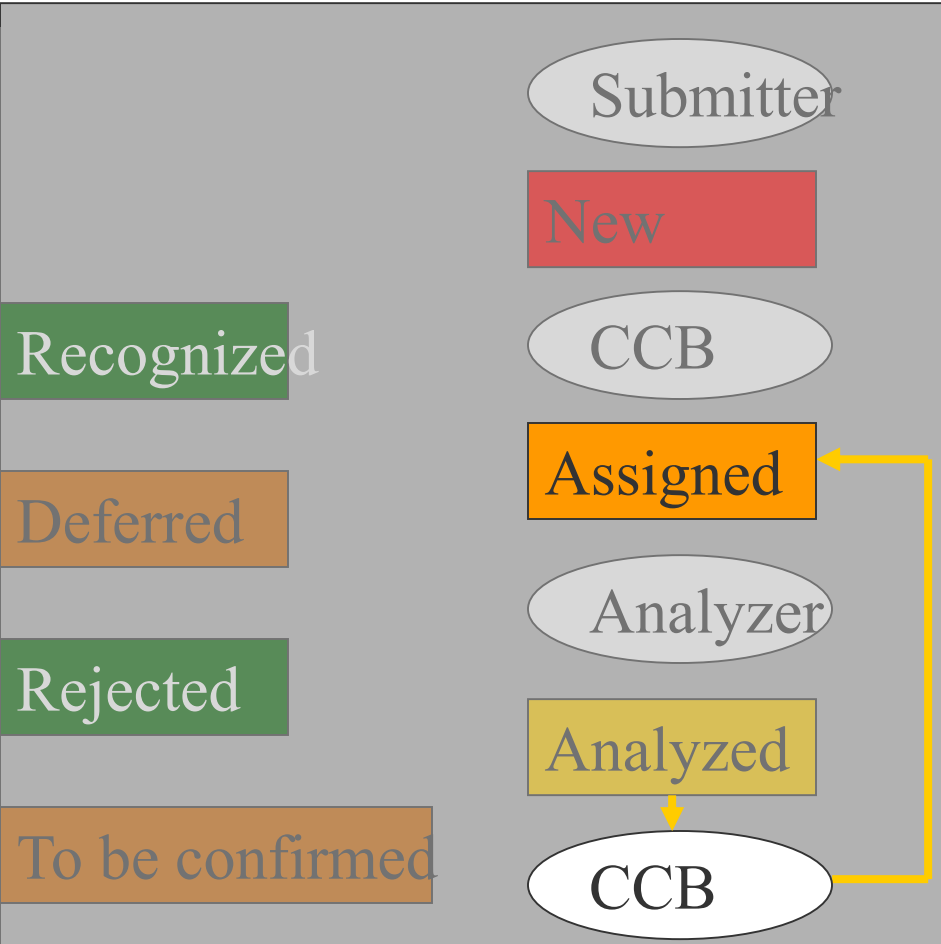
Back to Analyzed



The CCB will have the rights to take PR/CR in **Deffered** status, back to **Analyzed** : the target Release for implementing the PR/CR now occurs

The CCB will have the rights to take PR/CR in **To be Confirmed** status, back to **Analyzed** : the confirmation of the PR/CR now occurs

Back to Assigned



The CCB checks each **Analyzed** PR/CR and may decide to :

Assign again:

- the Analysis results are not complete

- an additional analysis with another Analyzer is requested

The CCB provides :

- the Analyzer name

- the requested Analysis Date

The new Analyzer will be notified of the Analysis request

Implementing

The CCB checks each **Analyzed** PR/CR and may decide to :

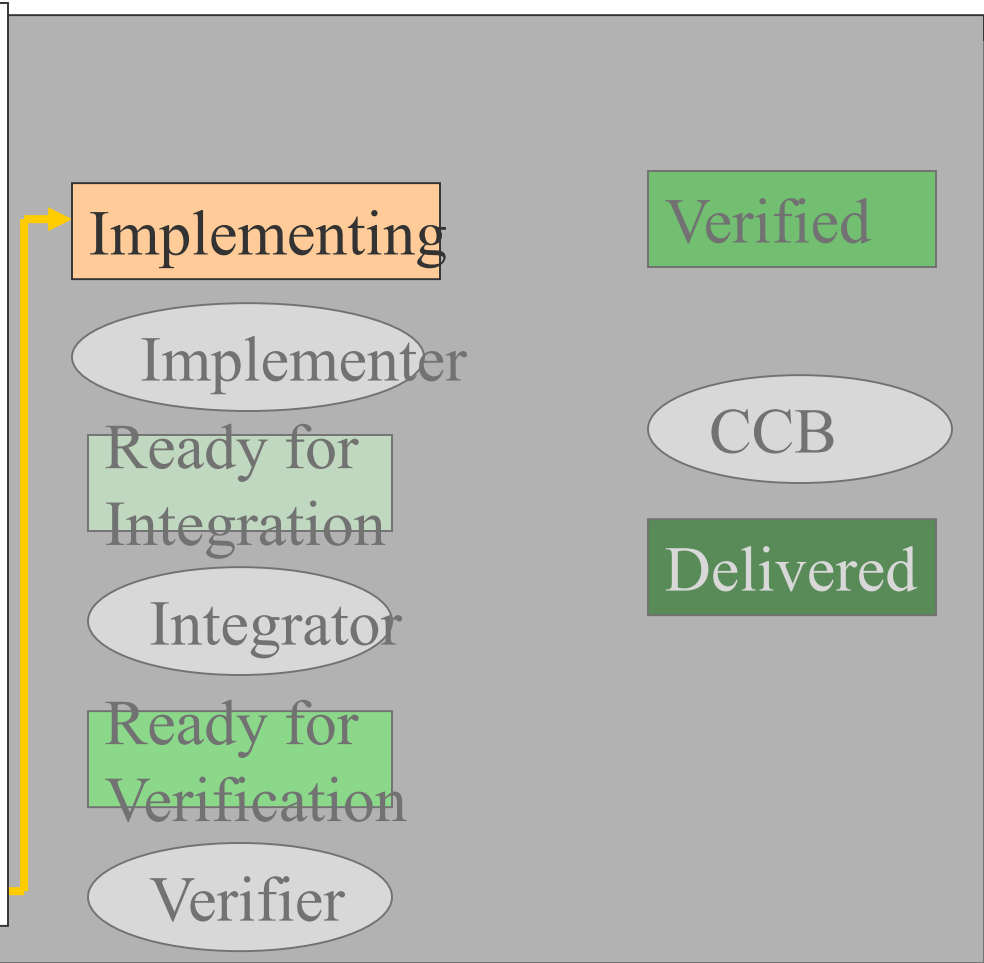
Implementing : the CCB designates

- the Implementer
- the Integrator
- the Verifier

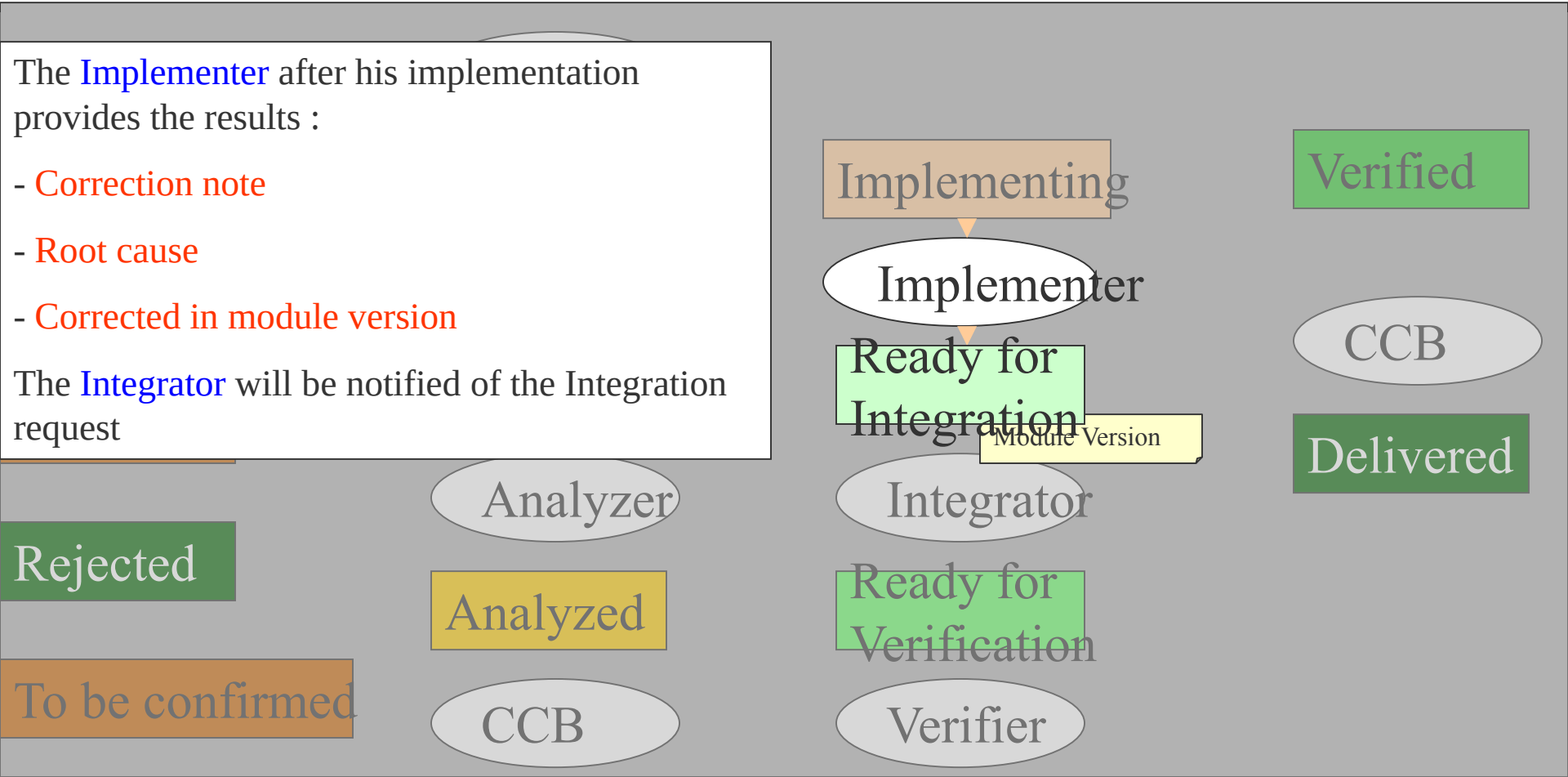
The CCB gives also :

- the requested RFI Date
- the requested RFV Date
- the requested Verified Date

The Implementer will be notified of the Implementation request



Ready for Integration



Back to Implementing / Ready for Verification

The Integrator after its integration, provides the results :

- Integration Note

and may decide to go :

-back to **Implementing** :

Integration tests failed

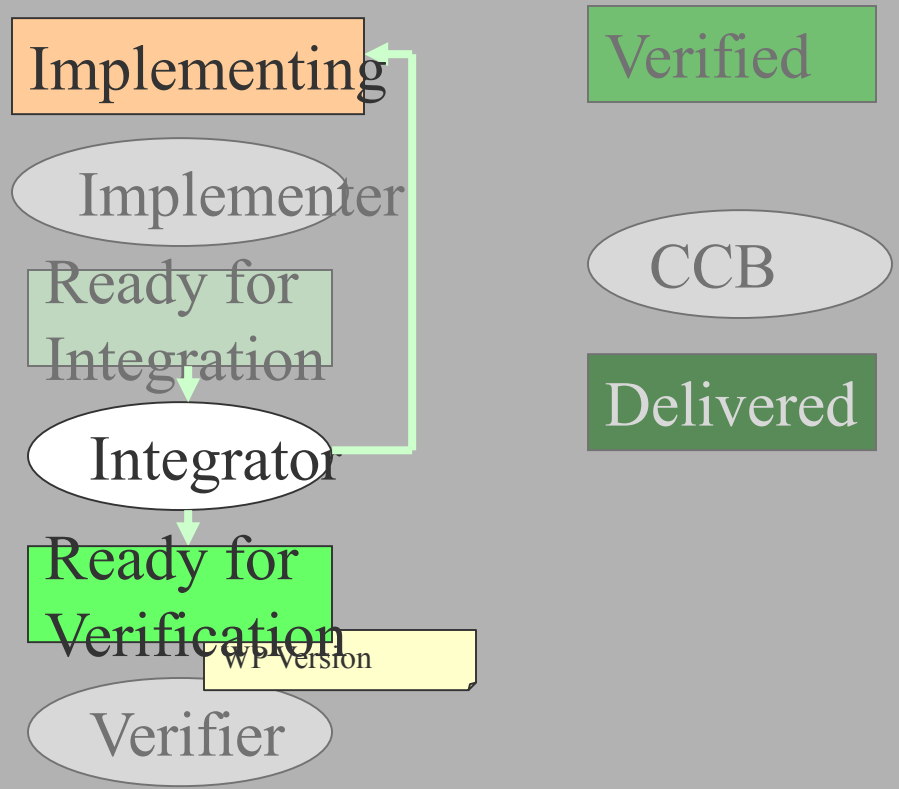
-to **Ready for Verification** :

the Corrected in WP version is provided

The Fixed in Release can be also

provided (if possible)

