Continental

# SW Lifecycle @ IMM IAS

Presented by: Simona Grigoras

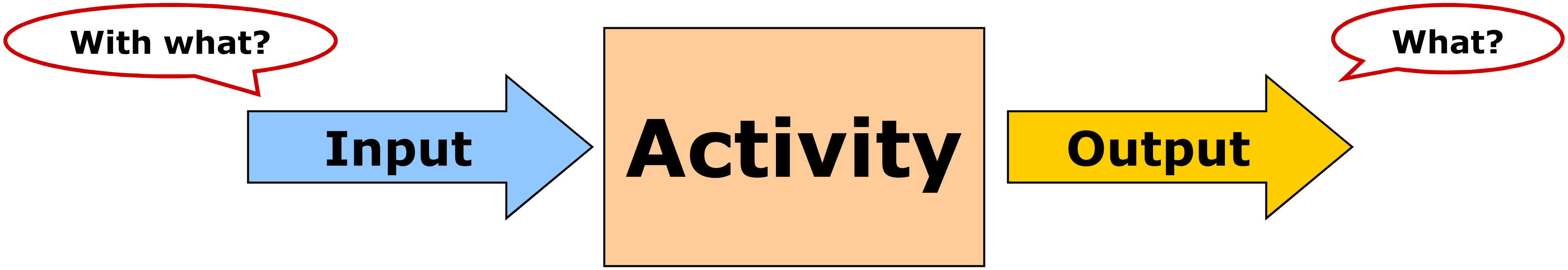
# Contents

- What is a process?
- Quality Management System - benefits
- Software Quality Assurance & Process Models
- PLC @ BU IMM & the V – cycle model
- Process assessment

# Processes – Definition, Measurement & Improvement

# What is a Process?

**Activity that transforms input into output**



With what?

Input

Activity

Output

What?

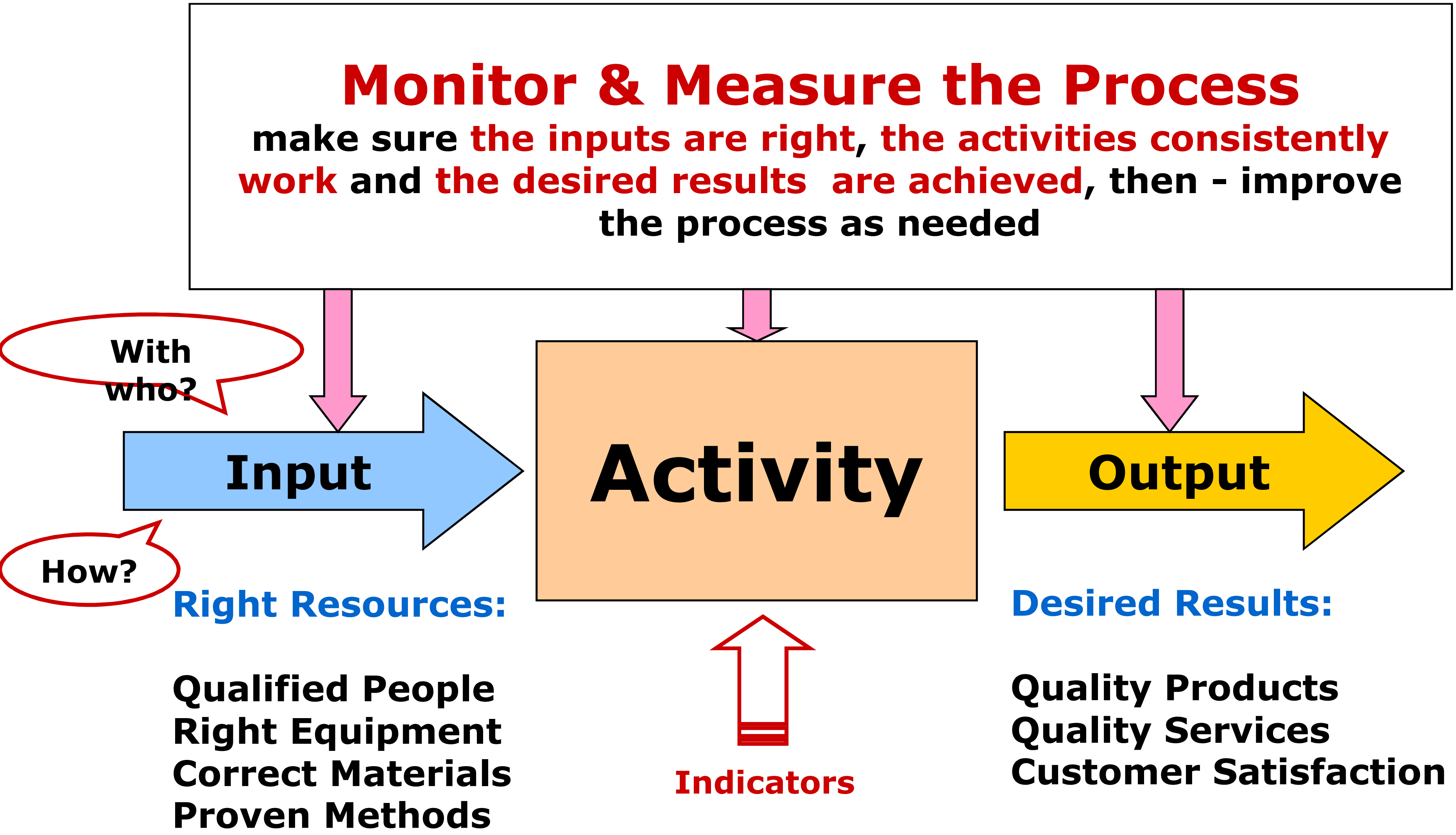
**Resources:**

**People**
**Equipment**
**Material**
**Methods**

**Results:**

**Products**
**Services**
**Performance**

# How is a Process Managed?



**Monitor & Measure the Process**

**make sure the inputs are right, the activities consistently work and the desired results  are achieved, then - improve the process as needed**

**With who?**

**Input**

**How?**

**Activity**

**Output**

**Right Resources:**

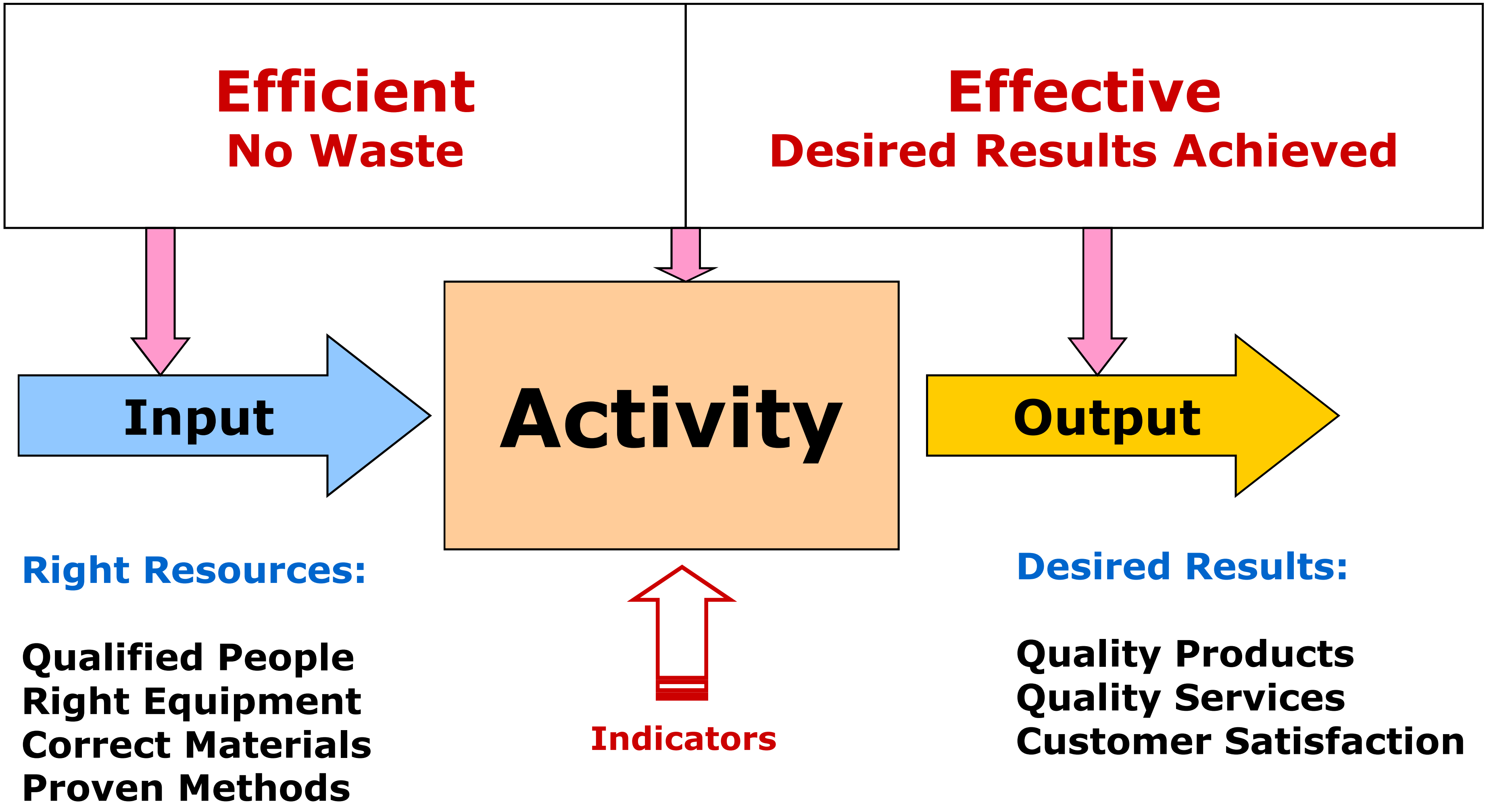
**Qualified People**
**Right Equipment**
**Correct Materials**
**Proven Methods**

**Indicators**

**Desired Results:**

**Quality Products**
**Quality Services**
**Customer Satisfaction**

# How is a Process Measured?



**Efficient**

**No Waste**

**Effective**

**Desired Results Achieved**

**Input**

**Activity**

**Output**

**Right Resources:**

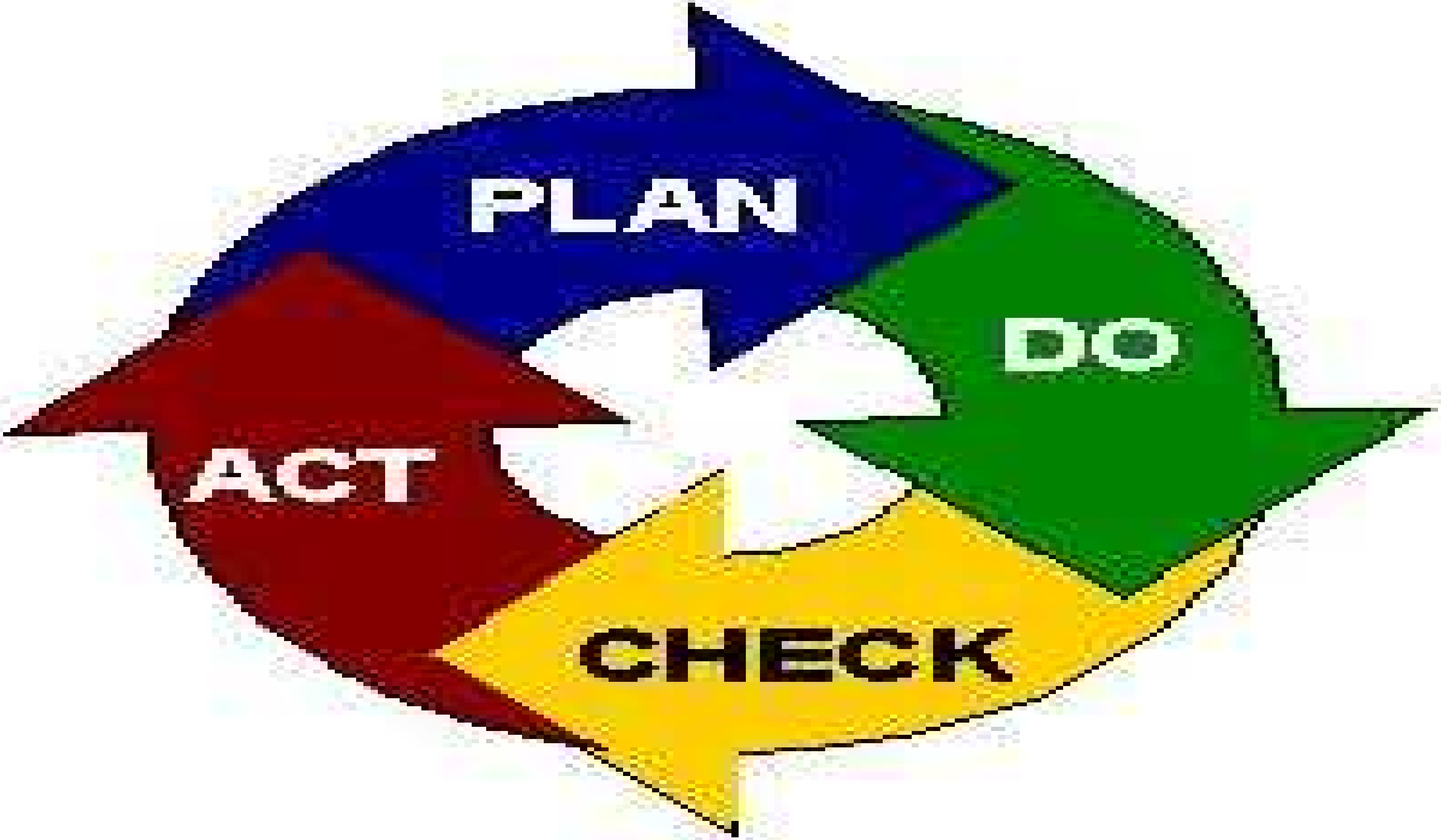
**Qualified People**
**Right Equipment**
**Correct Materials**
**Proven Methods**

**Indicators**

**Desired Results:**

**Quality Products**
**Quality Services**
**Customer Satisfaction**

# Why certification?



# Management System

The Basis of a Quality System is to …

- say what you do
- do what you say
- record what you did
- check the results
- and act on the difference

**ISO/TS 16949 is a quality management system designed specifically for automotive supply chain. It is applicable for auto makers and their suppliers.**