The Verifier will be notified of the Verification request



Back to Analyzed / Verified

The Verifier after its verification, provides the results :

- Verification Note

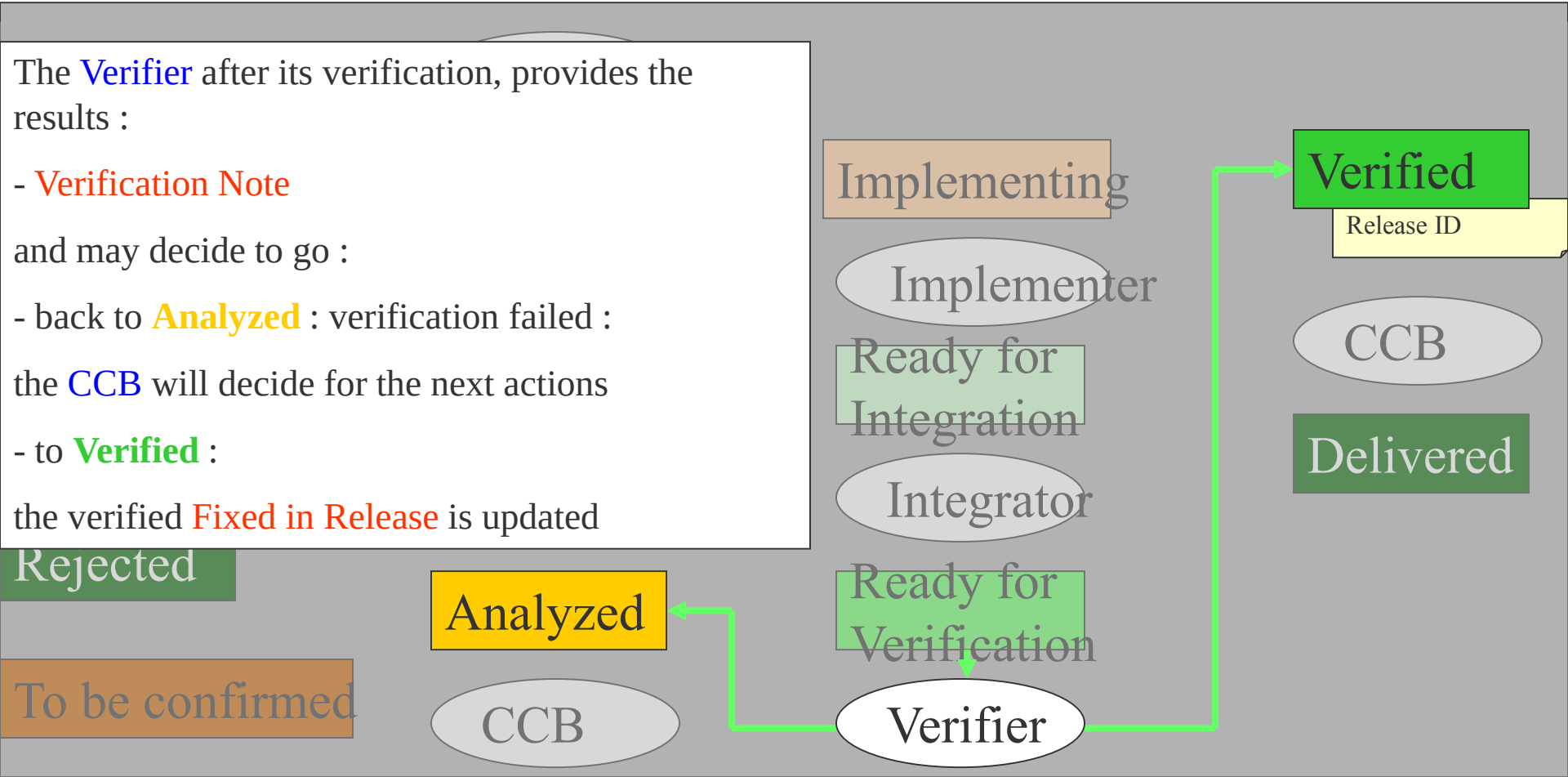
and may decide to go :

- back to **Analyzed** : verification failed :

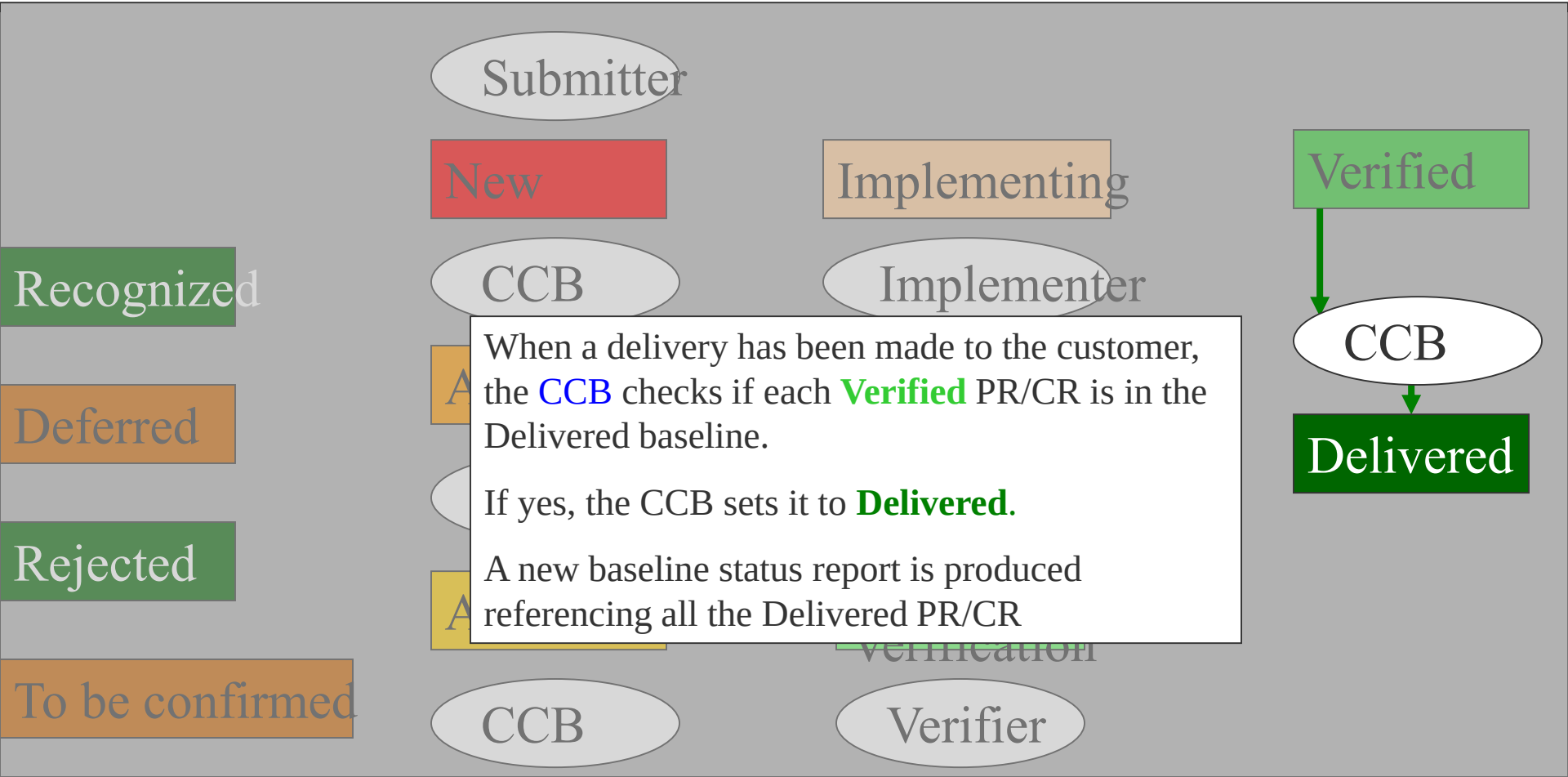
the CCB will decide for the next actions

- to **Verified** :

the verified Fixed in Release is updated



Delivered



When a delivery has been made to the customer, the CCB checks if each **Verified** PR/CR is in the Delivered baseline.

If yes, the CCB sets it to **Delivered**.

A new baseline status report is produced referencing all the Delivered PR/CR

Defect Analysis

Orthogonal Defect Classification

- Developed at IBM in the 1990s by Ram Chillarege
- Methodology to characterize software defects and translate into process defects

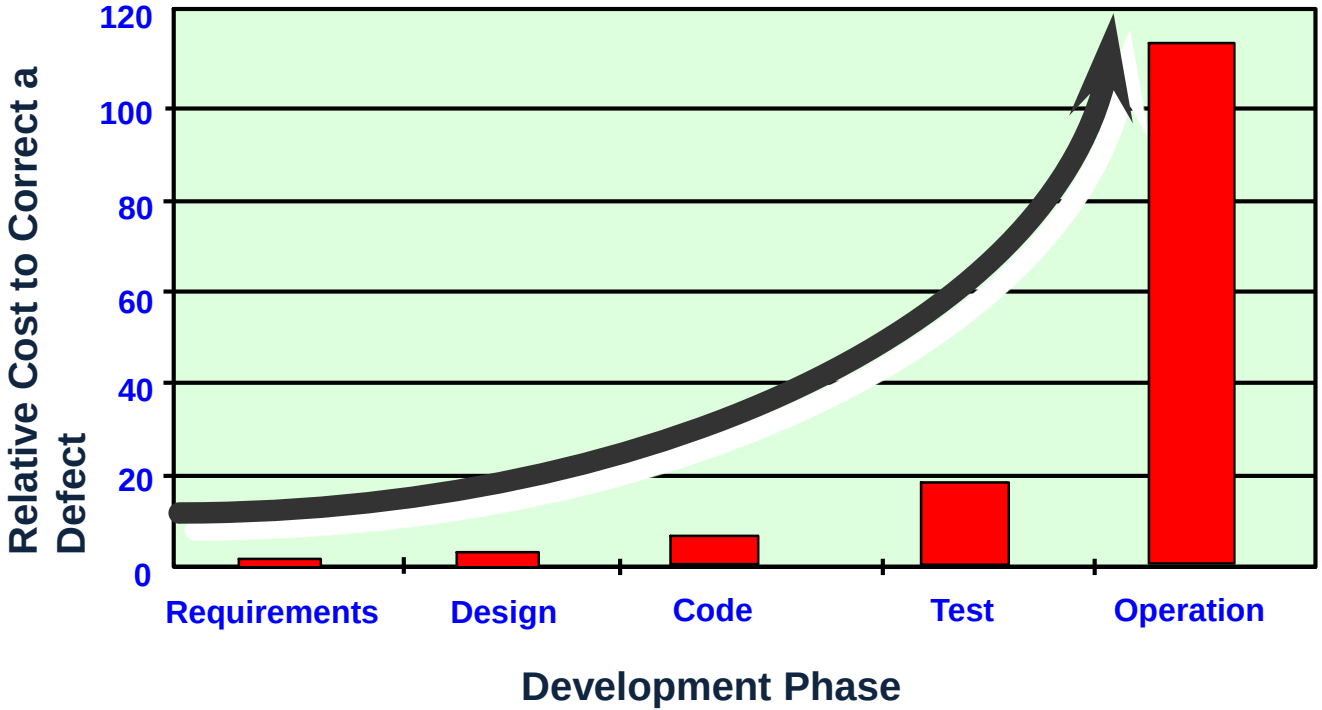
Rayleigh

- A conceptual tool used in physics, chemistry, and engineering. This form of dimensional analysis expresses a functional relationship of some variables in the form of an exponential equation. It was named after Lord Rayleigh

Root Cause Analysis

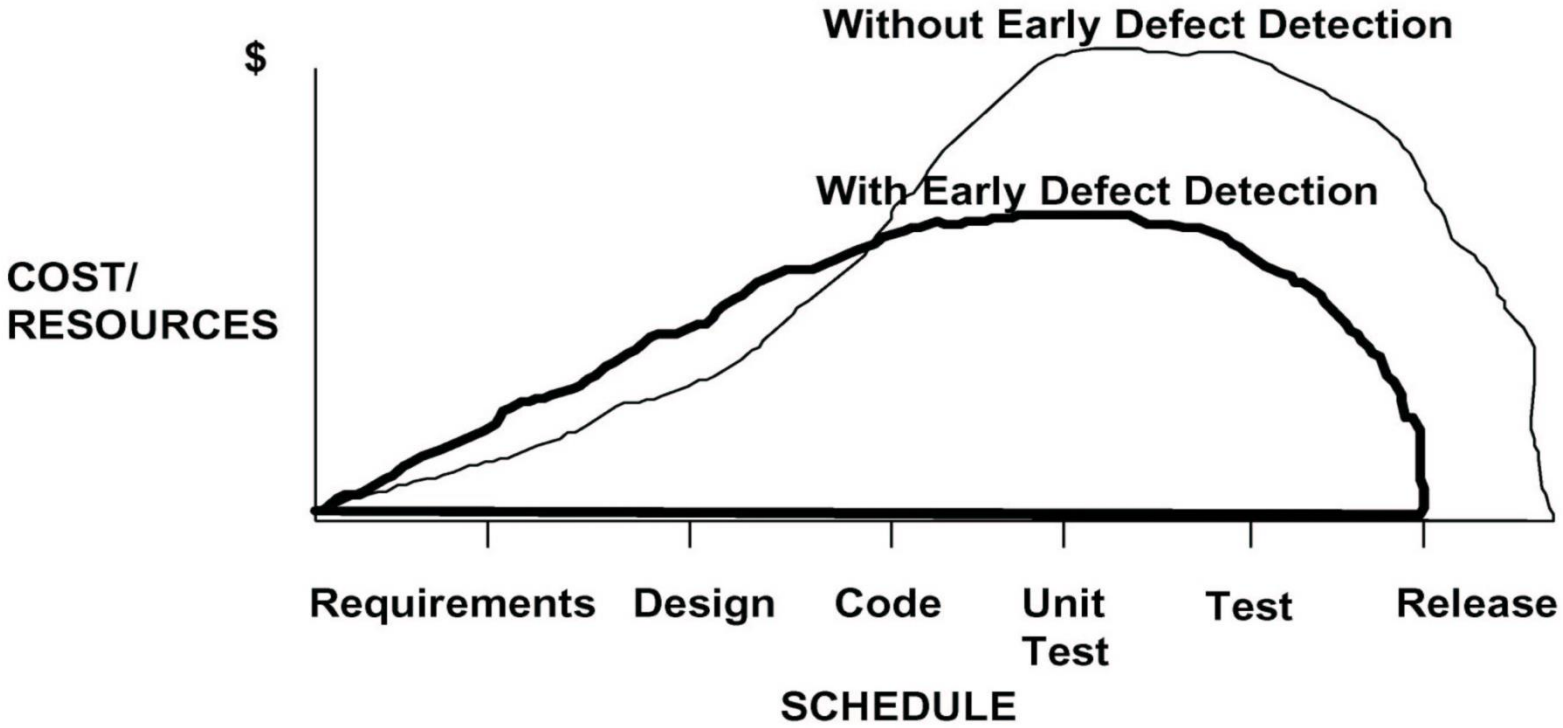
- Pareto Chart
- Fish Bone Diagram (Cause–and–Effect Diagram/Ishikawa Diagram)

Relative Cost to Fix a Defect



[from Robert Grady, “Applications of Software Measurement Conference,” 1999]

Early Defect Detection



•Adapted from Fagan, "Advances in Software Inspections", IEEE, July 1986

Defect Prevention

The most effective way to manage defects is to prevent their initial introduction.

How to do defect prevention (Open discussion)

SW Quality Model

SW Quality Concept

CMMI 1.3

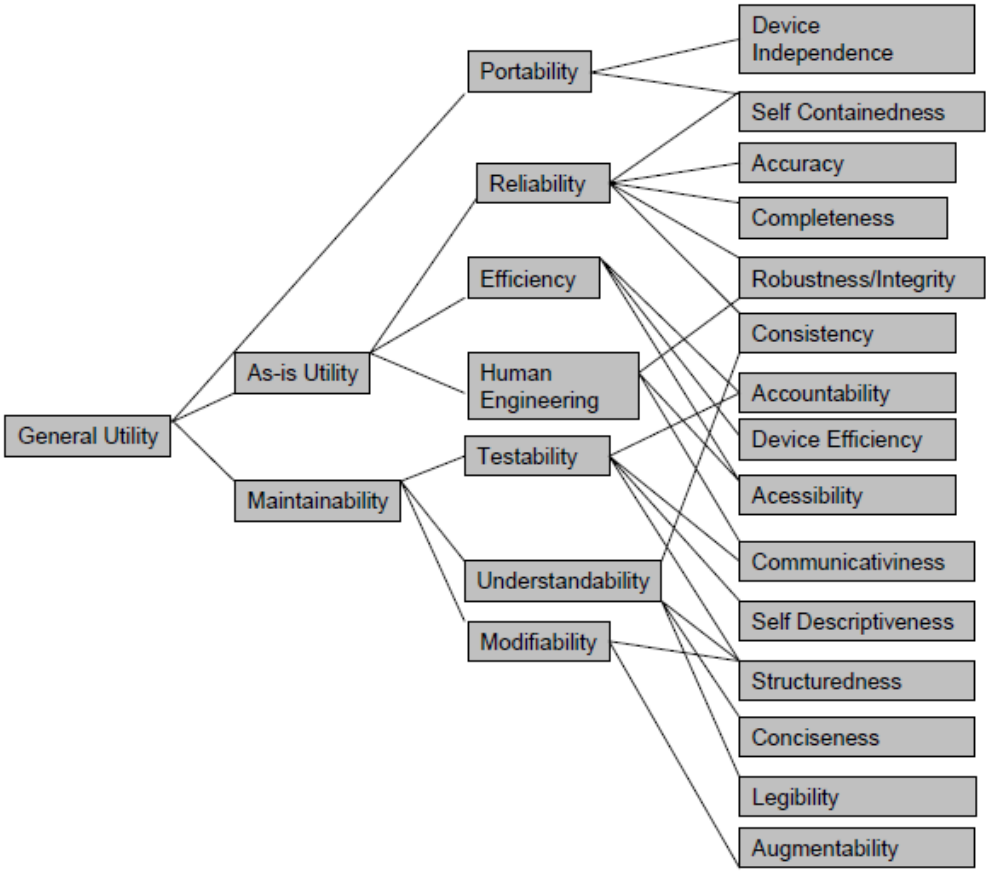
- The degree to which a set of inherent characteristics fulfills requirements.

ISO 9000

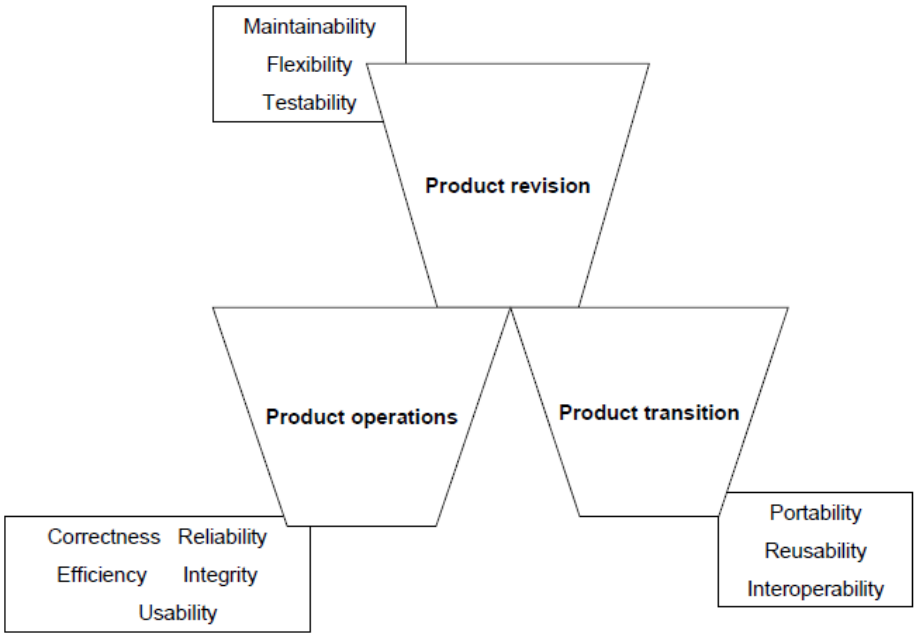
Degree to which a set of inherent characteristics (3.5.1) fulfils requirements (3.1.2)

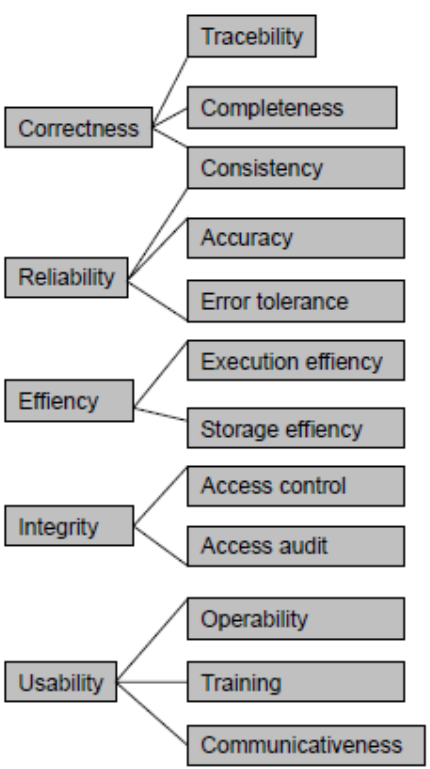
- NOTE 1 The term “quality” can be used with adjectives such as poor, good or excellent.
- NOTE 2 “Inherent”, as opposed to “assigned”, means existing in something, especially as a permanent characteristic.
  - Requirement
    - need or expectation that is stated, generally implied or obligatory
    - NOTE 1 “Generally implied” means that it is custom or common practice for the organization, its customers and other interested parties , that the need or expectation under consideration is implied.
    - NOTE 2 A qualifier can be used to denote a specific type of requirement, e.g. product requirement, quality management requirement, customer requirement.
    - NOTE 3 A specified requirement is one that is stated, for example in a document (3.7.2).
    - NOTE 4 Requirements can be generated by different interested parties (3.3.7).
    - NOTE 5 This definition differs from that provided in 3.12.1 of ISO/IEC Directives, Part 2:2004.