# Benefits of a Quality Management System (QMS)

- **Customer confidence**

- Better position on the **global market**

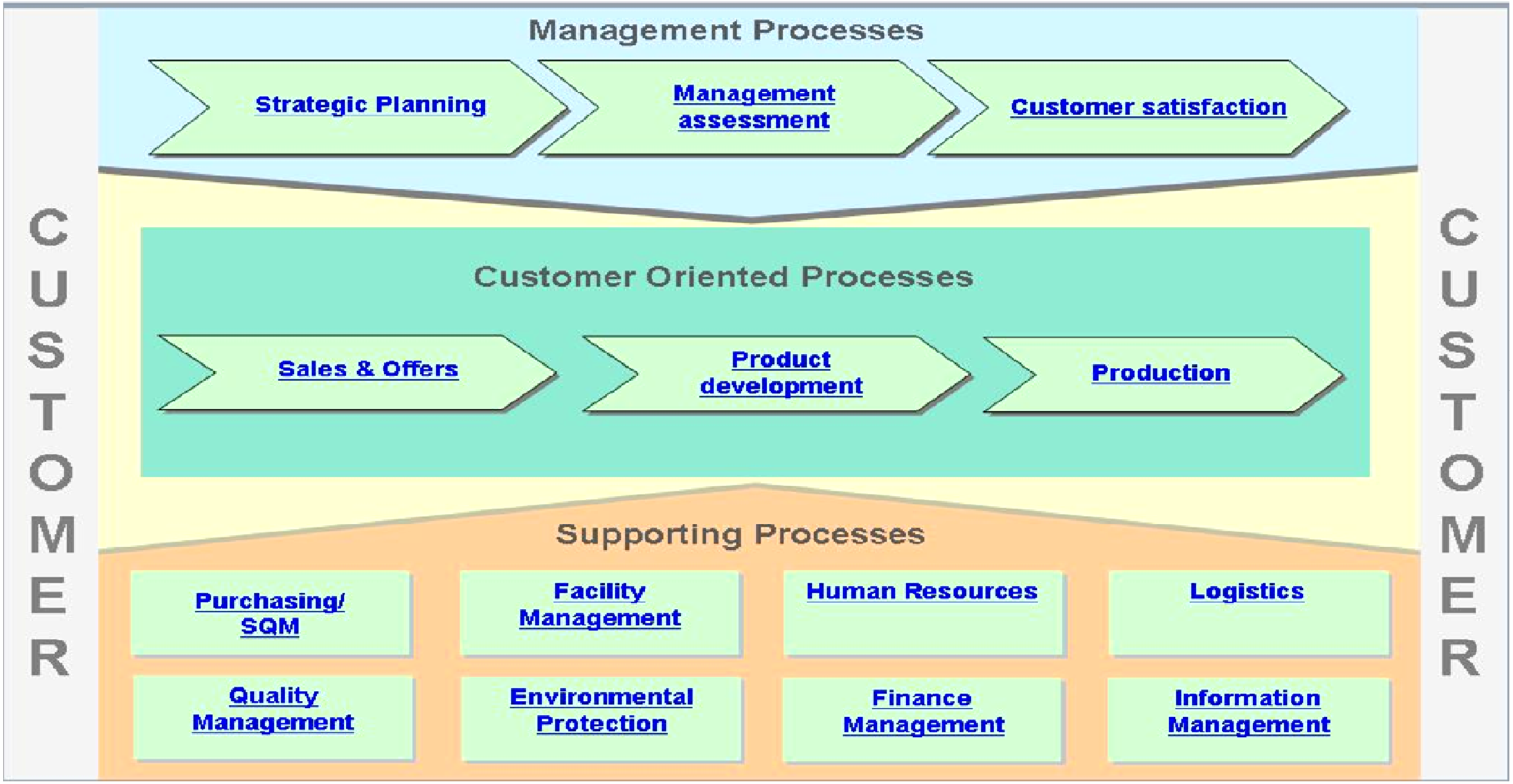
- Lower **production costs**

- Higher internal **operational efficiency**

- Higher **engagement and motivation** of employees

- Allows a **certification**

# Relevant processes



# Process improvement through PDCA cycle

The **process optimization** and **competitiveness enhancement** are reached via **continuous improvement** by means of **corrective and preventive actions** in the entire organization and during every business process.

*PLAN >> Establish the objectives and processes necessary to deliver results in accordance with customer requirements*

*DO >> Implement the processes*

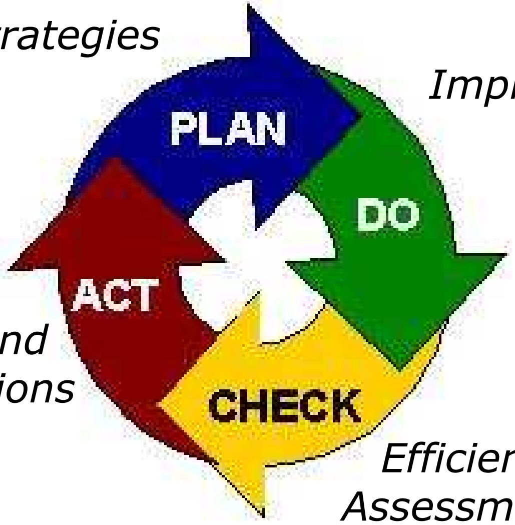
*CHECK >> Monitor and measure processes and product against objectives and requirements for the product and report the results*

*ACT >> Take actions to continually improve process performance*

***The way an optimal process runs***

*Strategies*

*Implementation*



*Preventive and corrective actions*

*Efficiency control Assessment/Analysis*

# Software Quality Assurance & Process Models

# Quality, Software Quality and Software Quality Assurance

## What is meant by quality?

- The **Institute of Electrical and Electronics Engineers' (IEEE)** defines quality as ***"the degree to which a system, component or process meets specified requirements and customer or user needs or expectations."***

- The **International Standards Organization (ISO)** defines quality as ***"the totality of features and characteristics of a product or service that bear on its ability to satisfy specified or implied needs."***

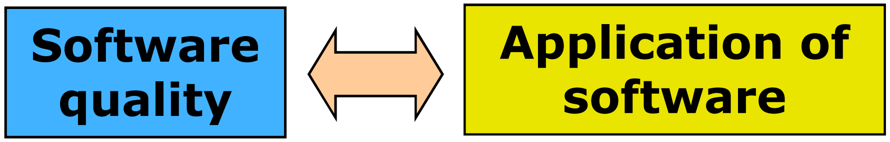
Kitchenham: “quality is hard to define, impossible to measure, easy to recognize."

**While these definitions seem to be clear and unambiguous, the concept of quality really is not!**

Gillies: "Quality is generally transparent when present, but easily recognized in its absence."

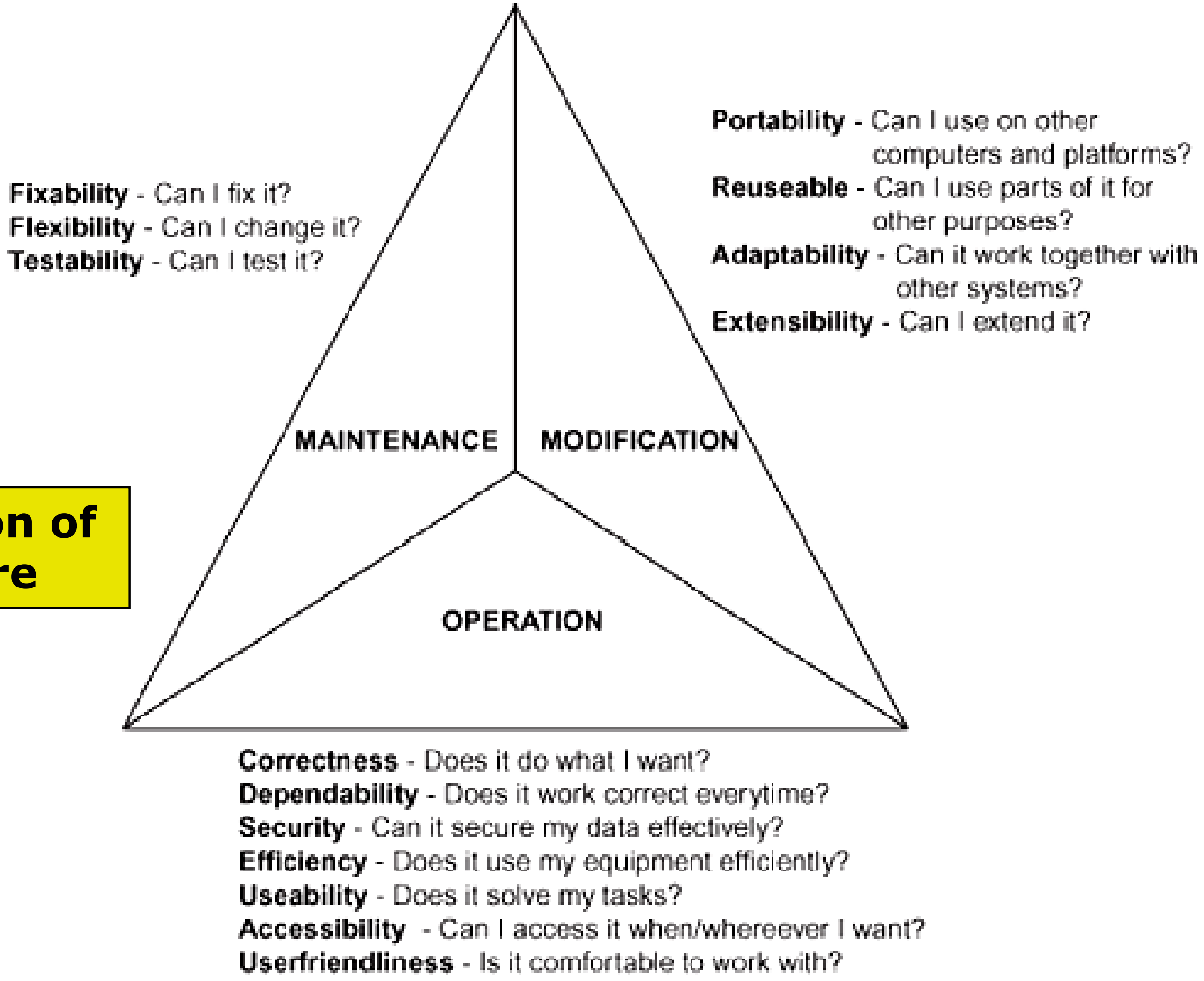
# Software Quality

- Definition from the **Handbook of Software Quality Assurance***:*
  ***"Software quality is the fitness for use of the software product."***



Software quality

Application of software



**Fixability** - Can I fix it?
**Flexibility** - Can I change it?
**Testability** - Can I test it?

**Portability** - Can I use on other computers and platforms?
**Reuseable** - Can I use parts of it for other purposes?
**Adaptability** - Can it work together with other systems?
**Extensibility** - Can I extend it?

MAINTENANCE

MODIFICATION

OPERATION

**Correctness** - Does it do what I want?
**Dependability** - Does it work correct everytime?
**Security** - Can it secure my data effectively?
**Efficiency** - Does it use my equipment efficiently?
**Useability** - Does it solve my tasks?
**Accessibility** - Can I access it when/whereever I want?
**Userfriendliness** - Is it comfortable to work with?

# The purpose of process

- **Effectiveness.** An effective process must help us produce the right product. The process should therefore help us determine what the customer needs, produce what the customer needs, and, crucially, verify that what we have produced *is* what the customer needs.
- **Maintainability.** However good the programmer, things will still go wrong with the software. Requirements often change between versions. In any case, we may want to reuse elements of the software in other products. One of the goals of a good process is to expose the designers' and programmers' thought processes in such a way that their intention is clear.
- **Predictability.** Any new product development needs to be planned, and those plans are used as the basis for allocating resources: both time and people. It is important to predict accurately how long it will take to develop the product. A good process helps lay out the steps of development. Furthermore, consistency of process allows us to learn from the designs of other projects.
- **Repeatability.** If a process is discovered to work, it should be replicated in future projects. A closely related issue, is that of **process re-use.** It is a huge waste and overhead for each project to produce a process from scratch. It is much faster and easier to adapt an existing process.
- **Quality.** Quality in this case may be defined as the product's fitness for its purpose. The process should provide a clear link between a customer's desires and a developer's product.
- **Improvement.** Even if we were as good as we could be now, both development environments and requested products are changing so quickly that our processes will always be running to catch up. A goal of our defined process must then be to identify and prototype possibilities for improvement in the process itself.
- **Tracking.** A defined process should allow the management, developers and customer to follow the status of a project. It keeps track of how good our predictions are, and hence how to improve them.

# Software Quality Assurance

- In the **Handbook of Software Quality Assurance**, the following definition is given: ***"Software quality assurance is the set of systematic activities providing evidence of the ability of the software process to produce a software product that is fit to use."***

**Objective**: identify and remove defects throughout the lifecycle, as early as possible

**The focus of SQA is to monitor continuously to ensure the quality of the delivered product:**

**Monitoring the processes**

Provides management with objective feedback regarding process compliance to approved plans, procedures, standards.

**Monitoring the products**

Focus on the quality of product within each phase of the SDLC (e.g., requirements, design, code, test plan, etc.).

# Process Models

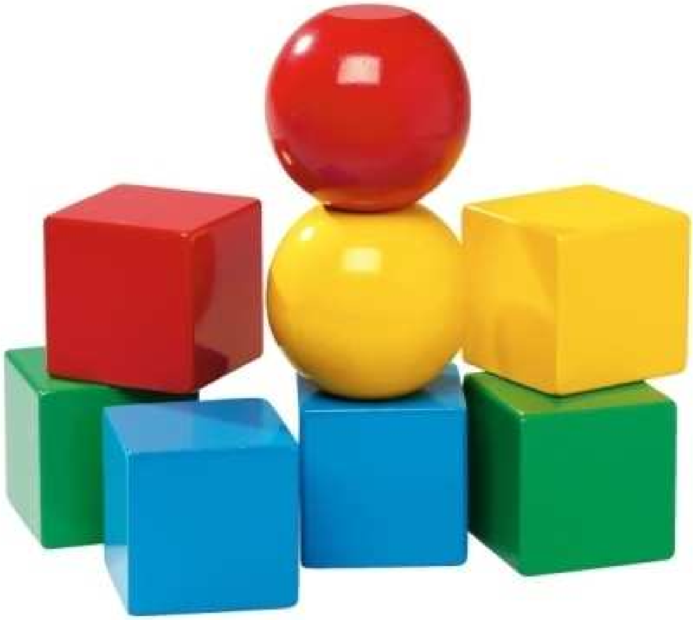
**Process Models** guide the **analysis**, **design**, **development** and **maintenance** of information systems.

Most system development Process Models in use today have evolved from three **primary approaches**:

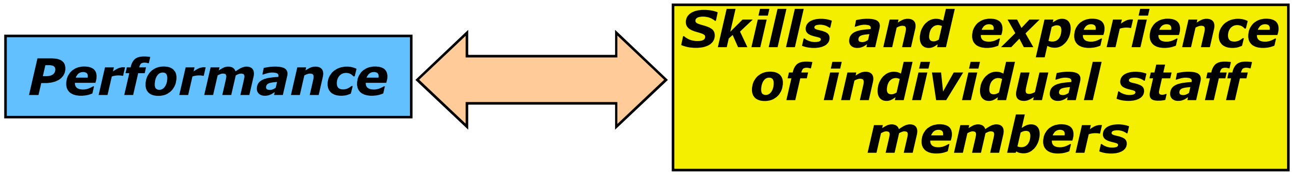
**1. Ad-hoc Development**

**2. Waterfall Model**

**3. the Iterative process.**



# Ad-hoc Development



***Performance***

***Skills and experience of individual staff members***

- In the absence of an organization-wide software process, repeating **results** depends entirely on having **the same individuals** available for the next project.

- **Success** that rests solely on the availability of specific individuals provides **no basis for long-term productivity!**

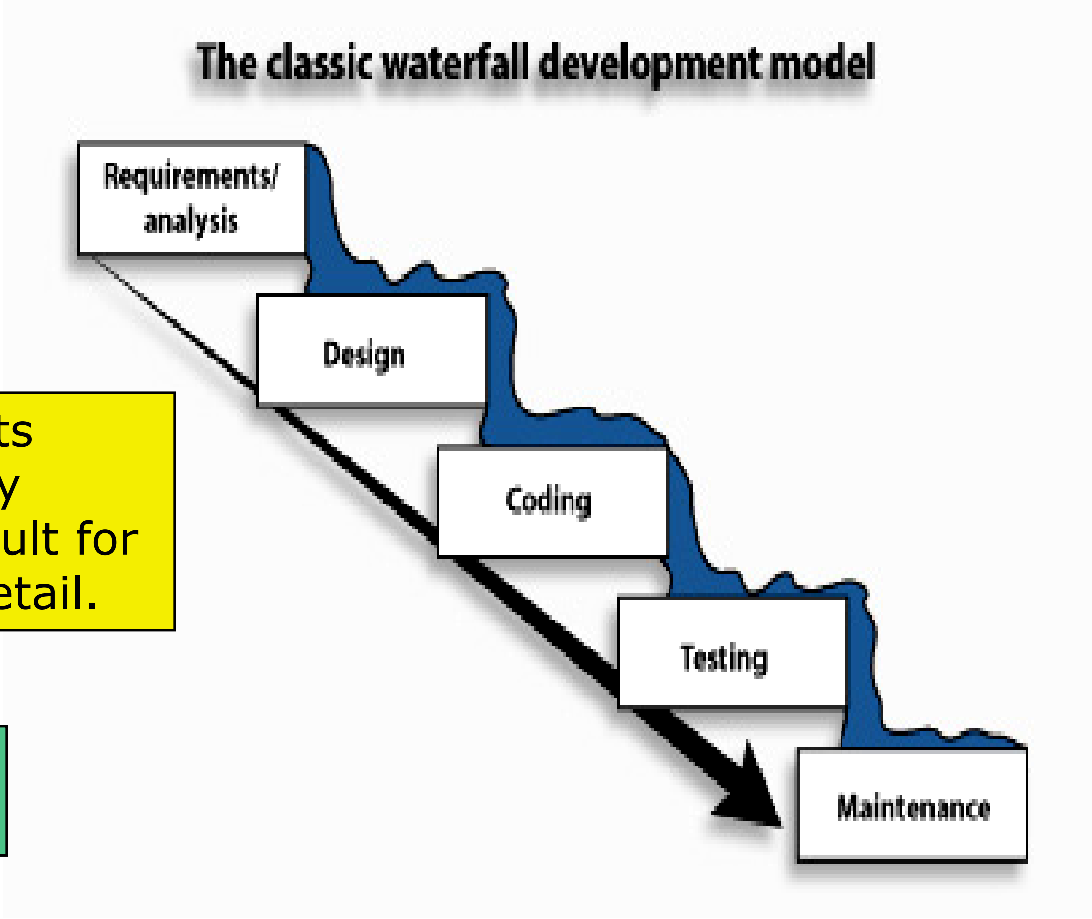
# The Waterfall Model

***Disadvantages come from rigid design:***

Real projects rarely follow the sequential flow proposed.

At the beginning of most projects there is great deal of uncertainty about requirements, being difficult for customers to identify them in detail.

A working version of the system is not yield until late.



# Iterative Development

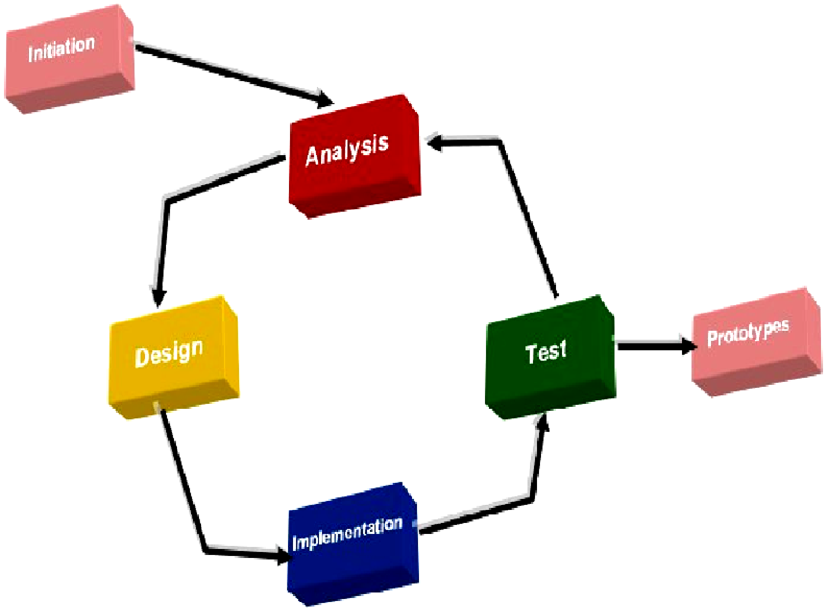
## *Advantages :*

**Provides faster results and requires less up-front information. The project is divided into small parts allowing to demonstrate results earlier on in the process and obtain valuable feedback from system users.**

## *Disadvantages:*

**The user community needs to be actively involved throughout the project. This is demanding on the time of the staff and can add project delay.**

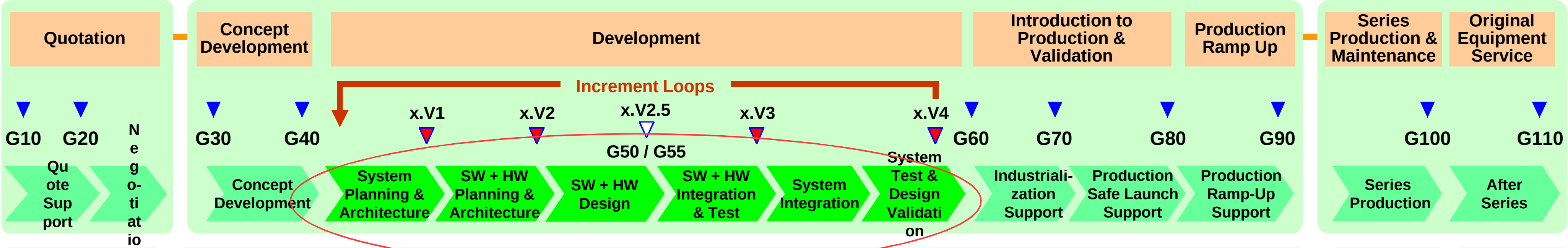
**User feedback following each phase may lead to increased customer demands. As users see the system develop, they may realize the potential of other system capabilities.**



# PLC @ BU IMM

# Conti PLC @ BU MM - Poster

## Main Deliverables per Phase


Quotation
Concept Development
Development
Introduction to Production & Validation
Production Ramp Up
Series Production & Maintenance
Original Equipment Service
Increment Loops
x.V1
x.V2
x.V2.5
x.V3
x.V4
G10
G20
G30
G40
G50 / G55
G60
G70
G80
G90
G100
G110
Quote Support
Nego-tiatio
Concept Development
System Planning & Architecture
SW + HW Planning & Architecture
SW + HW Design
SW + HW Integration & Test
System Integration
System Test & Design Validation
Industriali-zation Support
Production Safe Launch Support
Production Ramp-Up Support
Series Production
After Series

- Regularly: PM Project Report, Action Item List, Risk List, QM PQR, PPR's, Baselines

- RfQ
- PCIS (Quote)
- PM Effort Estimation
- Risk List
- Project Work Breakdown Structure
- Business Case
- Product Cost Calculation
- BoM
- QM Qualification Concept

- Disc. Effort Est.

- Feature List
- Basic System Architecture

- HW Non-Compliance List
- HW Technical Addendum

- Basic Production Concept
- MP OEL

- Project Plan
- Project Orgchart
- Project Schedule
- Staff.& Train. Plan
- PCIS (awarded)
- Business Case
- Prod. Cost Calc.
- QM Qualification Concept & Plan
- QM PQP
- Estimation of Quality Figures
- Initial DFT
- Special Char. List
- Service Concept

- Disc. Effort Est.

- Feature List
- (SyRD)
- SyADD, ConOps
- Patent Check
- System FMEA
- SE Computer Resource Est.
- System Rel. Plan
- SE Dev+QA Plan
- SE CM Plan

- (SWRD), SWAD
- SW Schedule
- SW Dev+QA Plan
- SW Test Plan

- HW-RD + RCS
- HW-ADD
- Justific. Sheet
- BoM
- Basic HW Design
- HW Verification Test Plan
- HW Schedule
- HW Dev. Plan

- MP Plan
- Prod. Concept
- Initial DFA

- Project Plan
- Project Orgchart
- Project Schedule
- PM Effort Estimation
- Object Rev. Status List
- QM Production Launch Checklist

- Feature List
- SyRD
- (SyADD)
- (ConOps)
- System Integr. Plan
- System Release Plan
- SE Effort Estimation
- SE Schedule

- PVV Plan
- PVV Schedule

- SWRD+ RCS
- SWAD
- SW Integr. Plan
- SW Effort Estimation
- SW Schedule
- SW Dev. Plan

- HW-RD+RCS
- HW-ADD
- Justific. Sheet
- HW Verific. Test Plan
- HW Effort Estimation
- HW Schedule
- HW Dev.Plan

- PVV V&V Plan

- SW Design
- Code
- Code Review
- Static Code Analysis
- SW Module Test Spec + Report
- SW Integr. Test Spec
- SW Verific. Test Spec

- BoM
- HW-SW Interface Spec.
- HW Comp Spec
- HW Electr Spec
- Patent Check
- Schematics
- Layout
- MCAD Data
- Drawings
- D-FMEA
- HW Verific.

- P-FMEA
- MP OEL

- System Integr. Test Spec
- System Module Acceptance Report

- SW Integr. Test Report
- SW Verific. Test Report
- SW Release Note
- SW CM Audit

- HW Prototype Build
- HW Verific. Test Report
- HW Release Note

- PVV Test Cases
- PVV Qual. Test Spec

- System Integr. Test
- System Ver. Test
- System Release Note
- SE CM Audit

- Lessons Learned

- PVV Trace & Baseline Report
- PVV Test Report
- PVV Qual. Test Report
- PVV Maturity Report

C-Prototype:
- Production BoM
- Working Instructions
- Pre Launch Control Plan

- QM Production Launch Checklist

- D-FMEA
- HW-PTS

- C-Prototype Report
- Process / Machine Capability Analysis
- Application Release on Production Equipment
- Ramp Up Plan
- P-FMEA

- PPAP file & approval
- Lessons Learned

- Series Control Plan
- Production Line Audit
- Release for Series Production

- Project Archived (Doc + Code)
- Lessons Learned

- SW Transfer to Maintenance

- HW Transfer to

- Series Production Started

Maintenance Project:

Update of:
- QM Effort Estimation
- QM PQP
- QM Qualific. Plan
- QM Prod. Launch Checklist
- Special Char. List
- PPAP file

- <Support>

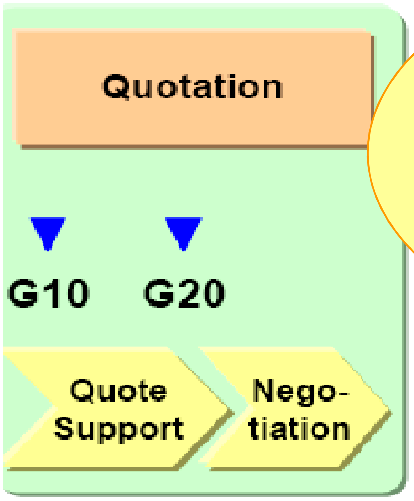
| |
|---|
| PM / other |
| all |
| SE |
| SW |
| HW |
| PVV |
| MP |
| PU* |

| PLC GATE | | Authorizes |
|---|---|---|
| G10 | "Opportunity approved" | Preparation of quotation package |
| G20 | "Quote approved | Submission of quotation package |
| G30 | "Order received" | Start of concept development |
| G40 | "Final Concept approved" | Supplier pre-selection |
| G50 | "Design defined"<br>**= B1 V2.5** (GoForPrototype) | First (B1-) Build of prototypes and submission to customer |
| G55 | "Design ready to start manufacturing process"<br>**= C V2.5** (GoForTooling) | Order supplier tooling and pre-order of production equipment |
| G60 | "Product Design approved" | Procurement of production equipment |
| G70 | "Production Process confirmed" | Production of PPAP samples |
| G80 | "Production Part approved" | Unlimited series production |
| G90 | "Production Launch completed" | Closing of development project & budget |
| G100 | "Series Production ended" | Stock-keeping of parts / produced products |
| G110 | "Delivery Obligation ended" | Disposal of equipment, material & tooling |

* Main deliverables
PU to be defined

I [illegible]ining
Radauceanu Smaranda & Grigoras Simona
March 2015

# Quotation Phase



Cost is defined for the development of the product and for the product itself.

The Quote Bit is discussed with the customer and an agreement is reached (or not)

- RfQ
- PCIS (Quote) (Pre Commitment Information System)
- PM Effort Estimation
- Risk List
- Project Work Breakdown Structure
- Business Case
- Product Cost Calculation
- BoM

- Disc. Effort Est.

**PCIS** is an economic efficiency calculation.

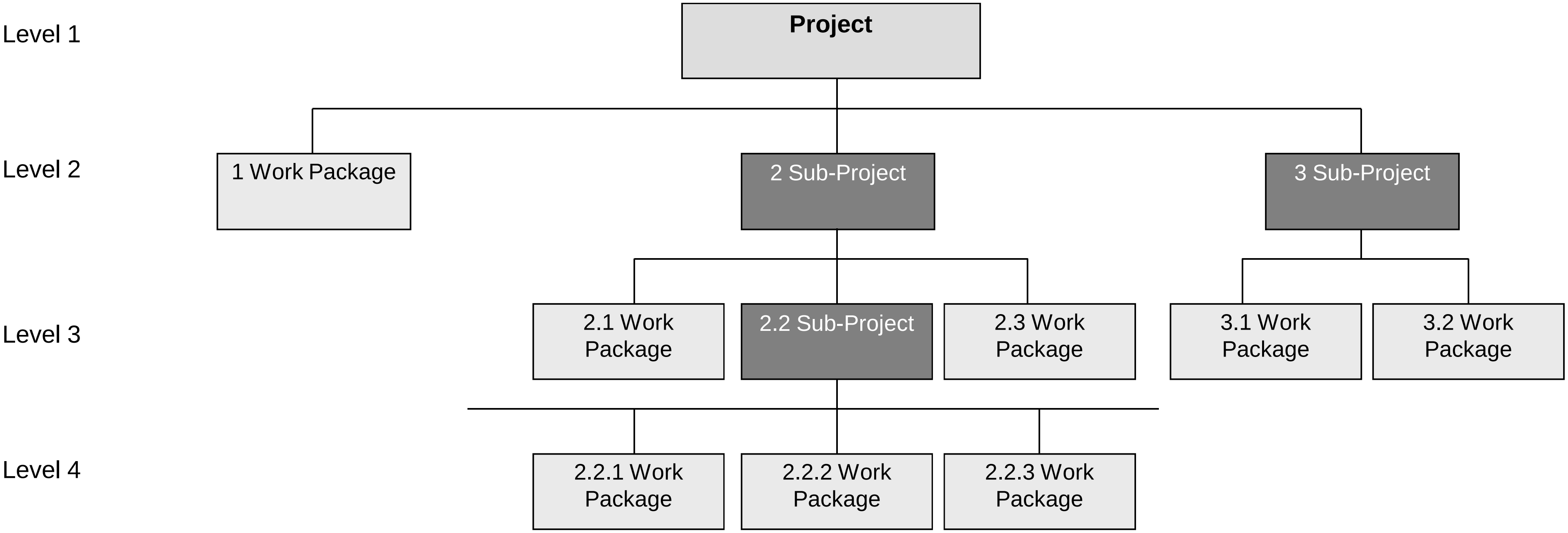
**PM Effort Estimation** consists in predicting how many hours of work and how many workers are needed to develop the project.

**Risk Management** is a continuous process to be performed throughout the entire lifecycle of a project. The objective of risk management is to identify and prevent risks, to reduce their probability of occurrence or to mitigate the effects in case of risk occurrence.

A **Risk List** documents potential obstacles to the successful achievement of project targets (technical, costs, time schedule). A risk is a combination of the probability of an event and its consequence. Counter measured should be defined such as:

-avoidance ("preventive measures")
-transfer (e.g. to a third party)
-mitigation (early measures to reduce the probability of occurrence and/or impact)
-acceptance (counter measures are not possible or uneconomical; it is advisable in this case to plan reserves of time, resources, etc).

# Quotation Phase - WBS

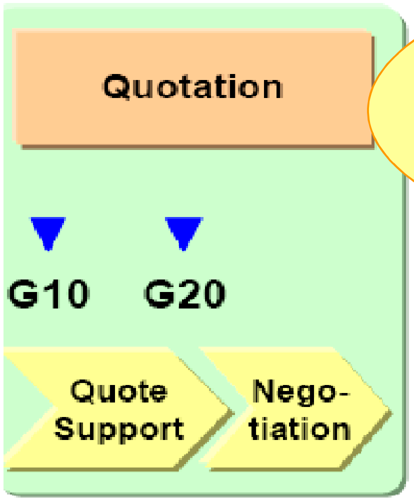


This WBS is illustrative only. It is not intended to represent the full project scope of any specific project, nor to imply that this is the only way to organize a WBS.

A project **WBS** is a decomposition of the (complex) major project deliverables into smaller, more manageable components.

Each lower level contains a more detailed description of the project items on the level above. **W**ork **P**ackages constitute the lowest level of the WBS. At the top level, the WBS should be structured according to project phases.