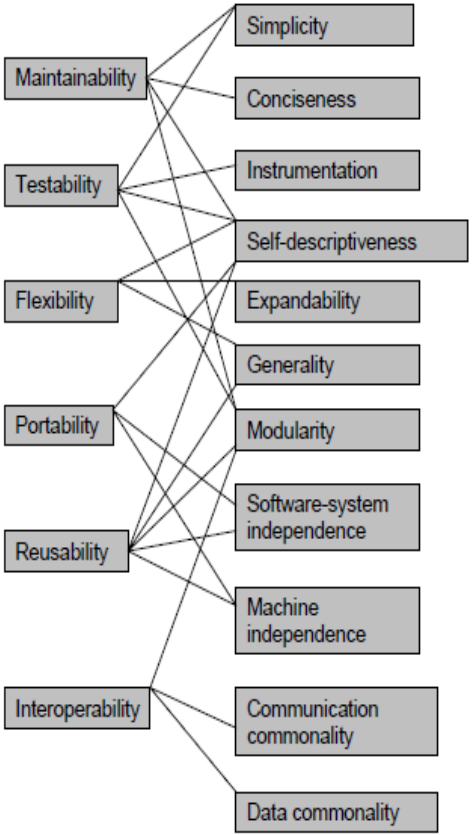
Boehm Quality Model



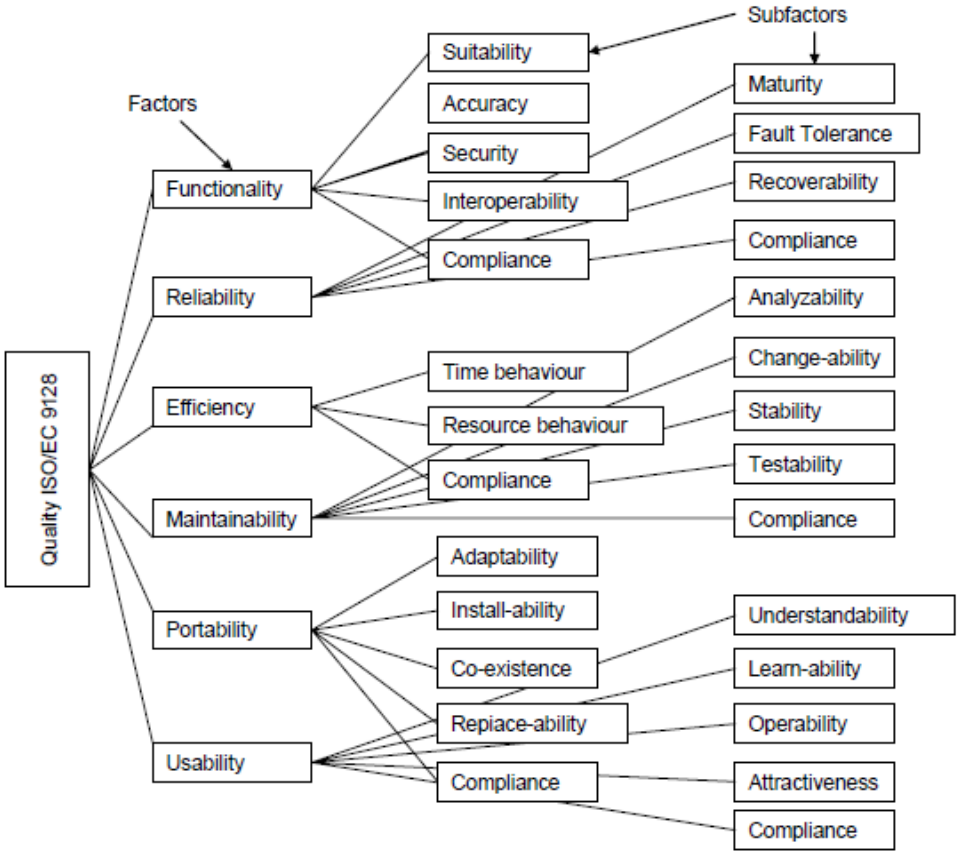
McCall Quality Model







ISO 9126 Quality Model



First published in 1991, revised in 2001, Superseded by ISO/IEC 25010 in 2011)

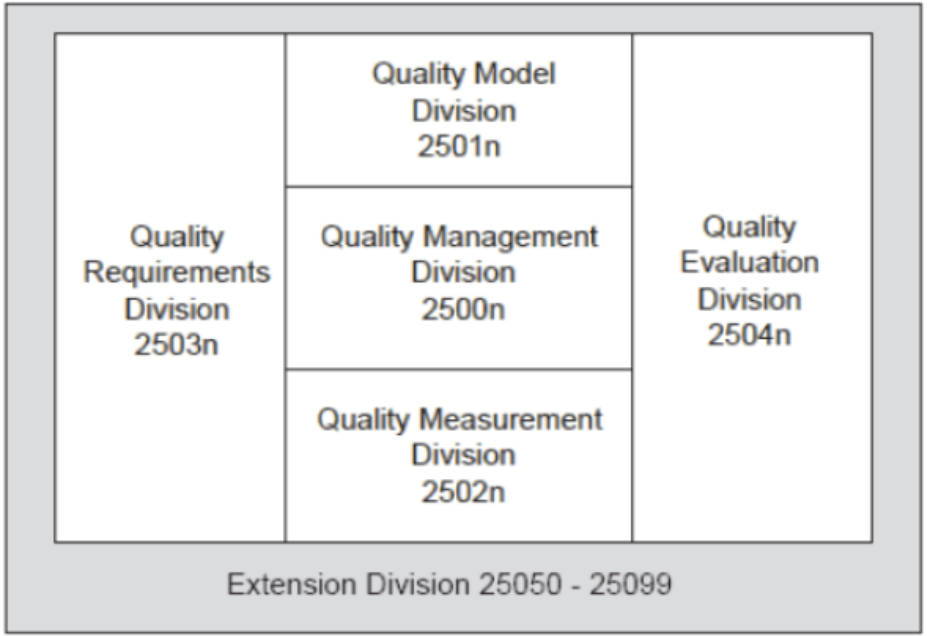
Quality Model Comparison

Criteria/goals	*McCall, 1977*	*Boehm, 1978*	*ISO 9126, 1993*
Correctness	*	*	maintainability
Reliability	*	*	*
Integrity	*	*	
Usability	*	*	*
Effiency	*	*	*
Maintainability	*	*	*
Testability	*		maintainability
Interoperability	*		
Flexibility	*	*	
Reusability	*	*	
Portability	*	*	*
Clarity		*	
Modifiability		*	maintainability
Documentation		*	
Resilience		*	
Understandability		*	
Validity		*	maintainability
Functionality			*
Generality		*	
Economy		*	

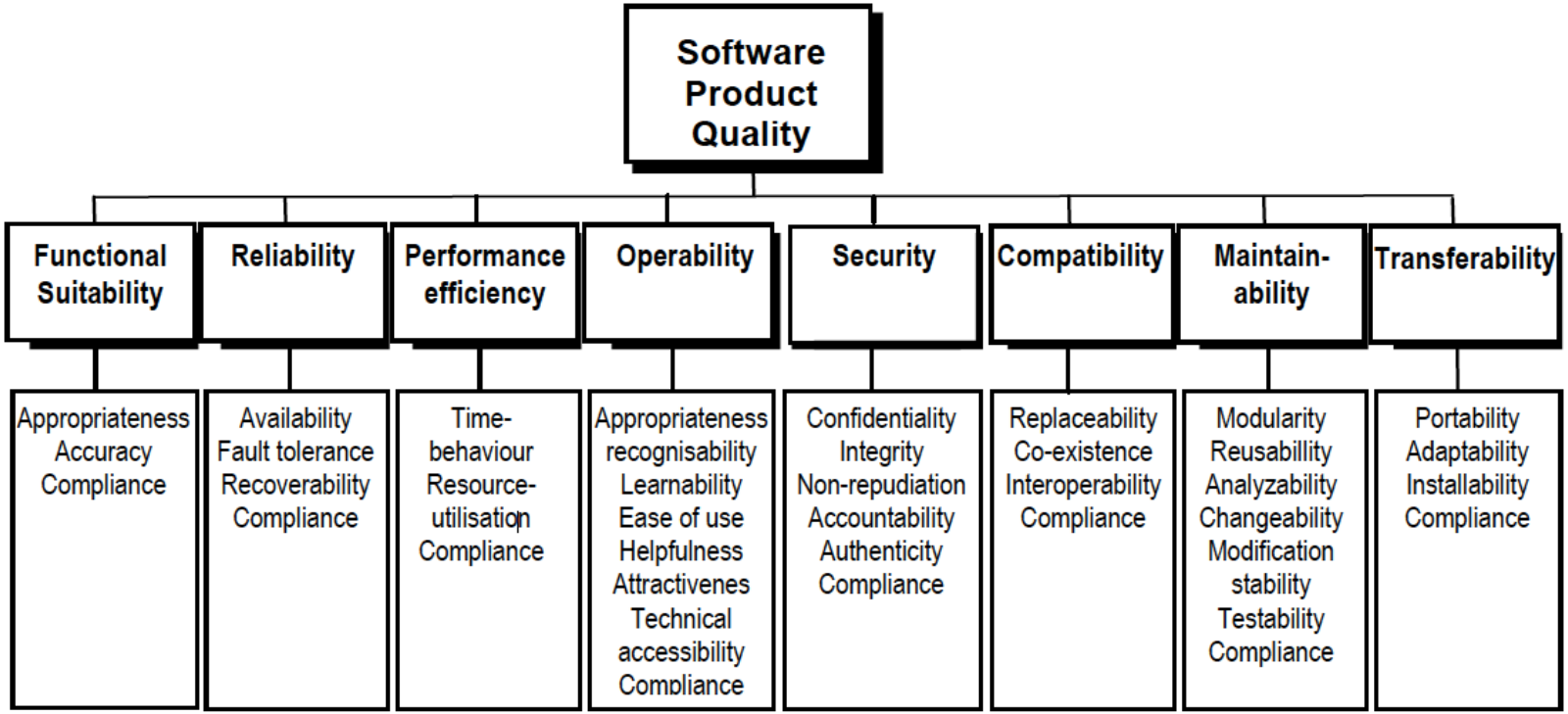
ISO/IEC 25010 Quality Model

Differentiates between

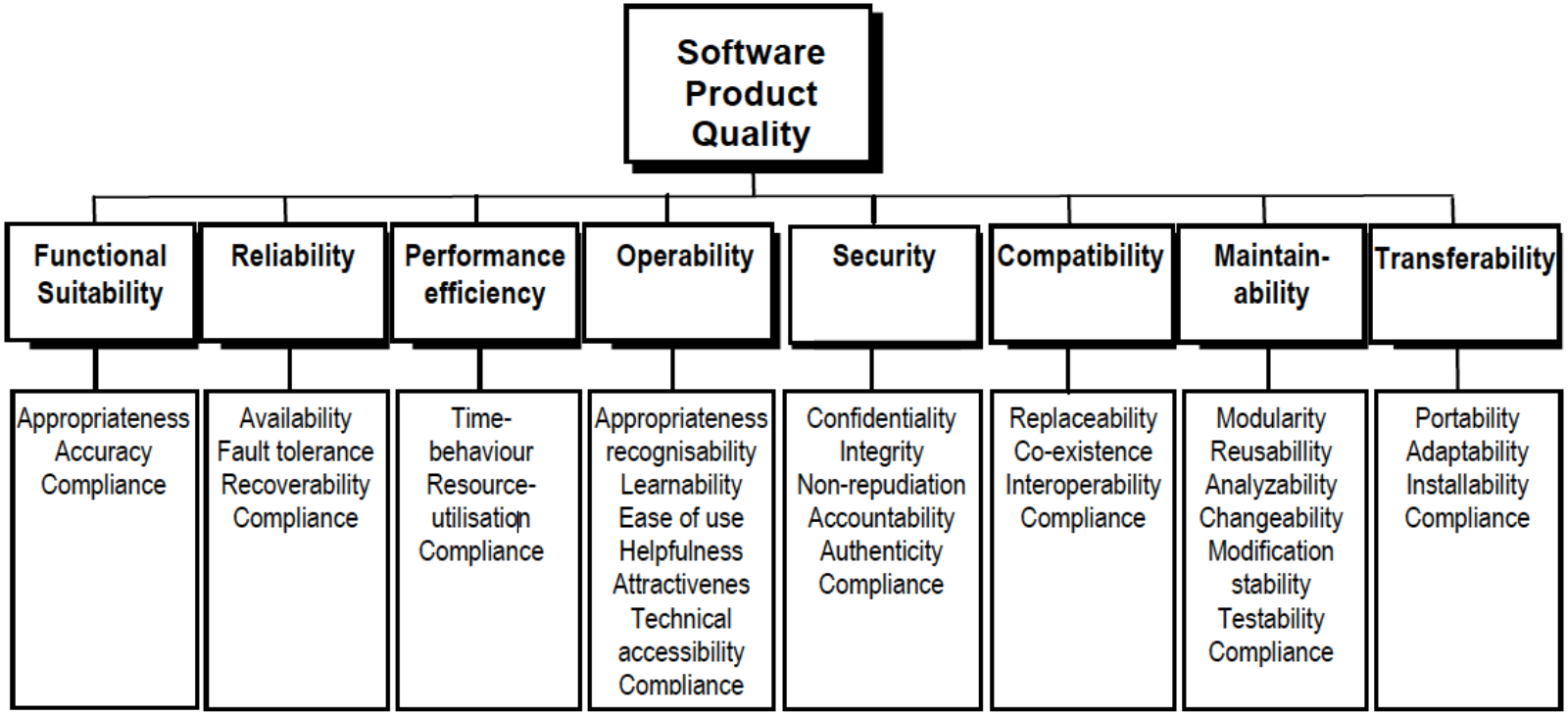
- Product quality model
- Quality in use model