# Quotation Phase



Cost is defined for the development of the product and for the product itself.

The Quote Bit is discussed with the customer and an agreement is reached (or not)

- RfQ
- PCIS (Quote) (Pre Commitment Information System)
- PM Effort Estimation
- Risk List
- Project Work Breakdown Structure
- Business Case
- Product Cost Calculation
- BoM

- Disc. Effort Est.

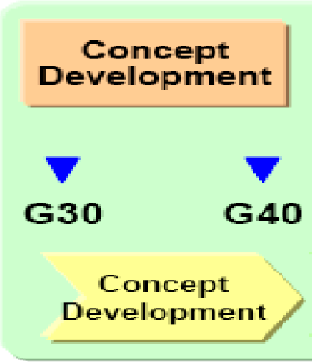
The purpose of a **business case** is to give an investor all necessary information that allow him to weigh the chances and risks of his investment. The business case is governed by three questions:
1. Will it be a product that the customer wants to buy?
2. How big will the market be and will it be a profitable market?
3. Is the strategy to develop the product and approach the market convincing?

The **Product Cost Calculation** lists the combined costs of raw materials, labor, maintenance necessary in producing the product.

**Disc. (SW) Effort Est.** - the goal of Estimation in a discipline (SW) is to obtain realistic effort estimates for all activities to be performed within a discipline (SW) Project, in order to generate a realistic schedule and cost estimate for the discipline (SW) Project.

The effort required for the implementation of a work package or a module (e.g. in hours, person days) is estimated, based on the project's WBS. Effort is frequently estimated bottom-up, breaking the work to be done into small work packages. At the lowest level, it is estimated on a per WP basis and then added up step by step to calculate the total estimated project effort.

# Concept Development Phase



development of System Req., System Arch. and detailed Project Plan

- Project Plan
- Project Orgchart
- Project Schedule
- Staff.& Train. Plan
- Initial DFT
- Special Char. List

- Feature List
- (SyRD)
- SyADD, ConOps
- SE Computer Resource Est.
- System Rel. Plan
- SE Dev+QA Plan
- SE CM Plan

- Disc. Effort Est.

(SWRD), SWAD
SW Schedule
SW Dev+QA Plan
SW Test Plan

The **Project Plan** is a document used to guide both *project execution* and *project control*. The primary uses of the project plan are to document planning assumptions and decisions, facilitate communication among *stakeholders*, and document approved scope, cost and schedule *baselines*.

A **Project Orgchart** is the graphic illustration of a project structure, showing hierarchical authority and relationships between team members and jobs.

The **Project Schedule** is the detailed plan of major project phases, gates, activities, tasks and the resources allocated to each task.

**Initial DFT** - In the literature, avoiding bugs is referred to as DFT, Design For Testing. Great care has to be taken in defining a safe subset of C (arranged in quality goals), so that also the novice C-programmer will be guided to produce safe, readable and maintainable code.

**Special Char. List** are product characteristics or manufacturing process parameters, which can affect safety or compliance with regulations, fit, function, performance or subsequent processing of product or as directed by the customer.

In automotive, the product scope is often planned in the form of **functions to be implemented**, by means of a **Feature list.** This list is used to plan the content of individual samples and SW releases and to track the implementation. The customer knows which functions are planned for which delivery and when they are available for him for testing.

**ConOps – Concept of Operation** is a document describing the characteristics of a proposed system from the viewpoint of an individual who will use that system. It is used to communicate the quantitative and qualitative system characteristics to all stakeholders.

# Project Planning

**Project planning identifies the project’s needs and constraints and lays out the activities, resources, budget, and timeline for the project.**

A project plan can be considered to have **five key characteristics** that have to be managed:

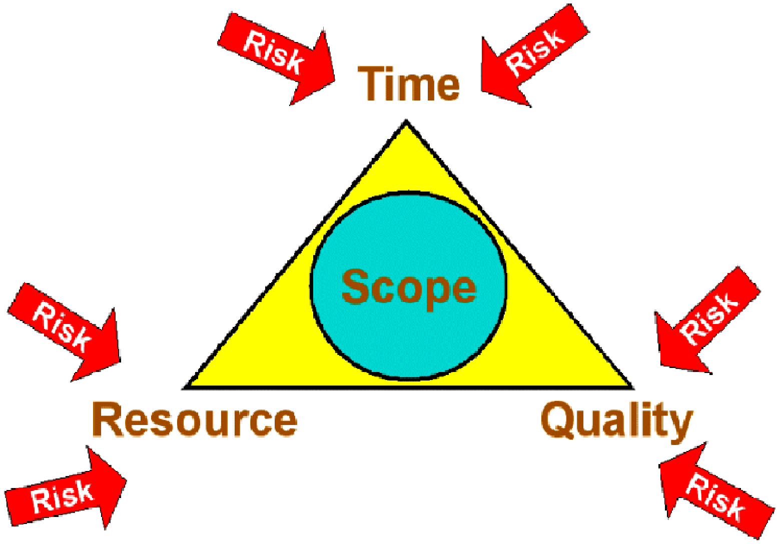
**Scope:** defines what will be covered in a project.

**Resource:** what can be used to meet the scope.

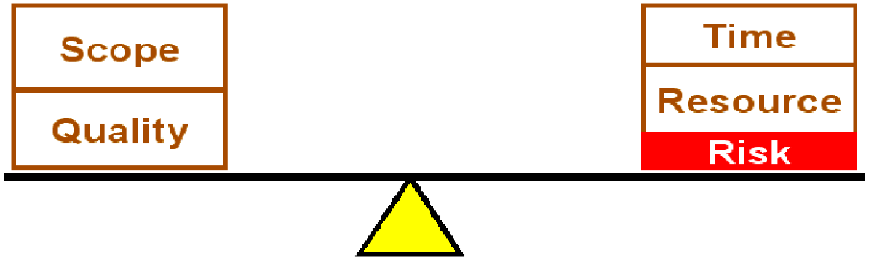
**Time:** what tasks are to be undertaken and when.

**Quality:** the deviation allowed from a desired standard.

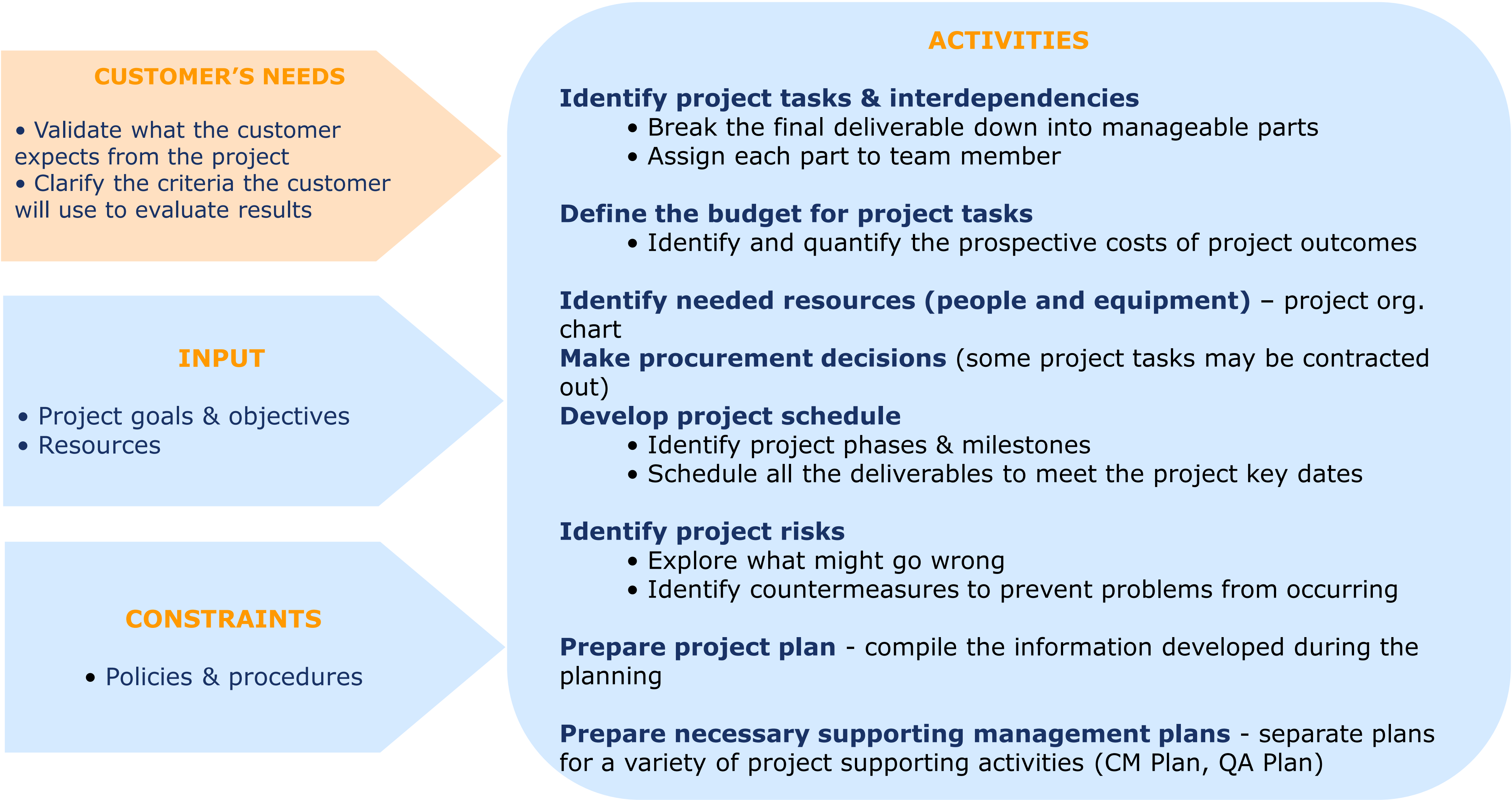
**Risk:** defines in advance what may happen to drive the plan off course and what will be done to recover the situation.



The point of a plan is to **balance**:
the **scope**, and **quality** constraint against the **time** and **resource** constraint, while minimizing the **risks**.



# Project Planning



**CUSTOMER’S NEEDS**

- Validate what the customer expects from the project
- Clarify the criteria the customer will use to evaluate results

**INPUT**

- Project goals & objectives
- Resources

**CONSTRAINTS**

- Policies & procedures

**ACTIVITIES**

**Identify project tasks & interdependencies**

- Break the final deliverable down into manageable parts
- Assign each part to team member

**Define the budget for project tasks**

- Identify and quantify the prospective costs of project outcomes

**Identify needed resources (people and equipment)** – project org. chart

**Make procurement decisions** (some project tasks may be contracted out)

**Develop project schedule**

- Identify project phases & milestones
- Schedule all the deliverables to meet the project key dates

**Identify project risks**

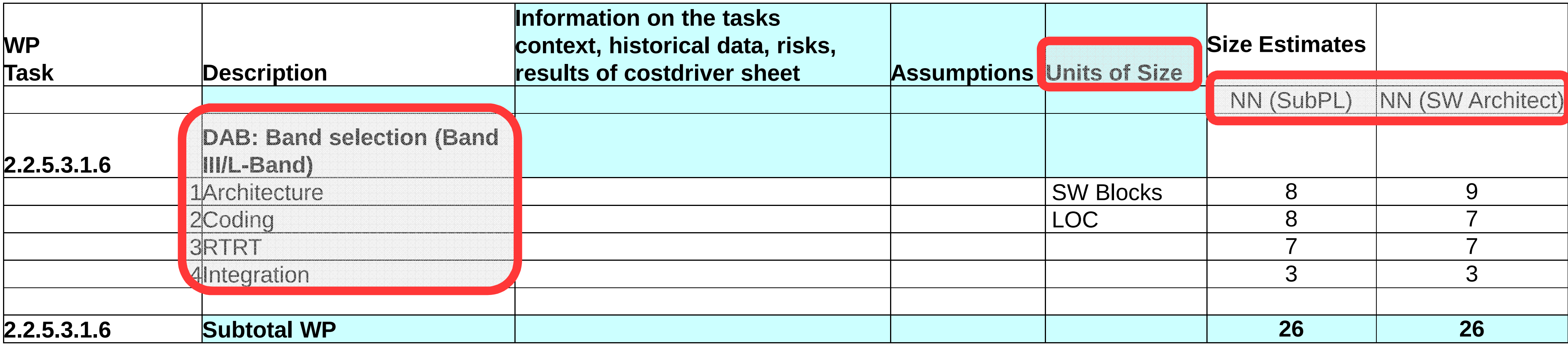
- Explore what might go wrong
- Identify countermeasures to prevent problems from occurring

**Prepare project plan** - compile the information developed during the planning

**Prepare necessary supporting management plans** - separate plans for a variety of project supporting activities (CM Plan, QA Plan)

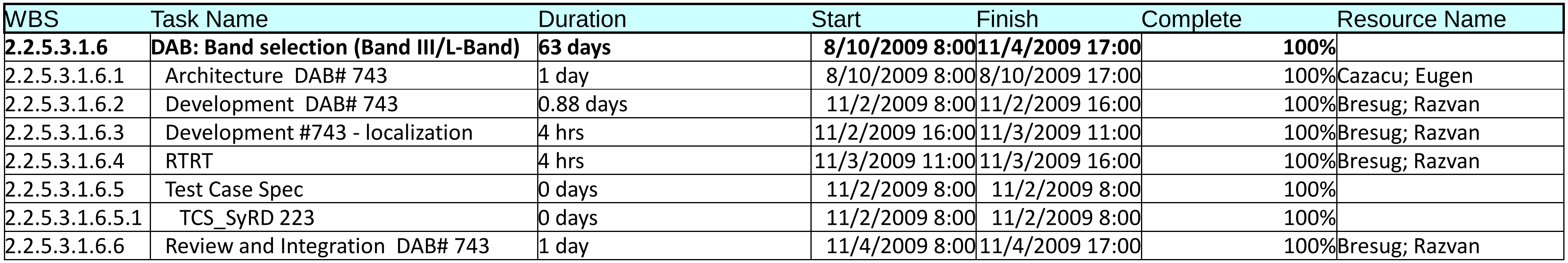
# Effort Estimation

**Effort Estimation** (ex: Four eyes)



| WP<br>Task | Description | Information on the tasks context, historical data, risks, results of costdriver sheet | Assumptions | Units of Size | Size Estimates | |
|---|---|---|---|---|---|---|
| | | | | | NN (SubPL) | NN (SW Architect) |
| 2.2.5.3.1.6 | DAB: Band selection (Band III/L-Band) | | | | | |
| 1 | Architecture | | | SW Blocks | 8 | 9 |
| 2 | Coding | | | LOC | 8 | 7 |
| 3 | RTRT | | | | 7 | 7 |
| 4 | Integration | | | | 3 | 3 |
| | | | | | | |
| 2.2.5.3.1.6 | Subtotal WP | | | | 26 | 26 |

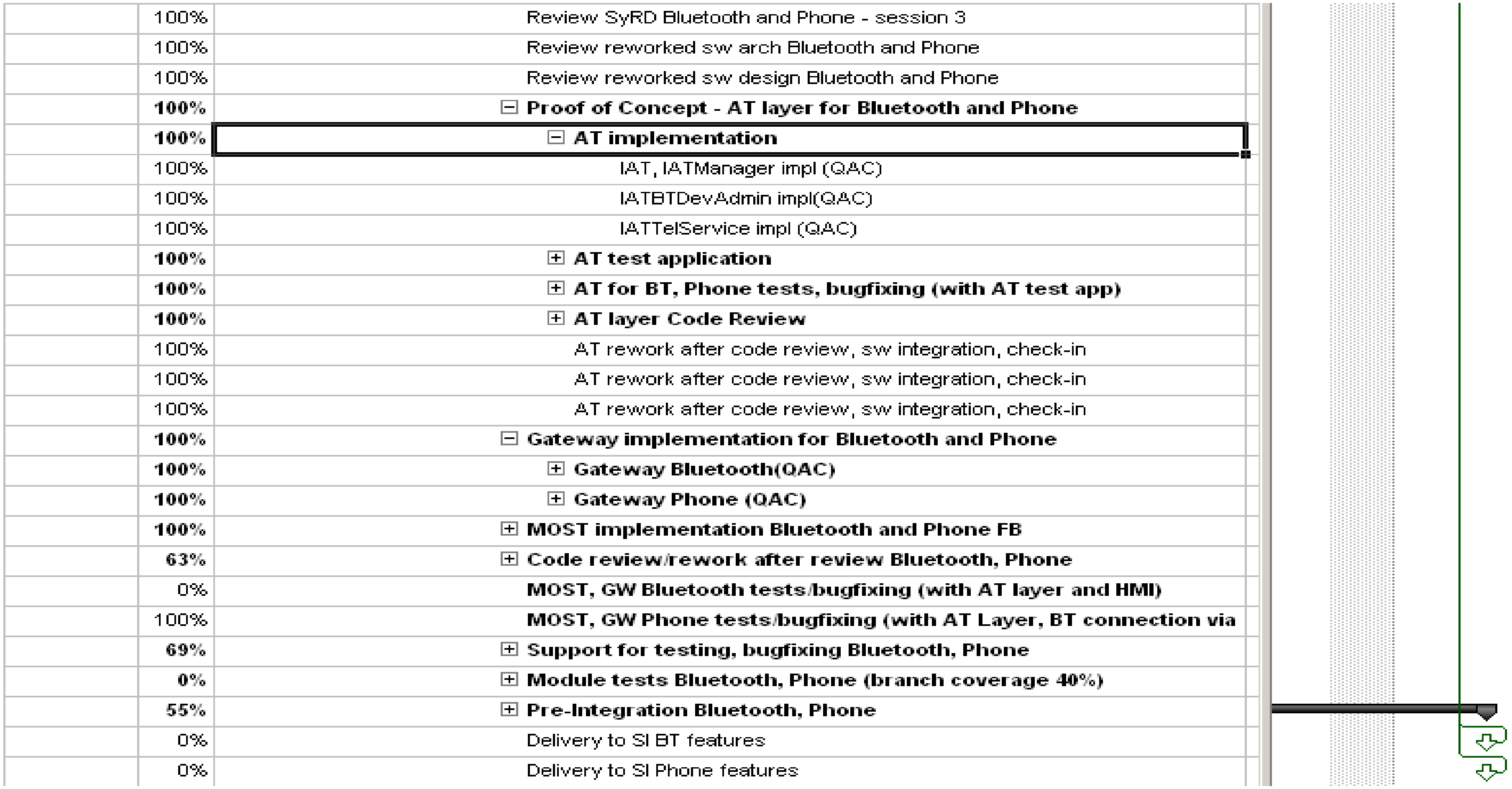
Microsoft **Project Planning** (Online via MSPE or Offline as share drive copy of the planning)



| WBS | Task Name | Duration | Start | Finish | Complete | Resource Name |
|---|---|---|---|---|---|---|
| **2.2.5.3.1.6** | **DAB: Band selection (Band III/L-Band)** | **63 days** | **8/10/2009 8:00** | **11/4/2009 17:00** | **100%** | |
| 2.2.5.3.1.6.1 | Architecture DAB# 743 | 1 day | 8/10/2009 8:00 | 8/10/2009 17:00 | 100% | Cazacu; Eugen |
| 2.2.5.3.1.6.2 | Development DAB# 743 | 0.88 days | 11/2/2009 8:00 | 11/2/2009 16:00 | 100% | Bresug; Razvan |
| 2.2.5.3.1.6.3 | Development #743 - localization | 4 hrs | 11/2/2009 16:00 | 11/3/2009 11:00 | 100% | Bresug; Razvan |
| 2.2.5.3.1.6.4 | RTRT | 4 hrs | 11/3/2009 11:00 | 11/3/2009 16:00 | 100% | Bresug; Razvan |
| 2.2.5.3.1.6.5 | Test Case Spec | 0 days | 11/2/2009 8:00 | 11/2/2009 8:00 | 100% | |
| 2.2.5.3.1.6.5.1 | TCS_SyRD 223 | 0 days | 11/2/2009 8:00 | 11/2/2009 8:00 | 100% | |
| 2.2.5.3.1.6.6 | Review and Integration DAB# 743 | 1 day | 11/4/2009 8:00 | 11/4/2009 17:00 | 100% | Bresug; Razvan |

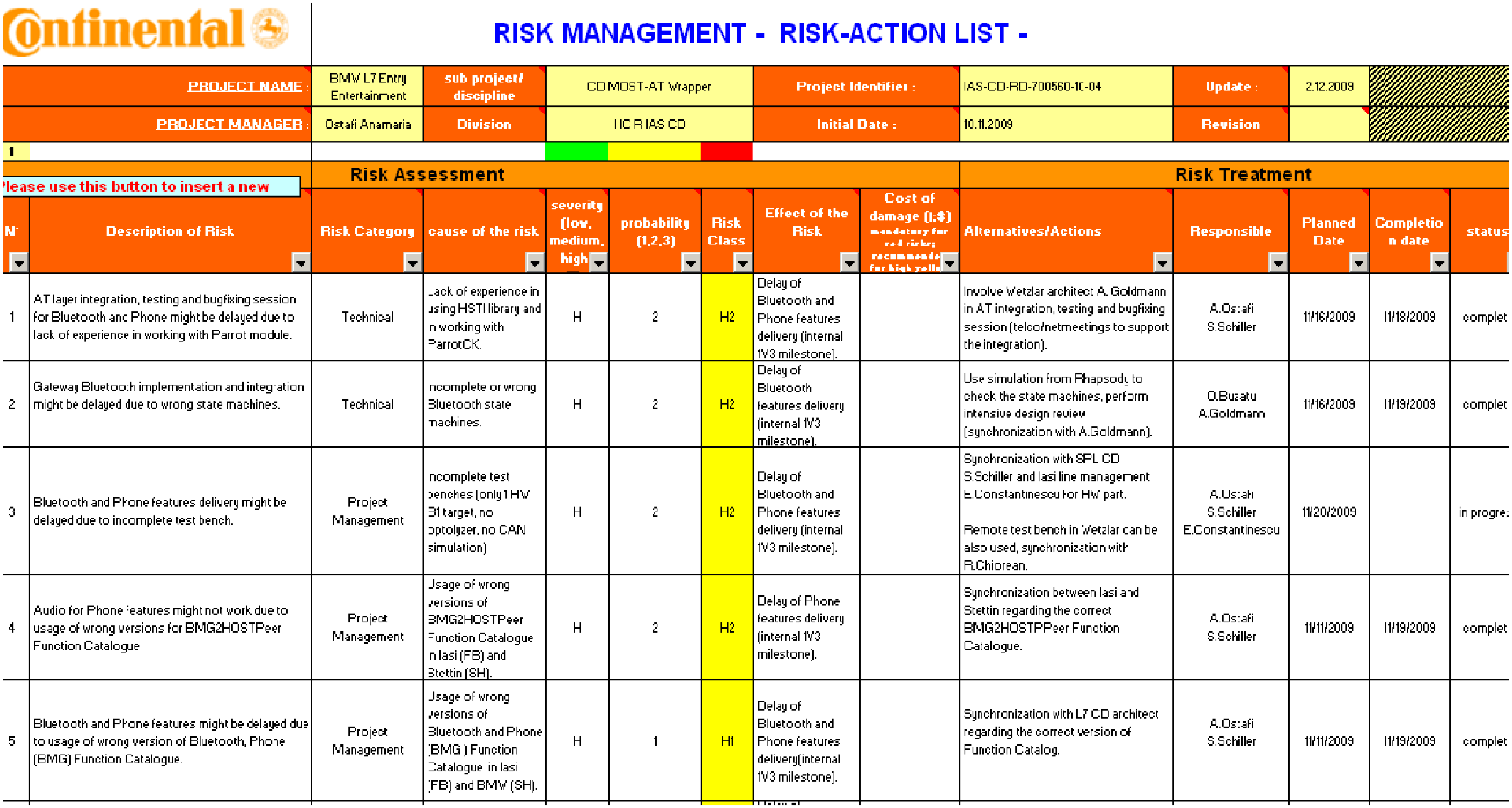
# Project Planning

- Example of planning including WBS and work packages for SW



| | | |
|---|---|---|
| | 100% | Review SyRD Bluetooth and Phone - session 3 |
| | 100% | Review reworked sw arch Bluetooth and Phone |
| | 100% | Review reworked sw design Bluetooth and Phone |
| | **100%** | ⊟ **Proof of Concept - AT layer for Bluetooth and Phone** |
| | **100%** | ⊟ **AT implementation** |
| | 100% | IAT, IATManager impl (QAC) |
| | 100% | IATBTDevAdmin impl(QAC) |
| | 100% | IATTelService impl (QAC) |
| | **100%** | ⊞ **AT test application** |
| | **100%** | ⊞ **AT for BT, Phone tests, bugfixing (with AT test app)** |
| | **100%** | ⊞ **AT layer Code Review** |
| | 100% | AT rework after code review, sw integration, check-in |
| | 100% | AT rework after code review, sw integration, check-in |
| | 100% | AT rework after code review, sw integration, check-in |
| | **100%** | ⊟ **Gateway implementation for Bluetooth and Phone** |
| | **100%** | ⊞ **Gateway Bluetooth(QAC)** |
| | **100%** | ⊞ **Gateway Phone (QAC)** |
| | **100%** | ⊞ **MOST implementation Bluetooth and Phone FB** |
| | **63%** | ⊞ **Code review/rework after review Bluetooth, Phone** |
| | 0% | **MOST, GW Bluetooth tests/bugfixing (with AT layer and HMI)** |
| | 100% | **MOST, GW Phone tests/bugfixing (with AT Layer, BT connection via** |
| | **69%** | ⊞ **Support for testing, bugfixing Bluetooth, Phone** |
| | **0%** | ⊞ **Module tests Bluetooth, Phone (branch coverage 40%)** |
| | **55%** | ⊞ **Pre-Integration Bluetooth, Phone** |
| | 0% | Delivery to SI BT features |
| | 0% | Delivery to SI Phone features |

# Risk List



RISK MANAGEMENT - RISK-ACTION LIST -

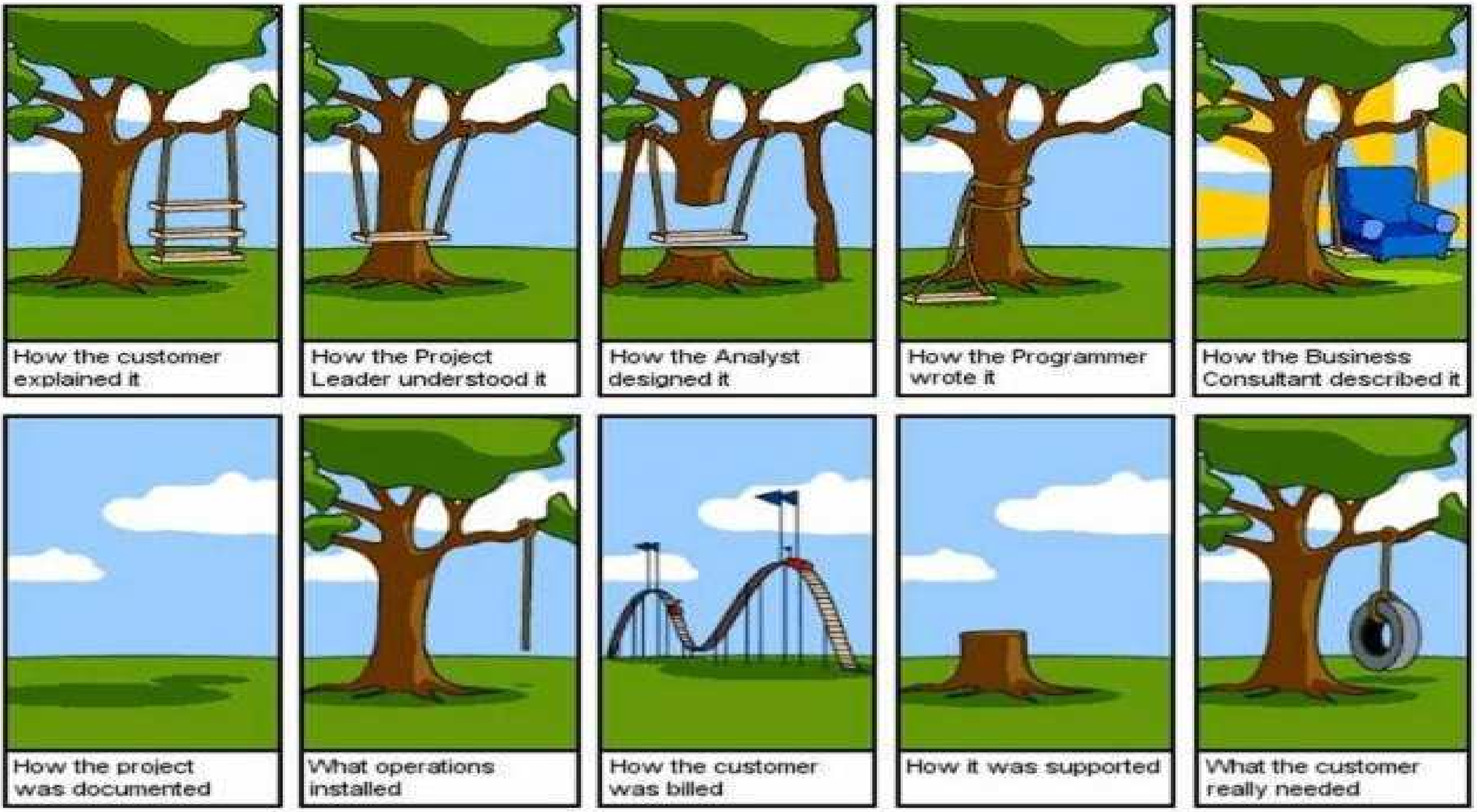
| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| PROJECT NAME : | BMW L7 Entry Entertainment | sub project/ discipline | CD MOST-AT Wrapper | Project Identifier : | IAS-CD-RD-700560-1C-04 | Update : | 2.12.2009 | ▨ |
| PROJECT MANAGER : | Ostafi Anamaria | Division | I IC R IAS CD | Initial Date : | 10.11.2009 | Revision | | ▨ |