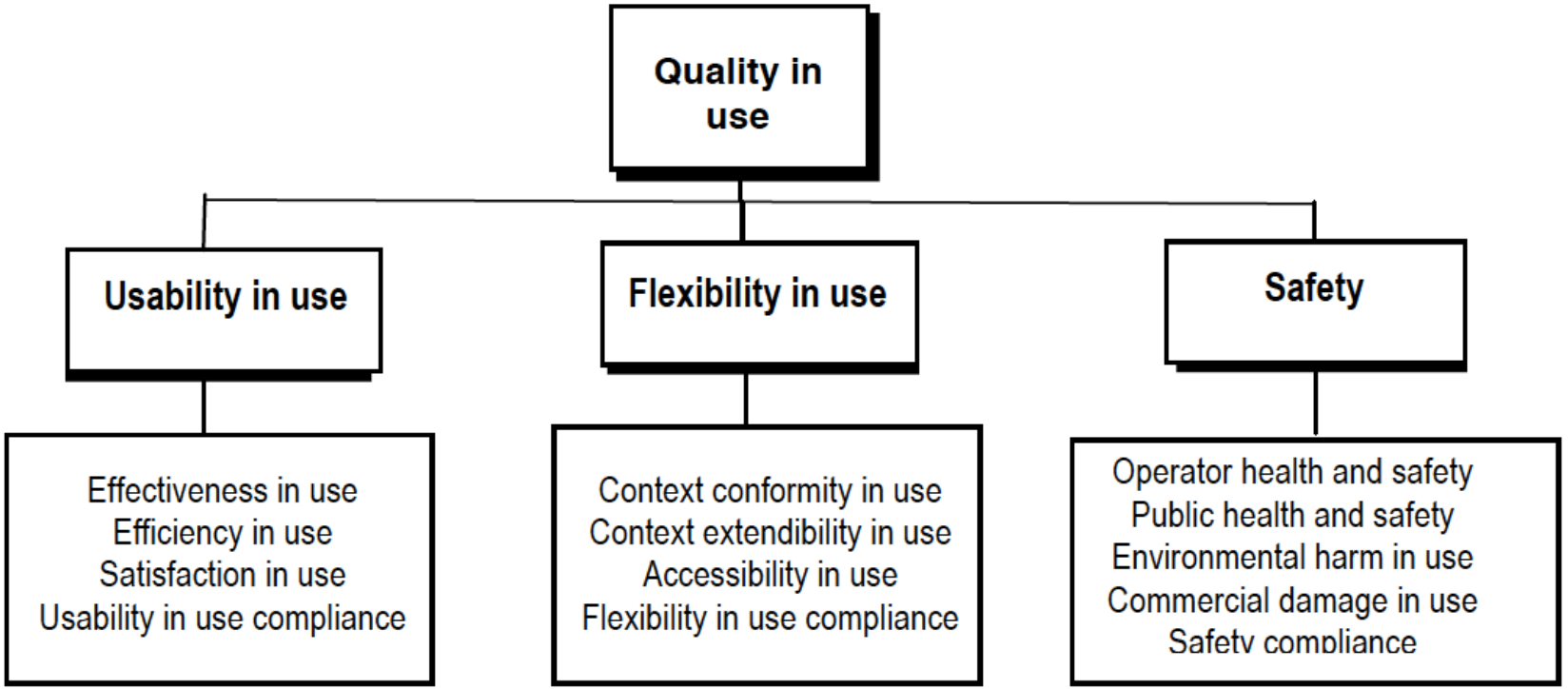
ISO/IEC 25010-Product Quality Model



ISO/IEC 25010-Product Quality Model



ISO/IEC 25010 Quality in Use Model



Chapter No. 5 -- Process Audit

Software Audit Definition

IEEE Standard states

- Audit is an **independent evaluation** of software products or processes to ascertain compliance to standards, specifications, and procedures based on objective criteria that includes document that specify:
  - The form or content of the product to be produced
  - The process by which the products shall be produced
  - How compliance to standards or guideline shall be measured

What are Audits used for

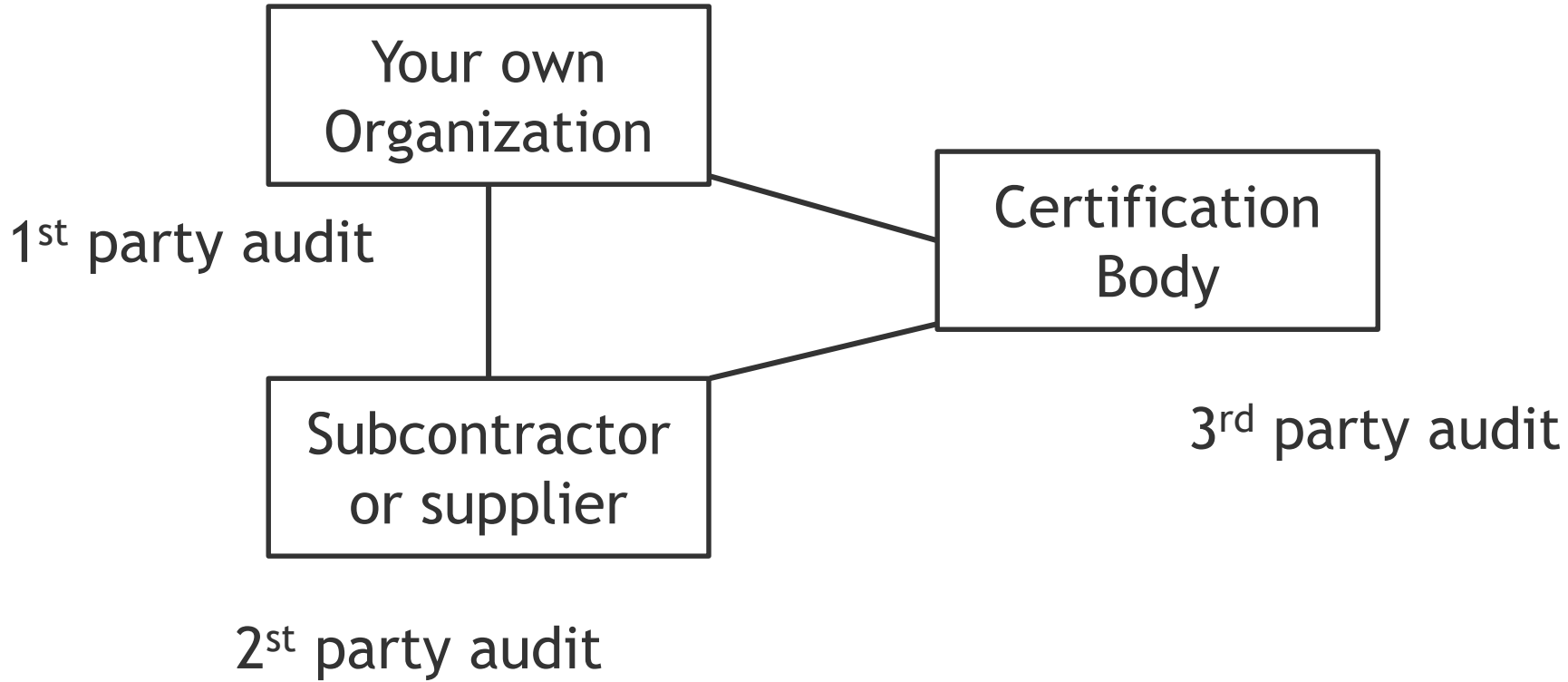
- To look at the overall process
- To audit conformity
- To audit a supplier / Subcontractors for contract reasons
- To verify that a company’s quality system has been implemented
- To determine if an organization is compliant with an outside standard
- To investigating problems
- Way of improvement

Audit Program Objectives

Auditing Objective should based on,

- Management priorities
- Commercial intentions
- Management system requirements
- Regulatory requirement
- Supplier evaluation
- Customer requirement
- Potential business risks

Types of Audit



Audit Types

- Internal audit
- External audit
- System audit
- Process audit
- Product audit
- Compliance audit
- Regulation audit
- Etc….

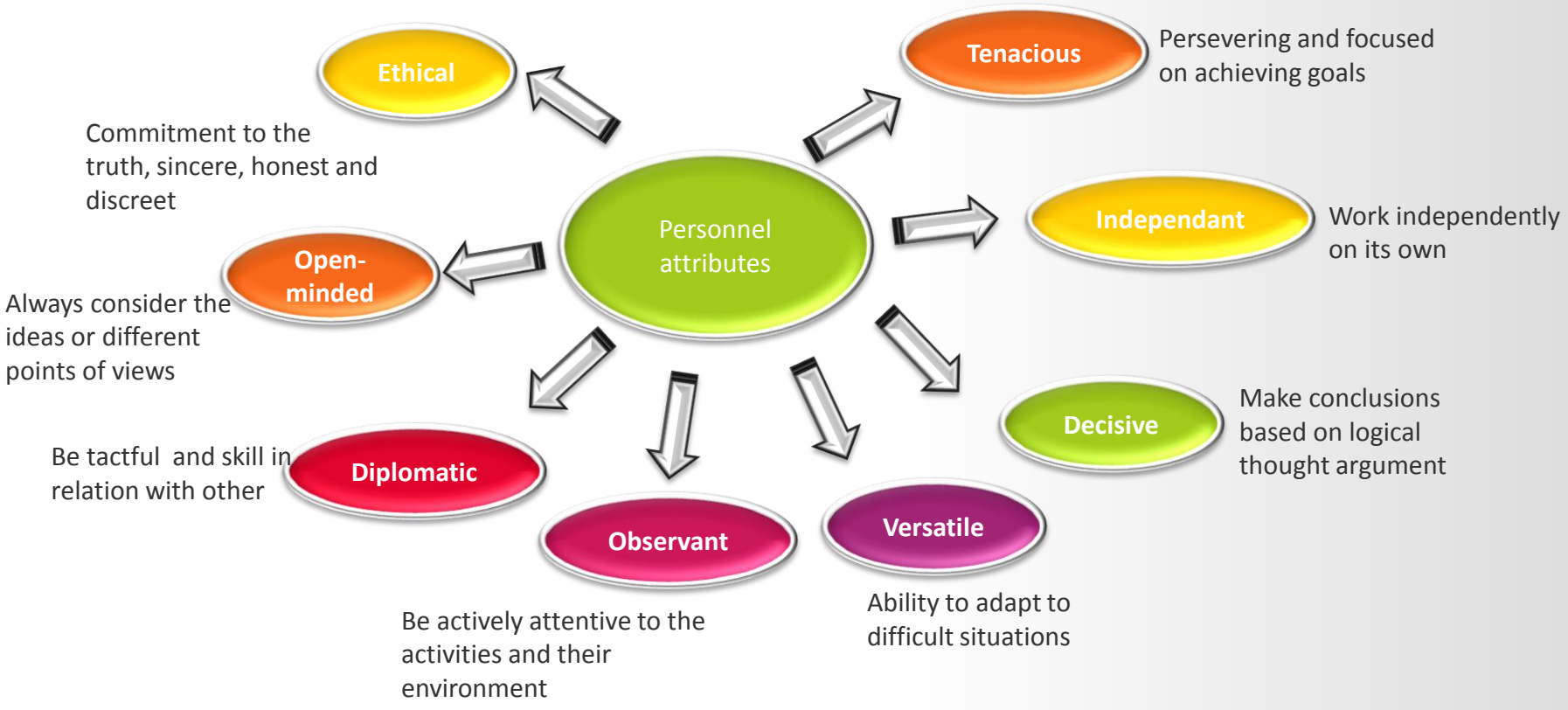
Common Definition relating to audit

- Audit
- Auditor, Leader auditor - The person with the competence to conduct an audit
- Auditee - The person /organization being audited
- Audit Criteria
- Audit Evidence
- Audit Conclusion
- Audit findings – Nonconformity
  - Critical Nonconformity - Relate to a customer safety issue, a production stop that affect deliveries, or a high return rate (in general this means when the failure rate > = to 5 % for one issue noticed)
  - Major Nonconformity -Relates to the absence, or total failure, of a process to meeting the requirements
  - Minor Nonconformity - Relates to a failure to meet one requirement of standard, or a single observed lapse in implementing one requirement of company quality management system procedures