1 🟩🟨🟥

Please use this button to insert a new

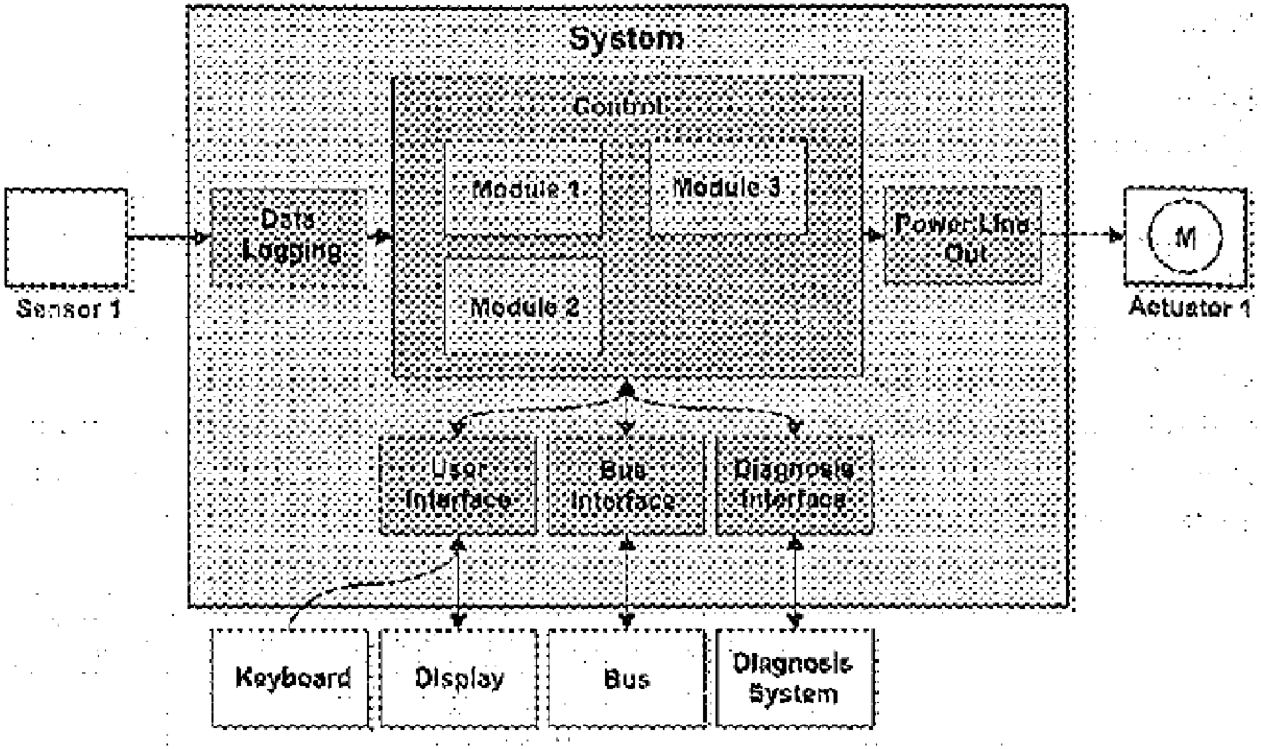
| | | Risk Assessment | | | | | | | Risk Treatment | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| N° | Description of Risk | Risk Category | cause of the risk | severity (low, medium, high | probability (1,2,3) | Risk Class | Effect of the Risk | Cost of damage (I,$) mandatory for red risks; recommende for high yello | Alternatives/Actions | Responsible | Planned Date | Completion date | status |
| 1 | AT layer integration, testing and bugfixing session for Bluetooth anc Phone might be delayed due to lack of experience in working with Parrot module. | Technical | Lack of experience in using HSTI library and in working with ParrotCK. | H | 2 | H2 | Delay of Bluetooth and Phone features delivery (internal IV3 milestone). | | Involve Wetzlar architect A. Goldmann in AT integration, testing and bugfixing session (telco/netmeetings to support the integration). | A.Ostafi<br>S.Schiller | 11/16/2009 | 11/18/2009 | complet |
| 2 | Gateway Bluetooth implementation and integration might be delayed due to wrong state machines. | Technical | Incomplete or wrong Bluetooth state machines. | H | 2 | H2 | Delay of Bluetooth features delivery (internal IV3 milestone). | | Use simulation from Rhapsody to check the state machines, perform intensive design review (synchronization with A.Goldmann). | D.Buzatu<br>A.Goldmann | 11/16/2009 | 11/19/2009 | complet |
| 3 | Bluetooth and Phone features delivery might be delayed due to incomplete test bench. | Project Management | Incomplete test benches (only 1 HW B1 target, no optolyzer, no CAN simulation) | H | 2 | H2 | Delay of Bluetooth and Phone features delivery (internal IV3 milestone). | | Synchronization with SPL CD S.Schiller and Iasi line management E.Constantinescu for HW part.<br>Remote test bench in Wetzlar can be also used, synchronization with R.Chiorean. | A.Ostafi<br>S.Schiller<br>E.Constantinescu | 11/20/2009 | | in progre: |
| 4 | Audio for Phone features might not work due to usage of wrong versions for BMG2HOSTPeer Function Catalogue | Project Management | Usage of wrong versions of BMG2HOSTPeer Function Catalogue in Iasi (FB) and Stettin (SH). | H | 2 | H2 | Delay of Phone features delivery (internal IV3 milestone). | | Synchronization between Iasi and Stettin regarding the correct BMG2HOSTPeer Function Catalogue. | A.Ostafi<br>S.Schiller | 11/11/2009 | 11/19/2009 | complet |
| 5 | Bluetooth and Phone features might be delayed due to usage of wrong version of Bluetooth, Phone (BMG) Function Catalogue. | Project Management | Usage of wrong versions of Bluetooth and Phone (BMG) Function Catalogue in Iasi (FB) and BMW (SH). | H | 1 | H1 | Delay of Bluetooth and Phone features delivery(internal IV3 milestone). | | Synchronization with L7 CD architect regarding the correct version of Function Catalog. | A.Ostafi<br>S.Schiller | 11/11/2009 | 11/19/2009 | complet |

# System Requirements

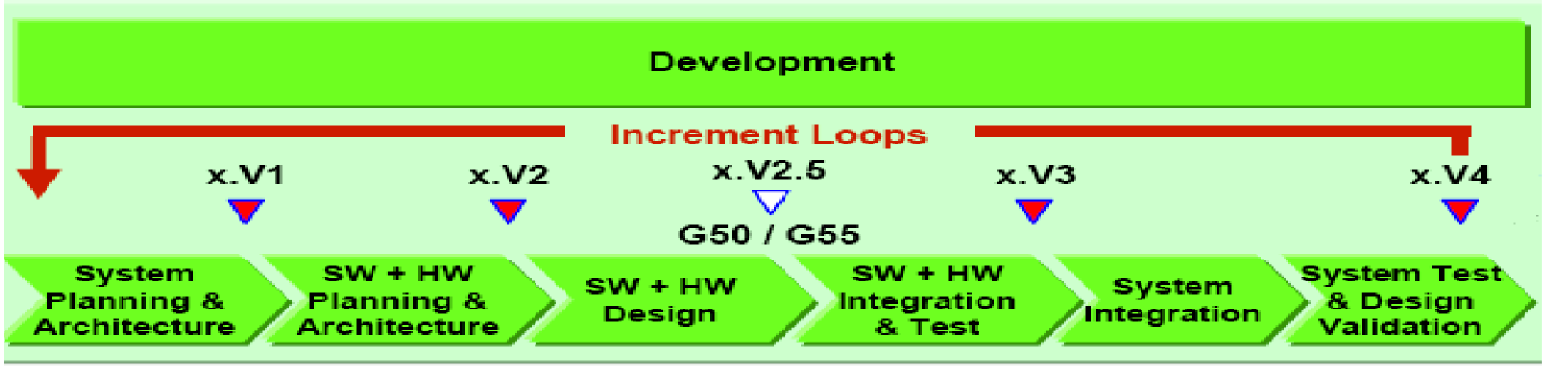

How the customer explained it
How the Project Leader understood it
How the Analyst designed it
How the Programmer wrote it
How the Business Consultant described it
How the project was documented
What operations installed
How the customer was billed
How it was supported
What the customer really needed

# System Architecture

- The **SyADD** provides an **overview of the system as a whole** and **describes the interaction of individual elements**, including software. The overall system typically consists of one or several **block diagrams** illustrating the **inter-relationships and data flows between system elements**. For each system element the associated system requirements must be shown.
- During the definition of the SyArch a decision is made as to which functions will be implemented in the hardware and which will be implemented by software.



# The Development Phase



Development

Increment Loops

x.V1 x.V2 x.V2.5 x.V3 x.V4

G50 / G55

System Planning & Architecture → SW + HW Planning & Architecture → SW + HW Design → SW + HW Integration & Test → System Integration → System Test & Design Validation

•Regularly: PM Project Report, Action Item List, Risk List, PPR's

Project Plan
Project Orgchart
Project Schedule
PM Effort Estimation
Object Rev. Status List

Feature List
SyRD
(SyADD)
System Integr. Plan
SE Effort Estimation
SE Schedule

PVV Plan
PVV Schedule

SWRD+ RCS
SWAD
SW Integr. Plan
SW Effort Estimation
SW Schedule
SW Dev. Plan

PVV Plan

SW Design
Code
Code Review
Static Code Analysis
SW Module Test Spec + Report
SW Integr. Test Spec
SW Verific. Test Spec

System Integr. Test Spec

SW Integr. Test Report
SW Verific. Test Report
SW Release Note
SW CM Audit

PVV Test Cases

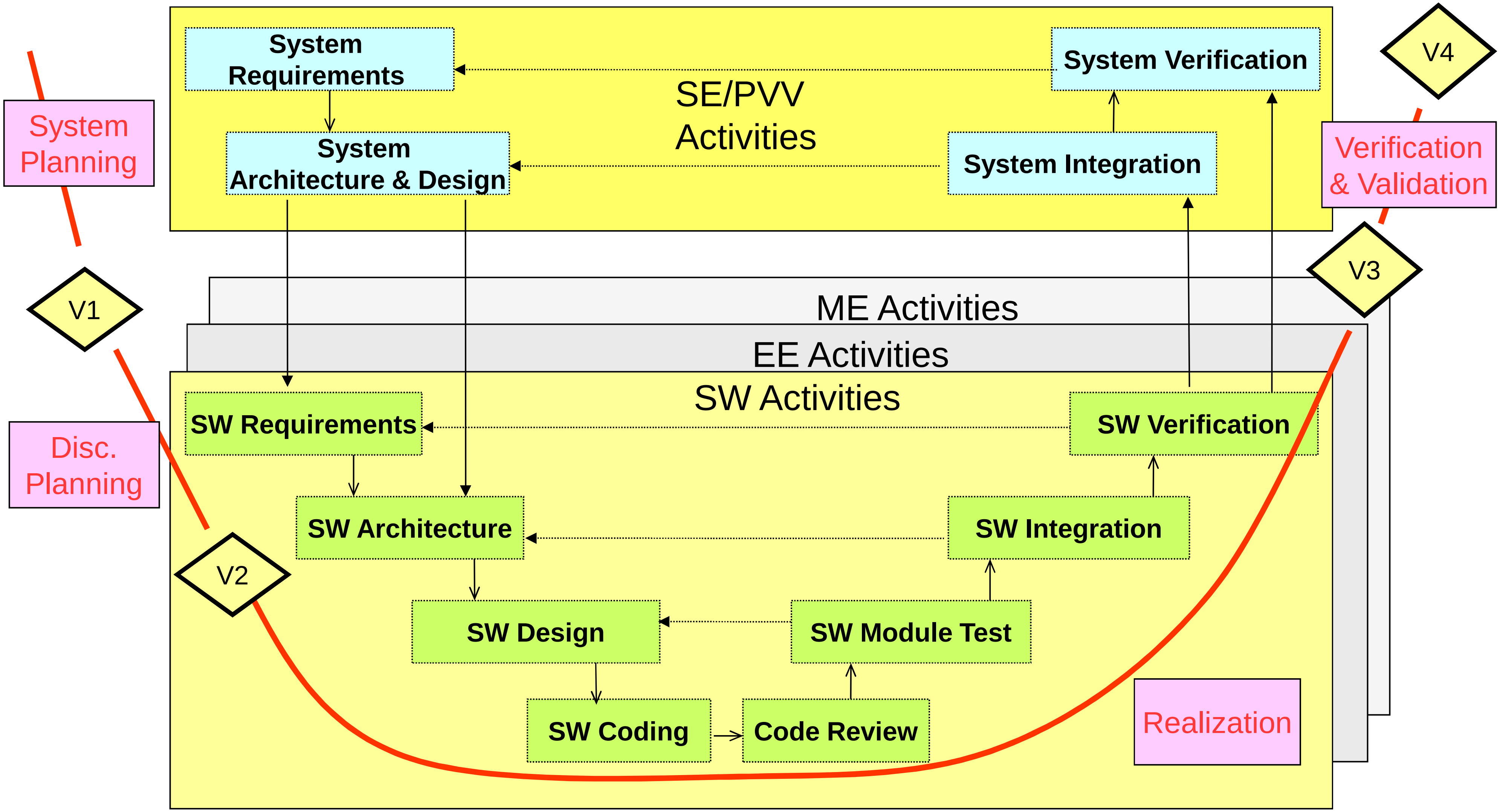
System Integr. Test
System Ver. Test
System Release Note
SE CM Audit

Lessons Learned

PVV Test Report

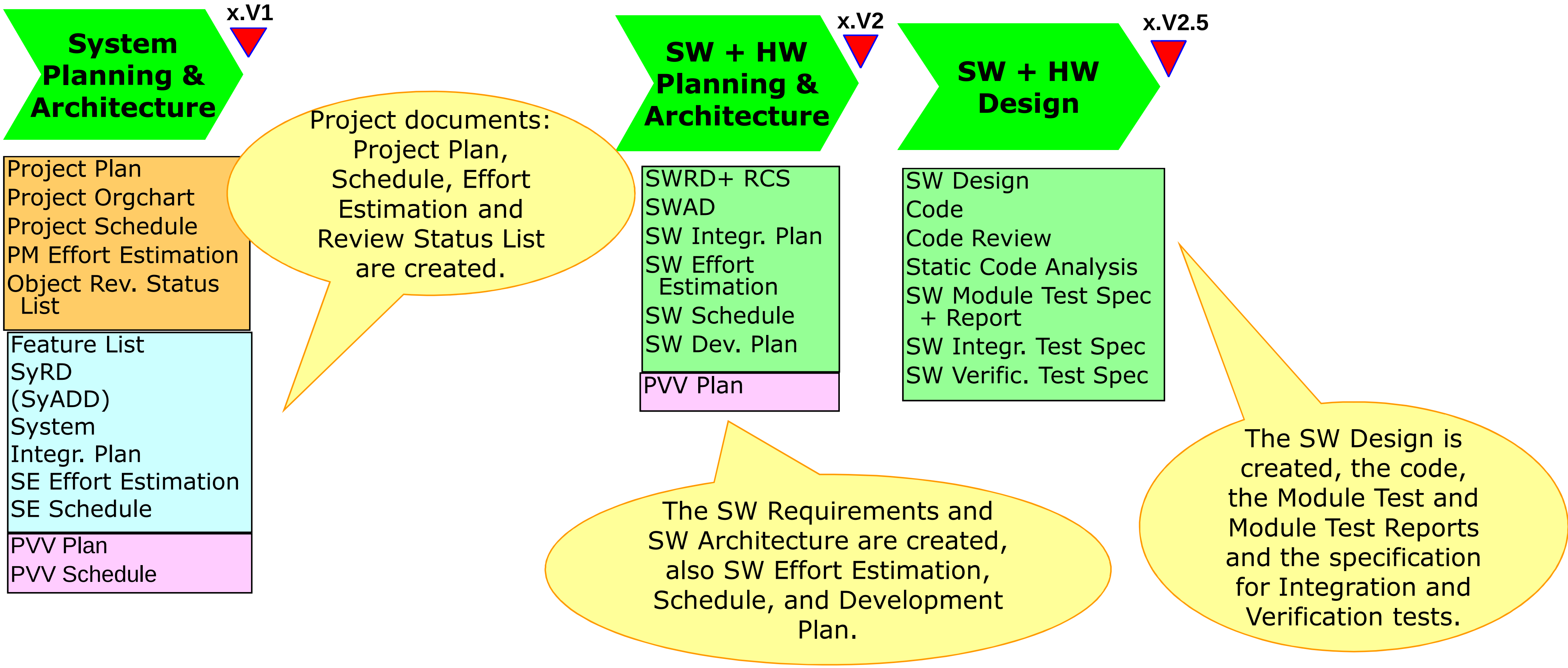
All activities related to design and development are completed.

# V-Cycle for all Increments



# Development Phase - 1

## Development



**System Planning & Architecture**

x.V1

Project Plan
Project Orgchart
Project Schedule
PM Effort Estimation
Object Rev. Status List

Feature List
SyRD
(SyADD)
System
Integr. Plan
SE Effort Estimation
SE Schedule

PVV Plan
PVV Schedule

Project documents: Project Plan, Schedule, Effort Estimation and Review Status List are created.

**SW + HW Planning & Architecture**

x.V2

SWRD+ RCS
SWAD
SW Integr. Plan
SW Effort Estimation
SW Schedule
SW Dev. Plan

PVV Plan

The SW Requirements and SW Architecture are created, also SW Effort Estimation, Schedule, and Development Plan.

**SW + HW Design**

x.V2.5

SW Design
Code
Code Review
Static Code Analysis
SW Module Test Spec + Report
SW Integr. Test Spec
SW Verific. Test Spec

The SW Design is created, the code, the Module Test and Module Test Reports and the specification for Integration and Verification tests.