Audit Responsibilities

Roles	Responsibilities
Lead Auditor	• Developing and gaining audit authority approval of the audit schedule • Planning the detail of the forthcoming audit • Planning and preparing the audit team safety needs • Conducting the audit • Acting as the audit team leader • Resolving conflicts • Formally report the audit result • Following up on actions taken
Auditor	• Comply with applicable audit requirements • Communicating and clarifying audit requirements • Planning and carrying out audit • Documenting audit observations and report audit result • Verify effectiveness of corrective actions
Auditee	• Cooperating with the auditors • Providing access to facilities and materials • Providing resourcing needed by audit team • Determining and initialing corrective actions

Auditor Competencies



Audit Approach - Process Oriented

What is a process:

- Sequence of interdependent and linked procedures which, at every stage, consume one or more resources (employee time, energy, machines, money) to convert inputs (data, material, parts, etc.) into outputs. These outputs then serve as inputs for the next stage until a known goal or end result is reached.(ISO 9000:2005)
- Looking at processes in term of proactive thinking in risk management

The process approach emphasize the importance of:

- Understanding and meeting the requirements of QMS
- Looking at processes in term of proactive thinking in risk management
- Looking at processes in terms of business added value
- Obtaining results of process performance
- Continual improvement of process

Process Oriented Audit - PDCA (Plan-Do-Check-Act)

The PDCA methodology has to be applied to all the processes

- Activities
- Controls
- Documentation
- Policy
- Objectives

- Deploy and conform with plan

- Measure and monitor for conformity and effectiveness

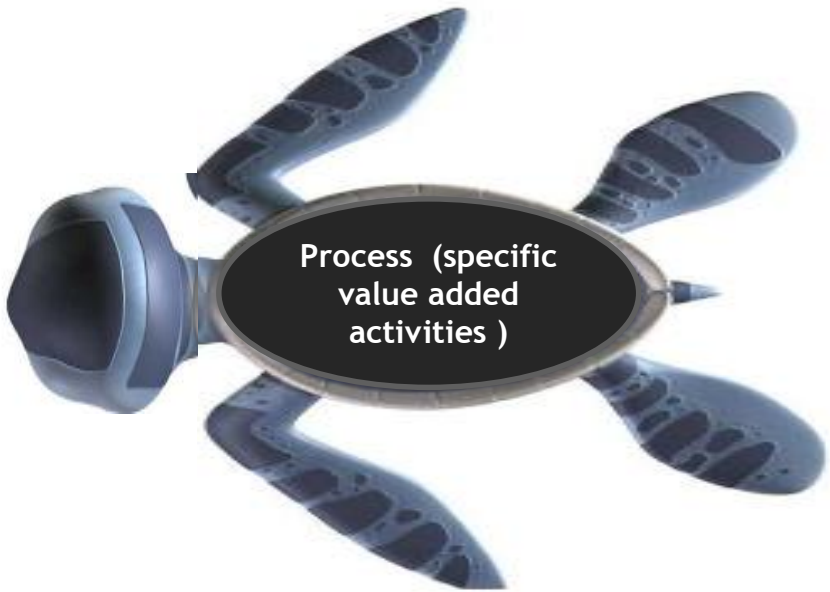
- Analyze Review
- Decide/change
- Improve effectiveness

technicolor

Process Oriented Audit - turtle diagram

With What resources?
- Technical
- Financials

With Who ?
- Personnel



Inputs
- From whom?
- Where ?

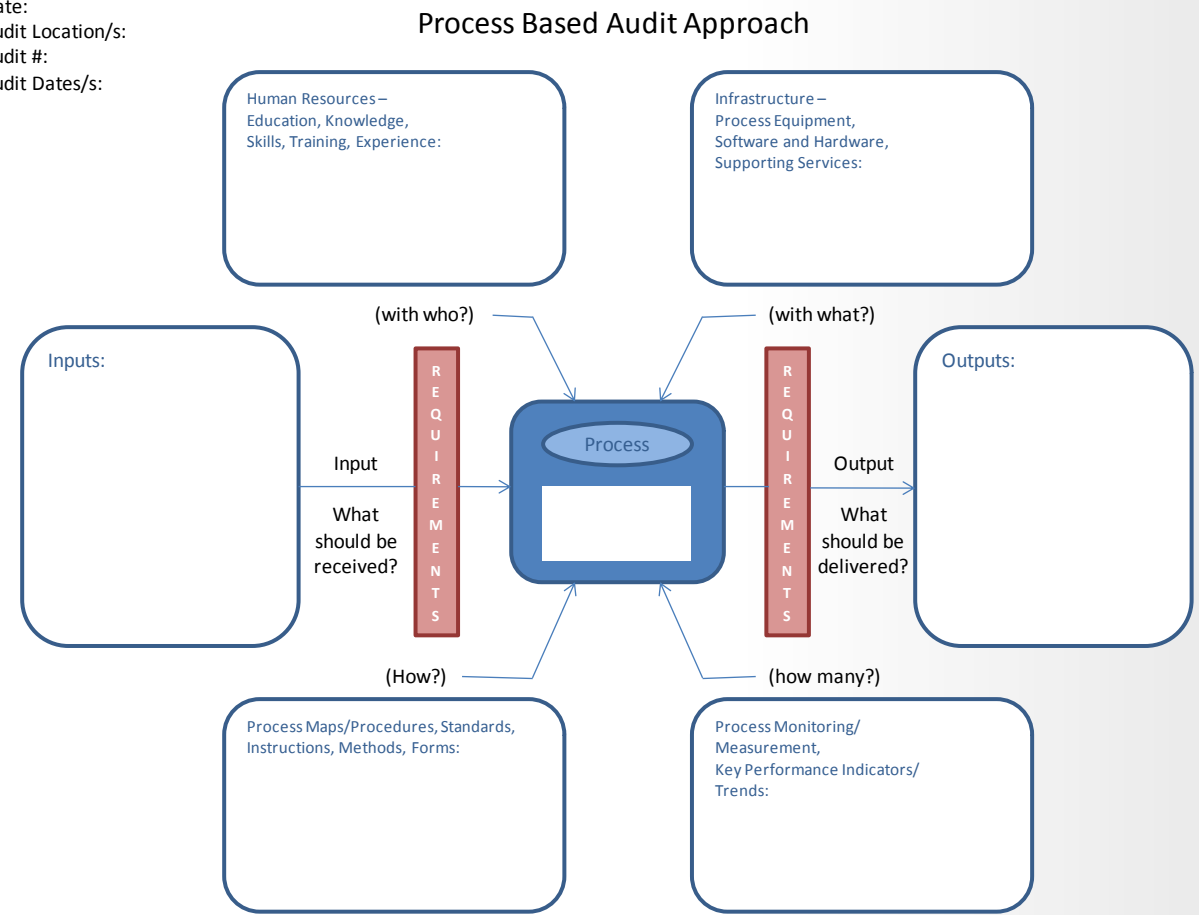
Outputs
- To whom?
- Where?

How done?
- Instructions
- Procedures
- Methods

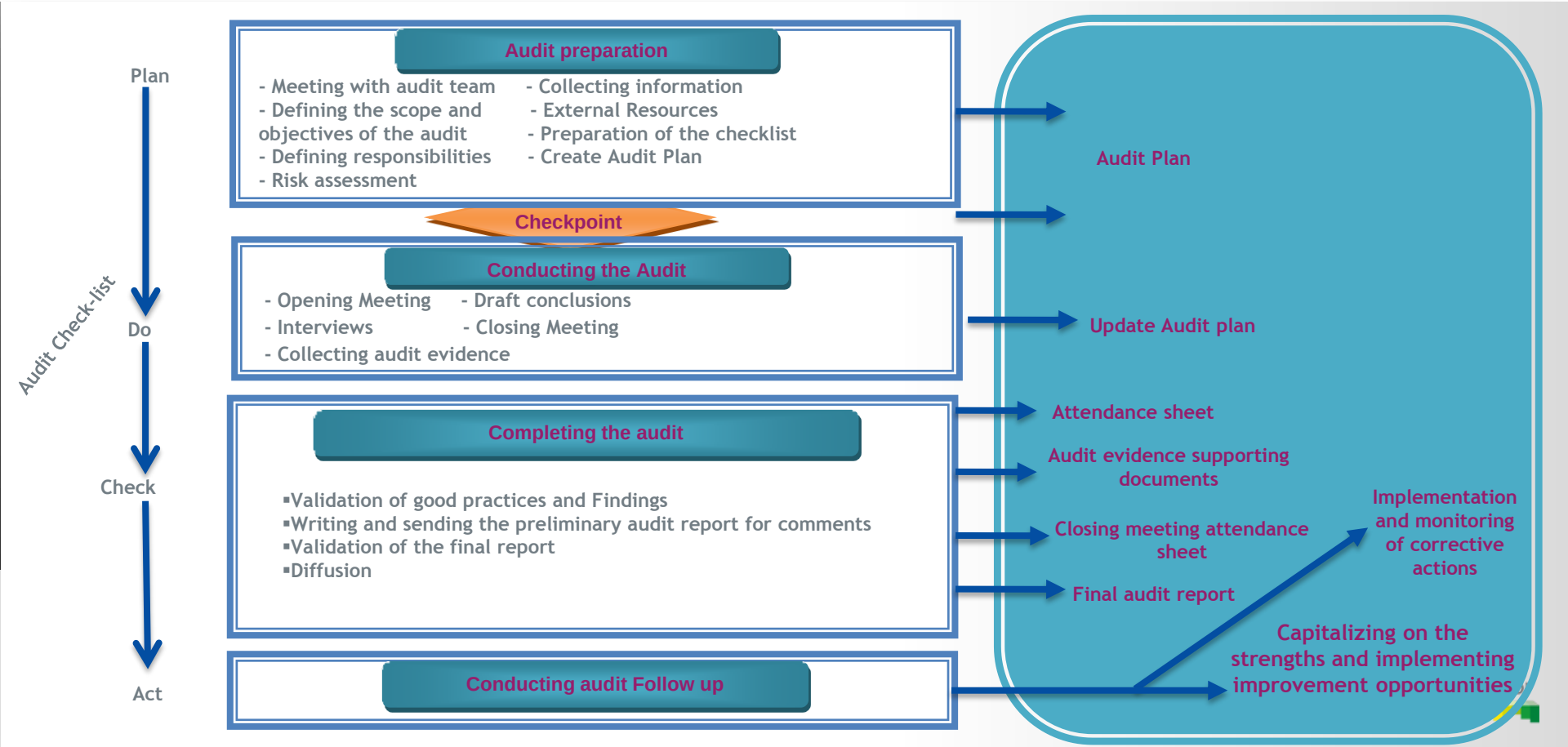
What results?
- KPI's

Process Oriented Audit - turtle diagram

Date:
Audit Location/s:
Audit #:
Audit Dates/s:



Software Audit Process – Process Map



Software Audit Process - Preparation

Initiating the audit includes

- Determining the feasibility of the audit
  - Check the availability of resources: human, technical, financial and logistical
  - Determine the duration: Based on the experience and feedback

Audit Planning What needs to be done

- Determine the objective of the audit
- Identify specified requirements
- Determine audit duration
- Contact the auditee – agree on the dates
- Define the audit plan
- Review with audit team
- Prepare work document