# SW Requirements & SW Architecture

- In practice, **System Requirements are separated for individual components, including for software** (and categorized into **functional** and **non-functional**, like modifiability, stability and testability).
- The **impact on the operating environment is determined** (limited to the system components on which the software runs). An important question one may ask is *what kind of malfunctions the software may cause under certain operating conditions*. To find out, a situation analysis can be performed in the form of a particular type of risk workshop (Operability Study).
- The software requirements are translated into a software architecture (top level design) and documented.

  **Software Architecture** is used to illustrate **how software requirements are to be implemented in code** and **what the components are that the software consists of**. It describes function, mode of operation and interaction of the components.

→ It consist of one or several **block diagrams** illustrating the **software components** with **their inter-relationships**. These overview diagrams are supplemented with technical (e.g. interface) descriptions.

**Ideally**, **software architecture components remain stable during development** (so that there is no need to modify them or their interfaces in the remaining development phases).

- In practice, several views of software architecture are necessary:
  - → **Structure view** (architecture, used architectural patterns)
  - → **Behavior/ state view** (sleep mode, startup, shutdown, description of boot-block, monitoring techniques, etc)
  - → **Use–case view** (implementation of the use cases in the architecture)
  - → **Process view** (task design, timing, memory layout)
  - → **BIOS and service view** (CAN drivers, hardware abstraction layer, operating system integration, etc)
  - → **Call hierarchy** e.g. as a block diagram
  - → **Resources view** (planned RAM/ROM consumption)

# SW Design & SW Module Test Specifications

- The **SW Detailed Design** provides the **detailed specification of the SW items**, their **interactions**, **interface descriptions** (input and output data), **algorithms**, **assignment of memory space**, **data specification** and specification concerning the **program structure**.

  → Description includes flow charts, finite state machines, state charts, message sequence charts or data relationship models. It includes furthermore the definition of naming conventions, formats of required data structures, data fields and the purpose of each required data element.

- In **verifying the software units against the detailed design** the goal is to **prove** with reasonable effort that **the software meets its specifications** and that **risks related to the correct function of the SW units are kept within reasonable limits** (in practice, one cannot prove 0 defect software with verification methods, but only reduce risks).

- SW Module verification:

  → Unit test data: input values for a test and expected results and their inter-relationship with each other (e.g. timing), as well as target values after execution of the test case, including tolerances, response time, and so on. **This data is part of the description of a test case!**

  → Coding rules and guidelines: MISRA & rules for file naming conventions, file organization, commenting, naming conventions, declarations.

- **Consistency** of each developed **SW unit** with the **SW design** must be ensured. **Code reviews** and **reviews of the unit test cases** are suitable means for consistency checking.

# SW Integration & SW Verification Test Specifications

- The **integration strategy** specifies **the stepwise sequence in which the SW units are integrated** into increasingly larger, integrated SW items, taking into account the release strategy. From a certain level onwards, the integrated SW items correspond to elements of the SW Architecture.

→ The strategy should consider different **approaches** for the integration of new or changed SW items.

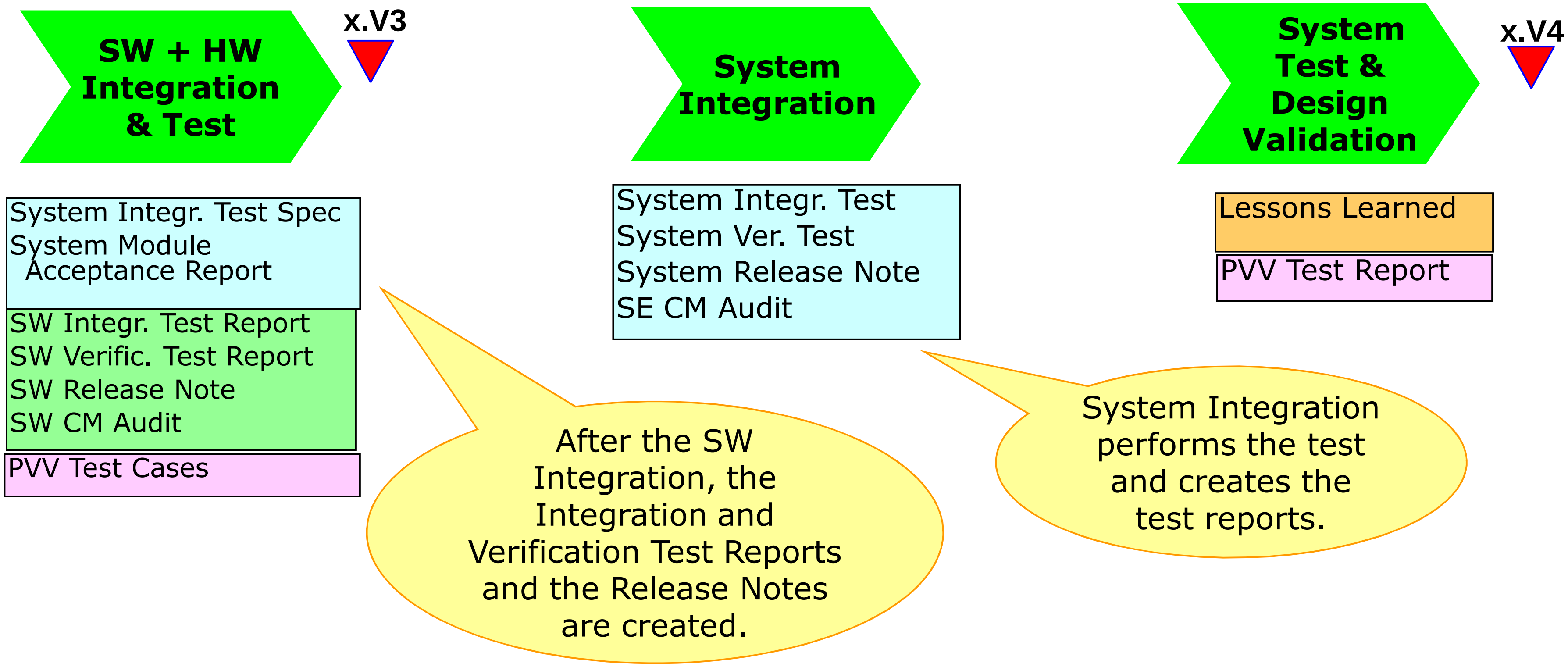
  - Bottom-up, starting with HW related SW
  - Top-down, starting with the user interface
  - Starting with a basic SW, then integration of critical modules
  - Integration in any sequence, e.g. according to availability
  - Integration of all parts in one single step (makes sense in “flat architectures”, with only a few functionally independent modules, integrated with a SW Base)

→ A specific integration sequence makes sense if one needs to follow the “onion layer principle” (functions in a layer can only be reasonable tested if the functions of the layers below already function reliably).

→ The integration strategy must also be compatible with priorities regarding SW requirements (e.g. release strategy)

- For each integrated SW unit the tests to be performed and method must be specified. This includes a description of the tests written in a way that **the tester knows exactly how the tests are to be executed**, also the **environmental set-up**, **operational procedures**, **input data** or data to be used, as well as a description of the **expected results or behavior**.
- **SW Integration Tests are done primarily on cross-module functions, interfaces, data flows**, etc. to prove that the design requirements have been met.
- **SW Verification test cases** must indicate additionally what requirement is examined by each test.  All of the tests must be appropriate to **demonstrate that all software requirements have been implemented**.

# Development Phase - 2



SW + HW Integration & Test

x.V3

System Integr. Test Spec
System Module
Acceptance Report

SW Integr. Test Report
SW Verific. Test Report
SW Release Note
SW CM Audit

PVV Test Cases

System Integration

System Integr. Test
System Ver. Test
System Release Note
SE CM Audit

System Test & Design Validation

x.V4

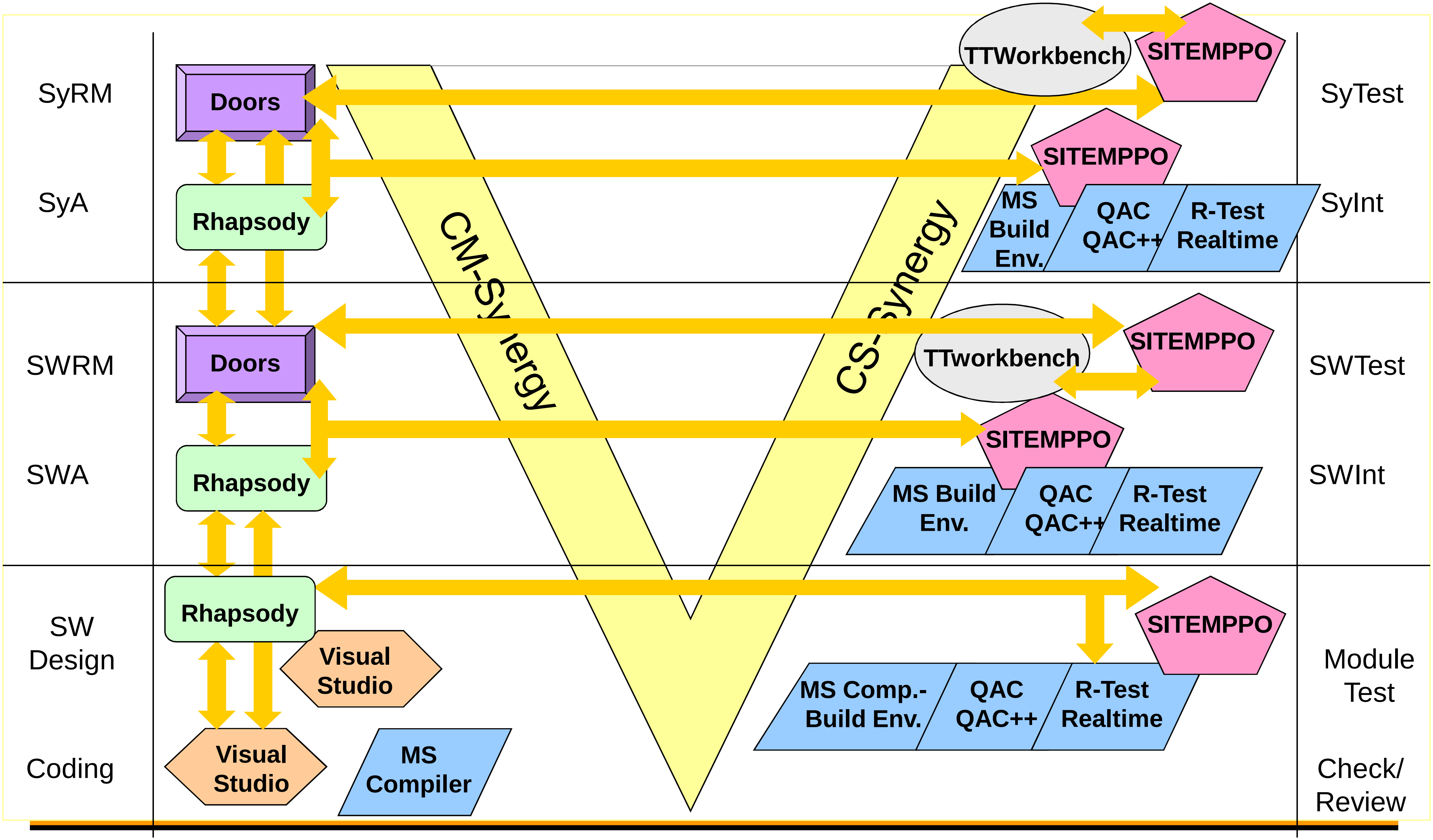
Lessons Learned

PVV Test Report

After the SW Integration, the Integration and Verification Test Reports and the Release Notes are created.

System Integration performs the test and creates the test reports.

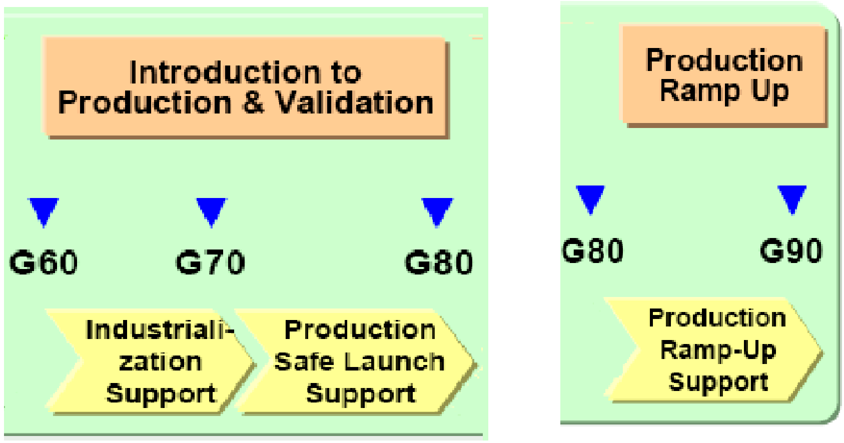
# Tool V-Cycle @ I MM



# Introduction to Production and Validation & Ramp-Up Phases

4

The project team must review the project to make sure it is ready for series production.



Introduction to Production & Validation
G60
G70
G80
Industriali-zation Support
Production Safe Launch Support
Production Ramp Up
G80
G90
Production Ramp-Up Support

5

The full manufacturing responsibilities is transferred to the receiving plant.

Regularly: PM Project Report, Action Item List, Risk List, PPR's

QM Production Launch Checklist

PPAP file & approval

Lessons Learned

Project Archived (Doc + Code)

Lessons Learned

SW Transfer to Maintenance

The product and process are ready for production.

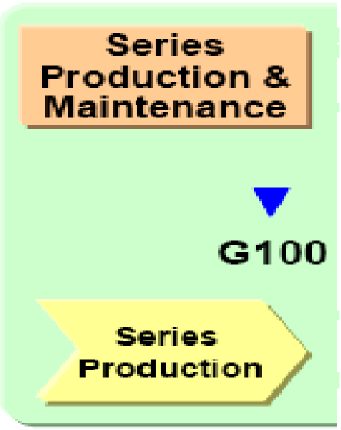
The project team reviews and compares the outcome of the project to the initial expectations.

# Series Production & Maintenance and Original Equipment Service

6

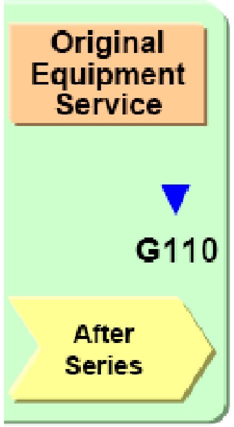
The production is ongoing to fulfill the market needs.

An annual improvement plan is created to address continuous quality, continued profitability and market growth.



<u>Maintenance Project:</u>

Update of:
QM Effort Estimation
QM Prod. Launch Checklist
PPAP file



<Support>

7

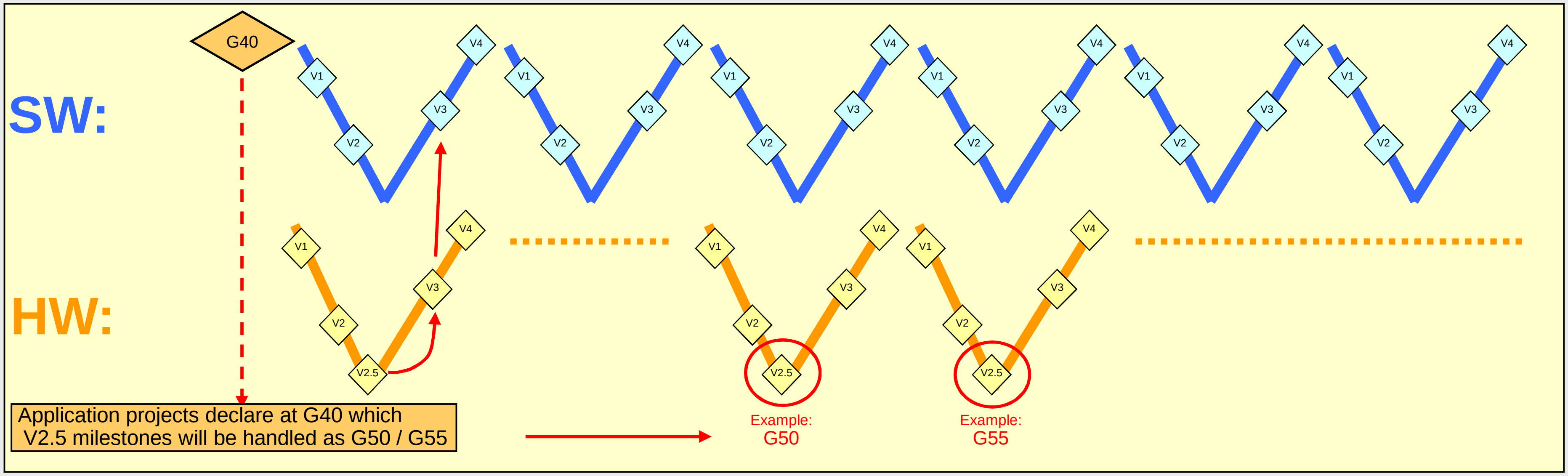
The post - series production activity required to meet the customer’s needs for service parts are covered.