Software Audit Process - Preparation

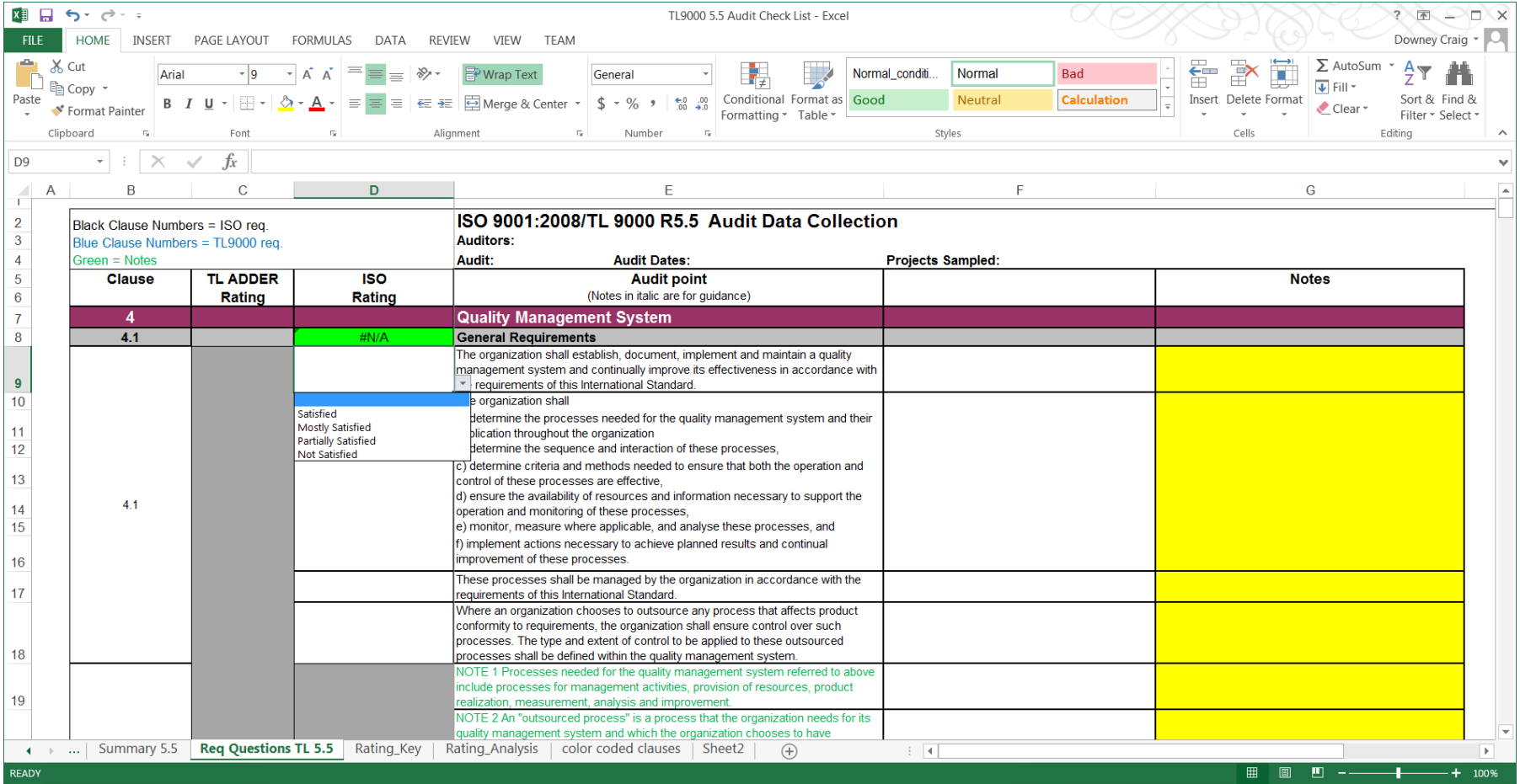
Prepare work documents

- Use as a reference and for recording audit proceedings
- Include checklists based on applicable standard, sampling, plans and forms
- Keep checklist flexible to allow changes resulting from information collected during document review
- Safeguard any confidential and proprietary information
- Retain work documents as reference until corrective actions are closed.

Checklists preparation: One approach is to :

- Identify audit scope and process(es) within the scope
- Identify applicable factors (Inputs, outputs, KPI's, resources and Activities…)
- Use these points and other requirements (ISO 9001: 2008, TL 9000 R5, system documentation etc…) to Plan what to look at and Plan what to look for (audit evidence)

Check List


TL9000 5.5 Audit Check List - Excel
FILE
HOME
INSERT
PAGE LAYOUT
FORMULAS
DATA
REVIEW
VIEW
TEAM
Downey Craig
Paste
Cut
Copy
Format Painter
Arial
9
Wrap Text
Merge & Center
General
Conditional Formatting
Format as Table
Normal_conditi...
Normal
Bad
Good
Neutral
Calculation
Insert
Delete
Format
AutoSum
Fill
Clear
Sort & Filter
Find & Select
Clipboard
Font
Alignment
Number
Styles
Cells
Editing
D9
Black Clause Numbers = ISO req.
Blue Clause Numbers = TL9000 req.
Green = Notes
ISO 9001:2008/TL 9000 R5.5 Audit Data Collection
Auditors:
Audit:
Audit Dates:
Projects Sampled:
Clause
TL ADDER Rating
ISO Rating
Audit point
(Notes in italic are for guidance)
Notes
4
Quality Management System
4.1
#N/A
General Requirements
The organization shall establish, document, implement and maintain a quality management system and continually improve its effectiveness in accordance with requirements of this International Standard.
Satisfied
Mostly Satisfied
Partially Satisfied
Not Satisfied
e organization shall
determine the processes needed for the quality management system and their plication throughout the organization
determine the sequence and interaction of these processes,
c) determine criteria and methods needed to ensure that both the operation and control of these processes are effective,
d) ensure the availability of resources and information necessary to support the operation and monitoring of these processes,
e) monitor, measure where applicable, and analyse these processes, and
f) implement actions necessary to achieve planned results and continual improvement of these processes.
4.1
These processes shall be managed by the organization in accordance with the requirements of this International Standard.
Where an organization chooses to outsource any process that affects product conformity to requirements, the organization shall ensure control over such processes. The type and extent of control to be applied to these outsourced processes shall be defined within the quality management system.
NOTE 1 Processes needed for the quality management system referred to above include processes for management activities, provision of resources, product realization, measurement, analysis and improvement.
NOTE 2 An "outsourced process" is a process that the organization needs for its quality management system and which the organization chooses to have
Summary 5.5
Req Questions TL 5.5
Rating_Key
Rating_Analysis
color coded clauses
Sheet2
READY
100%

Practice

Audit Plan

Software Audit Process – Conducting the Audit

What needs to be done:

- Conducting opening meeting (lead)
- Communicate during the audit
- Explain roles and responsibilities of participants
- Conducting the Interviews
- Collect and verify information
- Generate audit findings
- Prepare audit conclusions (lead)
- Conduct closing meeting (lead)

Software Audit Process – Conducting the Audit

Opening Meeting: Who attend?

- The Lead Auditor or his delegate
- The auditors responsible for the audit
- Management of entity (ies) audited (s)
- may include the entire staff or entity (ies) audited (s) or process owners

Opening meeting Agenda

- Introduction of audit team and their roles
- Confirm the purpose and scope of the audit and the standards to be used
- Explain or confirm the audit plan, availability of key staff, and reporting and reporting procedure
- Explain nonconformity and observation definitions
- Confirm confidentiality and that audit is based on a limited sample

Software Audit Process – Conducting the Audit

Goods Practices for Opening Meeting

- Prepare properly
- Ensure meeting is held in a suitable place
- Have a agenda
- Use time carefully (30mins)
- Explain exactly what is to happen
- Create a positive impression

Software Audit Process – Conducting the Audit

Interview process

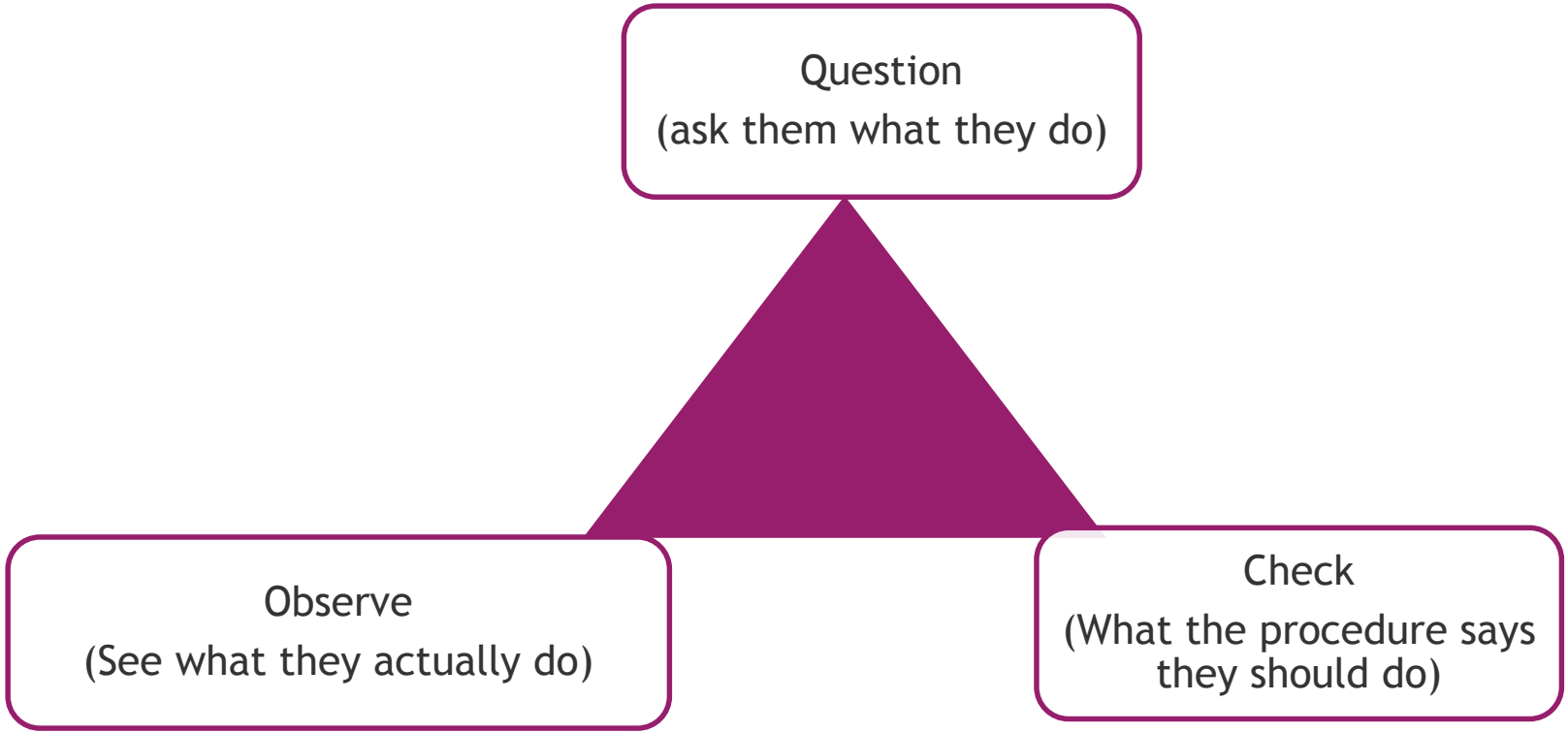
- Identify individual to be audited
- Introduce yourself
- Explain why you are there
- Ask Open questions
- Ask detailed questions
- Do ‘show me please’
- Check facts and make notes
- Thank auditee

Software Audit Process – Conducting the Audit

Suggest Checkpoint in Audit

- The purpose, input, output, controls and resources applicable to each process are clear
- Links are established between process and high level and local objective
- The output of the process are compared with desired outcomes, the purpose of the process
- The steps in the process and associated and responsibilityes re determined,
- Inter-relating processes are identified
- Process measurement are identified
- Evidence of continual improvement is sought
- Needs of internal and external customer are clear

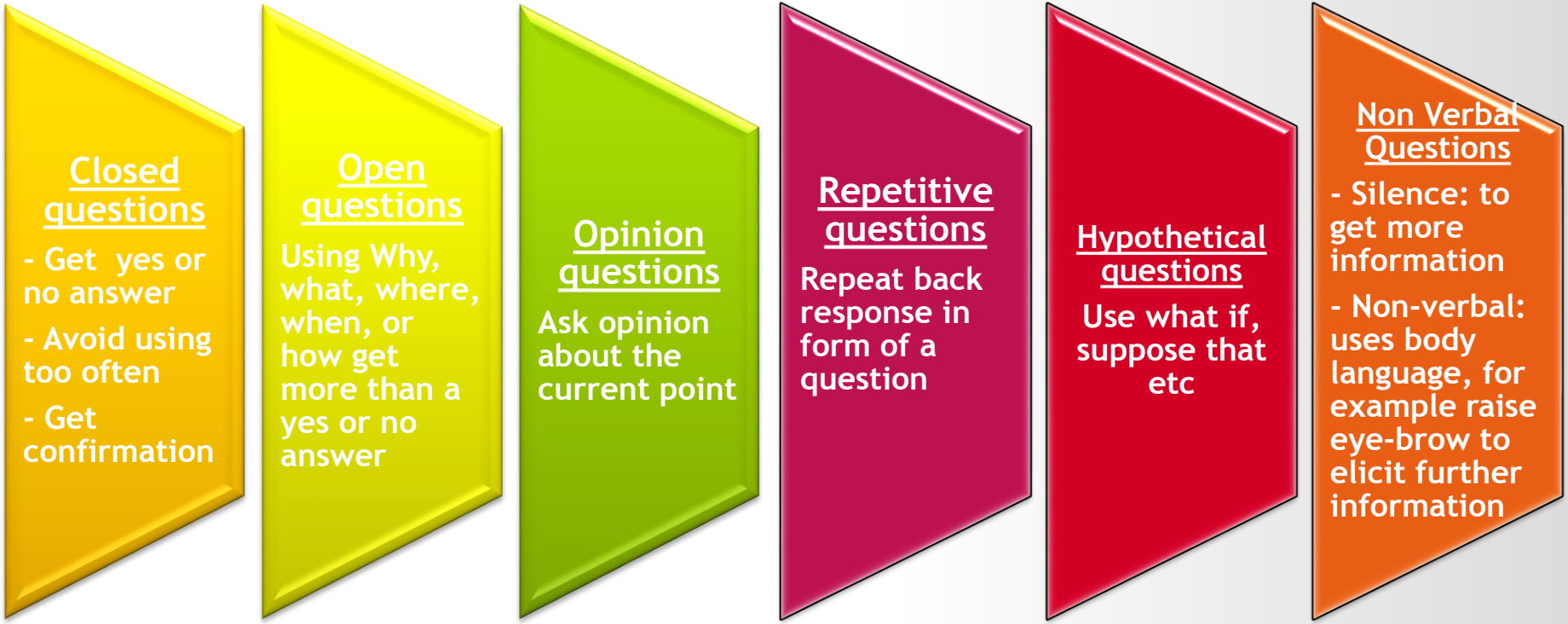
Software Audit Process – Conducting the Audit



Audit Triangle

Software Audit Process – Conducting the Audit

Questioning techniques



Software Audit Process – Conducting the Audit

- Sampling for auditors
  - Audit base on sampling basis
  - Sampling cannot be statistically significant
  - Follow the audit trail, using the example to provide you with evidence of conformance of the standard

Software Audit Process – Conducting the Audit

Generate audits findings:

- Evaluate audit evidence against Audit criteria to generate audit findings
- Indicate if findings are conformities, non-conformities , areas of concerns or opportunities for improvement
- Meet with (audit team) to review the findings
- Specify (with supporting evidences) or summarize conformity by location, function or processes, as required by audit plan
- Record non-conformity findings and supporting evidence
- Obtain auditee acknowledgement of non-conformities for the accuracy and understandability
- Try and resolve differences of opinion
- keep a record of unresolved issues

Software Audit Process – Conducting the Audit

Agreeing Nonconformities

Can be two stages:

- Agree with Auditee the facts
- Agree facts as nonconformities

Can be done:

- When they are found
- At the end of audit
- At the regular review points

Non conformities must:

- Be factual / Objective
- Be clear and concise
- Give clause number of quality standard / Company document
- Define the evidence
- Be categorized
- Be accepted by auditees

Software Audit Process – Conducting the Audit

Preparing audit conclusions

- Audit team meets prior the closing meeting
- Plan for the closing meeting
- Review audit findings and other information
- Agree on audit conclusions
- Prepare the audit report and recommendations
- If included in the audit plan, discuss audit follow-up

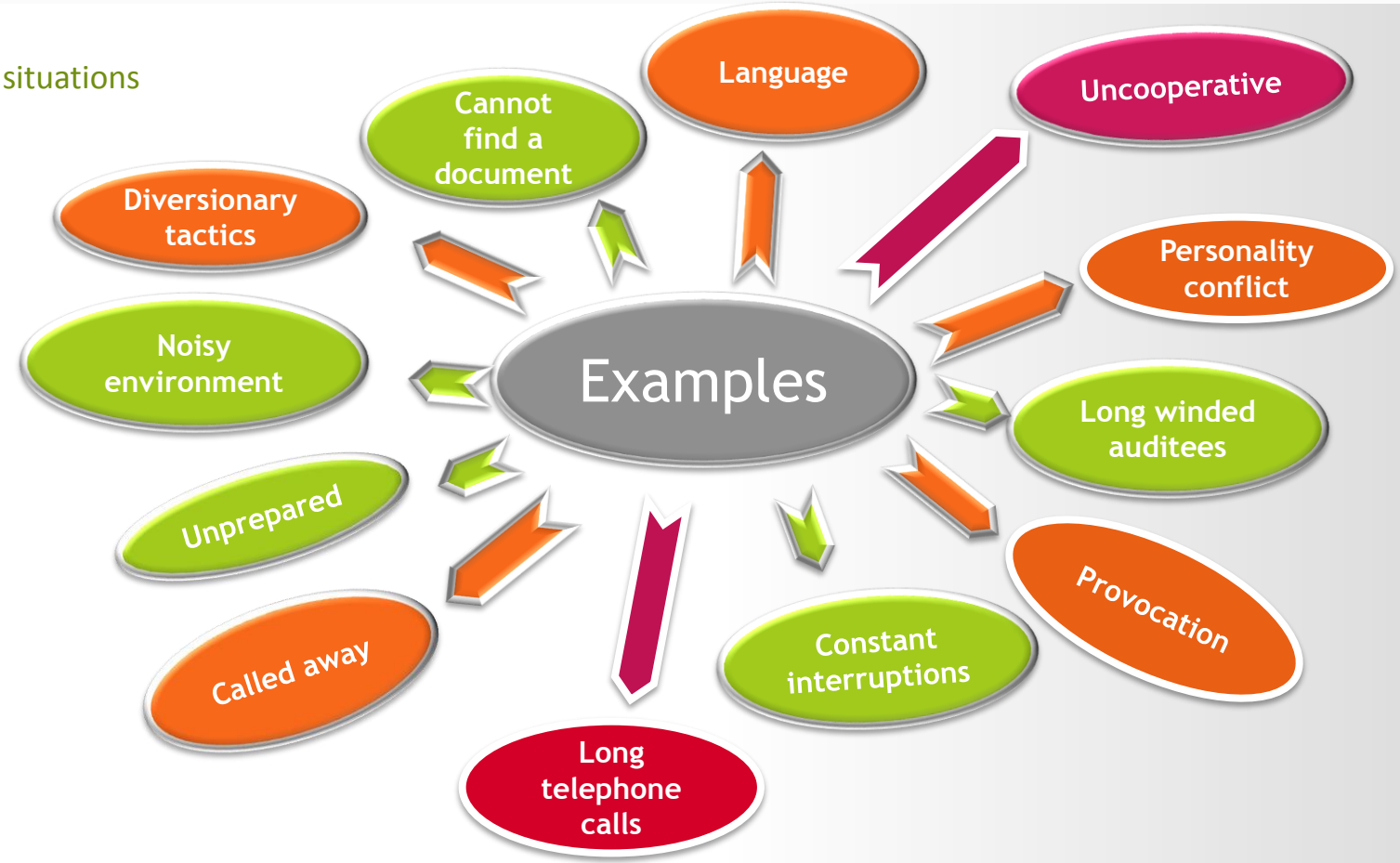
Software Audit Process – Conducting the Audit

Closing meeting

- Introduction
- Thanks
- Reconfirm the scope
- Comment the good point
- Summary of audit findings
- Submitting plan for corrective action
- Follow up action need and deadline

Software Audit Process – Conducting the Audit

Handling difficult situations



Software Audit Process – Completing the Audit

Prepare, Approve

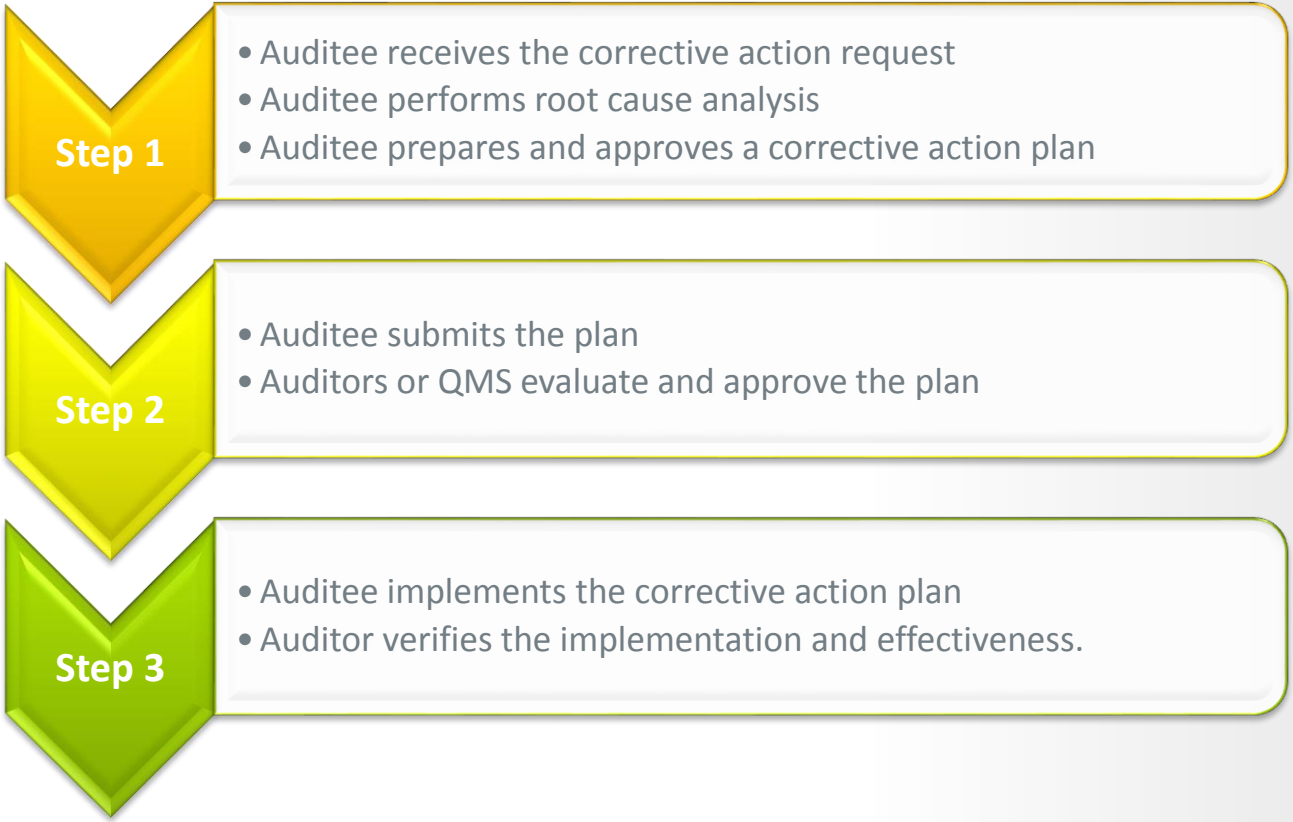
- Audit Report must include
  - Audit reference
  - Client and Auditee details
  - Audit team details
  - List of auditee representatives
  - Objectives, scope and criteria
  - Audit Plan- dates, places, areas audited and timing
  - Summary of audit process
  - audit summary of results including findings
  - Uncertainly due to sampling

Audit report distribution

- Audit Report has to be:
  - Issued within agreed time period
  - If delayed, provide reasons and agree on new issue date
  - Report must be dated, reviewed, and approved with team
  - Distributed to designated recipients
  - Report is a property of QMS
  - Recipients and audit team must respect the confidentiality of the report

Software Audit Process – Conducting Follow up

Corrective actions follow up



Practice

Process audit

- Opening Meeting
- Perform the audit
- Closing Meeting
- Corrective action follow up

Final report

以小组形式练习– Process audit

- Auditor小组提供给Auditee小组Check list
- Auditor小组提前review Auditee 小组相关材料
- 最后一次课现场评审process audit

- 课后各小组完成process audit report 以及corrective actions report

Thank you
Questions are welcome!