Customer arrangements are finalized and disposal of equipment and tooling occurs. Upon management approval, all program activities are terminated.

# HW & SW V - Cycles

## <u>Relation between HW- and SW V-Cycles:</u>

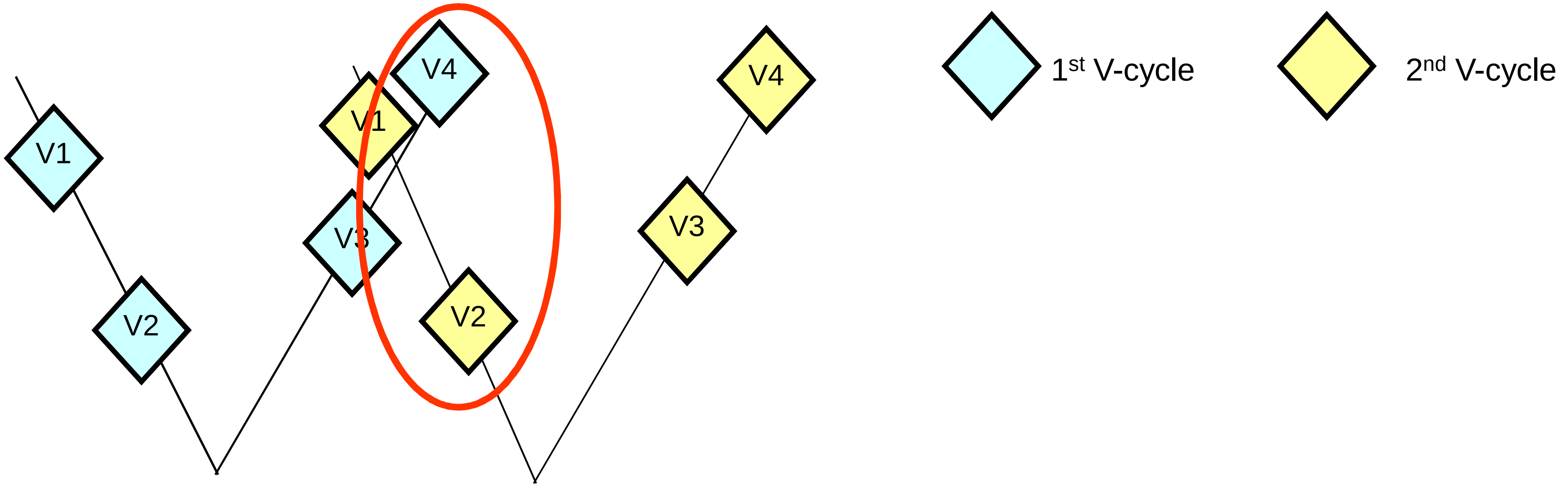
- HW lifecycle runs in parallel to SW lifecycle, if a new HW version is provided.
- 1st project V-cycle is always a SW <u>and</u> HW V-cycle.
- HW may skip a V-cycle, if no new HW version is provided.
- Projects define at G40 the number and timing of V-cycles.



Application projects declare at G40 which V2.5 milestones will be handled as G50 / G55

# Overlap of V-Cycles

- **Must:** V3 reached before V2 of next V-Cycle
- **Strongly recommended:** V4 reached before V2 of next V-cycle



# CMMI/SPICE

Short introduction

# Process Assessment

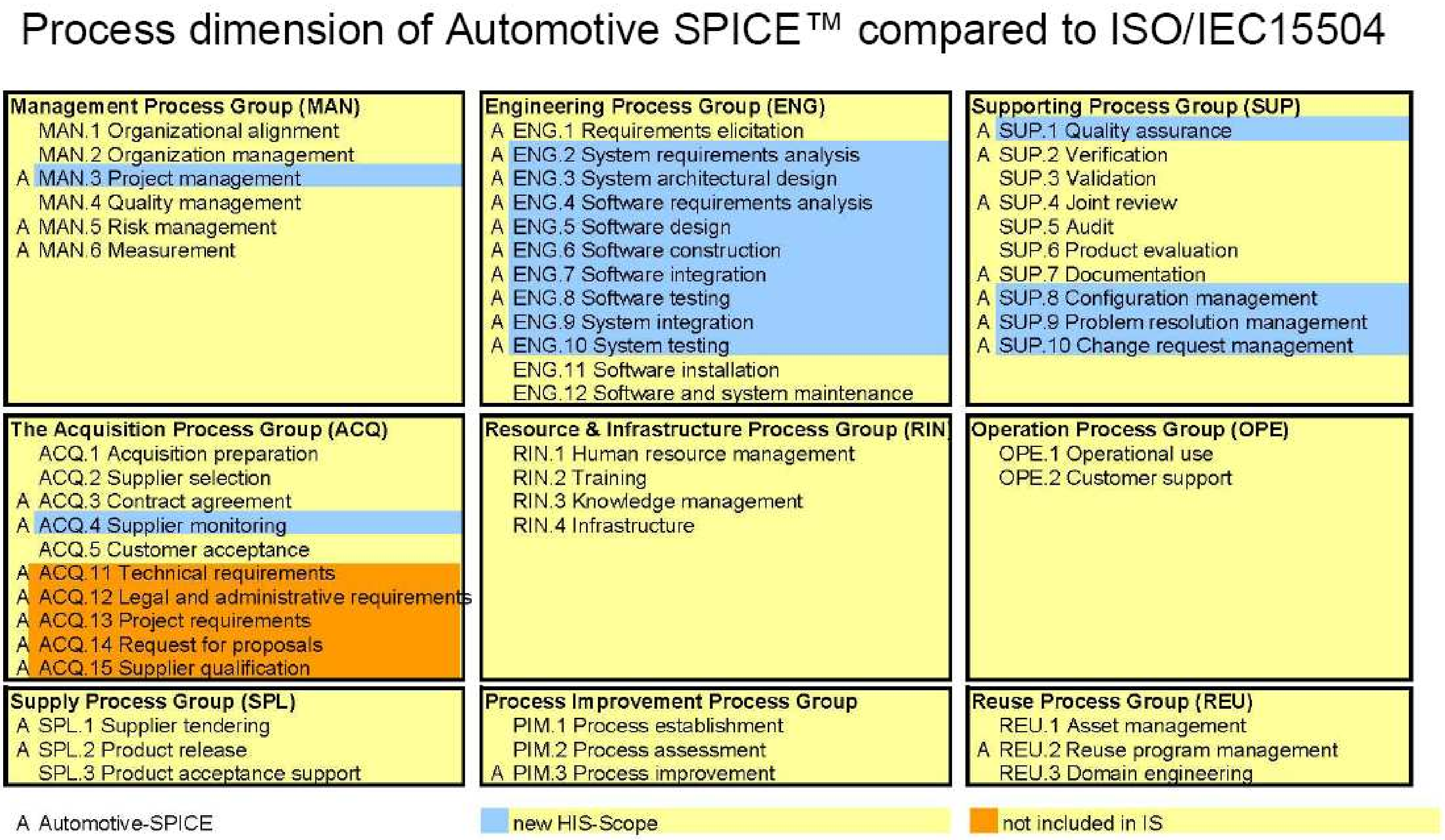
- Capability / maturity determination can be accomplished by performing process assessments using process assessment models such as:

CMMI = Capability Maturity Model Integration

SPICE = Software Process Improvement and Capability dEtermination

- the SPICE project uses ISO 15504 part 5 as an assessment model

# Automotive SPICE



## Process dimension of Automotive SPICE™ compared to ISO/IEC15504

**Management Process Group (MAN)**
- MAN.1 Organizational alignment
- MAN.2 Organization management
- A 🟦 MAN.3 Project management
- MAN.4 Quality management
- A MAN.5 Risk management
- A MAN.6 Measurement

**Engineering Process Group (ENG)**
- A ENG.1 Requirements elicitation
- A 🟦 ENG.2 System requirements analysis
- A 🟦 ENG.3 System architectural design
- A 🟦 ENG.4 Software requirements analysis
- A 🟦 ENG.5 Software design
- A 🟦 ENG.6 Software construction
- A 🟦 ENG.7 Software integration
- A 🟦 ENG.8 Software testing
- A 🟦 ENG.9 System integration
- A 🟦 ENG.10 System testing
- ENG.11 Software installation
- ENG.12 Software and system maintenance

**Supporting Process Group (SUP)**
- A 🟦 SUP.1 Quality assurance
- A SUP.2 Verification
- SUP.3 Validation
- A SUP.4 Joint review
- SUP.5 Audit
- SUP.6 Product evaluation
- A SUP.7 Documentation
- A 🟦 SUP.8 Configuration management
- A 🟦 SUP.9 Problem resolution management
- A 🟦 SUP.10 Change request management

**The Acquisition Process Group (ACQ)**
- ACQ.1 Acquisition preparation
- ACQ.2 Supplier selection
- A ACQ.3 Contract agreement
- A 🟦 ACQ.4 Supplier monitoring
- ACQ.5 Customer acceptance
- A 🟧 ACQ.11 Technical requirements
- A 🟧 ACQ.12 Legal and administrative requirements
- A 🟧 ACQ.13 Project requirements
- A 🟧 ACQ.14 Request for proposals
- A 🟧 ACQ.15 Supplier qualification

**Resource & Infrastructure Process Group (RIN)**
- RIN.1 Human resource management
- RIN.2 Training
- RIN.3 Knowledge management
- RIN.4 Infrastructure

**Operation Process Group (OPE)**
- OPE.1 Operational use
- OPE.2 Customer support

**Supply Process Group (SPL)**
- A SPL.1 Supplier tendering
- A SPL.2 Product release
- SPL.3 Product acceptance support

**Process Improvement Process Group**
- PIM.1 Process establishment
- PIM.2 Process assessment
- A PIM.3 Process improvement

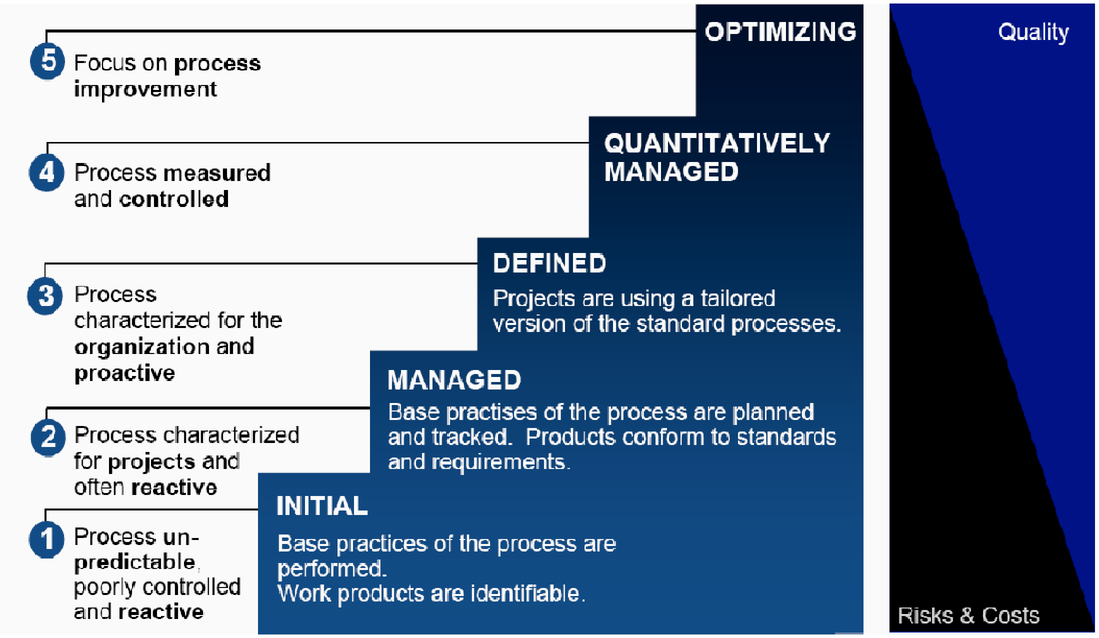
**Reuse Process Group (REU)**
- REU.1 Asset management
- A REU.2 Reuse program management
- REU.3 Domain engineering

A Automotive-SPICE

🟦 new HIS-Scope

🟧 not included in IS

# CMMI
# The Maturity Levels



5 Focus on process improvement

OPTIMIZING

4 Process measured and controlled

QUANTITATIVELY MANAGED

3 Process characterized for the organization and proactive

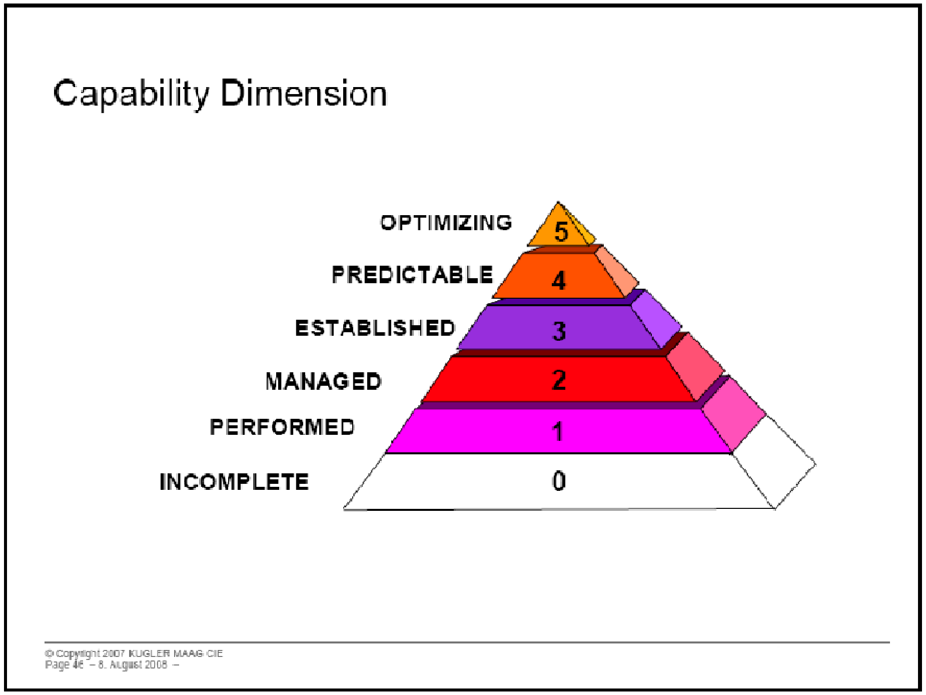
DEFINED
Projects are using a tailored version of the standard processes.

2 Process characterized for projects and often reactive

MANAGED
Base practises of the process are planned and tracked. Products conform to standards and requirements.

1 Process un-predictable, poorly controlled and reactive

INITIAL
Base practices of the process are performed.
Work products are identifiable.



Capability Dimension

© Copyright 2007 KUGLER MAAG CIE
Page 46 – 8. August 2008 –

# Training overview

- Processes – Definition & Motivation
- Software Quality Assurance
- PLC @ IMM Fundamentals
- V – Cycle Model
- CMMI/SPICE

Thank you!

Do you have any questions?