Dependable Systems

System Dependability Evaluation

Dr. Peter Tröger

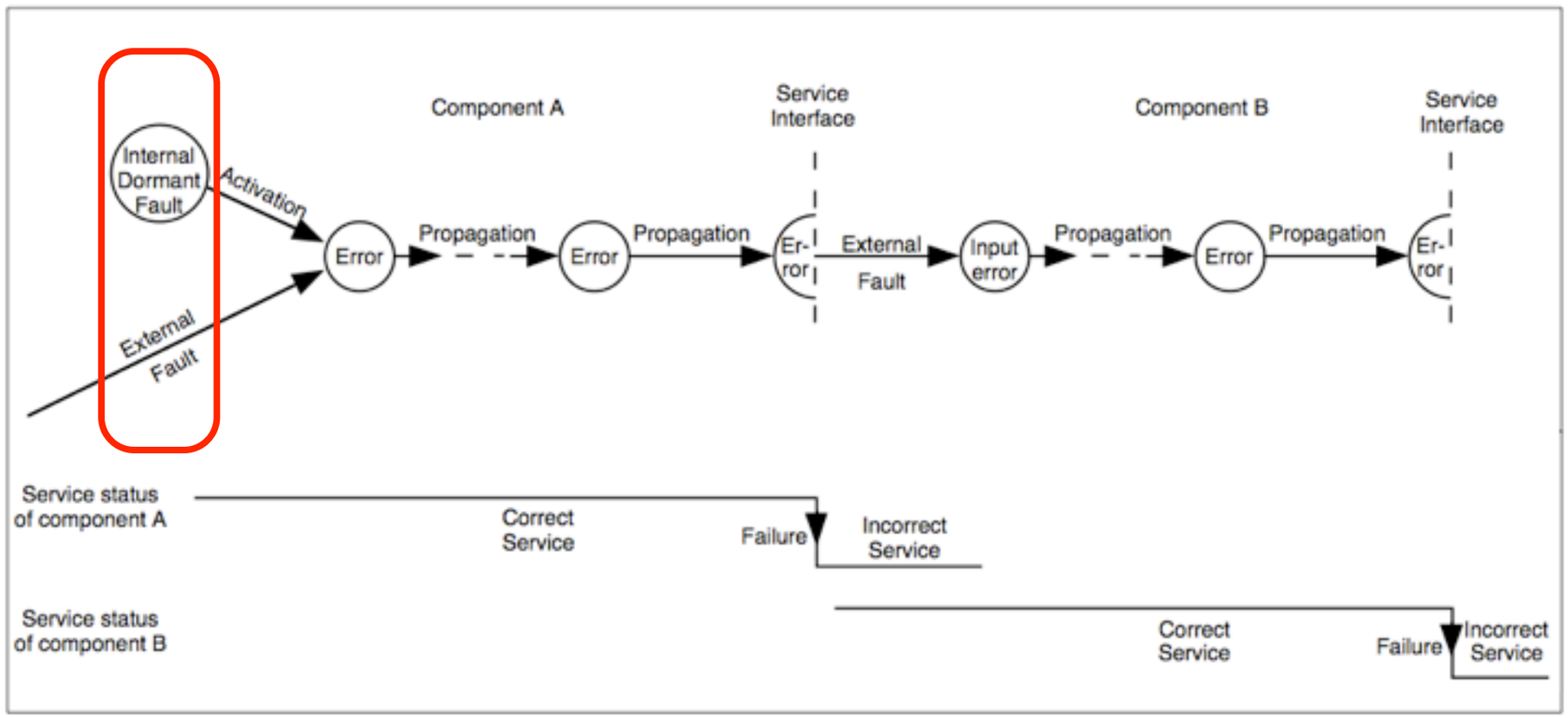
http://www.fmeainfocentre.com/handbooks/FMEA_Nasa_spacecraft.pdf
Alison Cartlidge et al.. An Introductory Overview of ITIL® V3. ISBN 0-9551245-8-1
Dr. Ralf Kneuer, Capability Maturity Model Integration (CMMI).

Qualitative Dependability Investigation

- Different approaches that focus on structural (qualitative) system evaluation
  - Root cause analysis
    - Broad research / industrial topic about error diagnosis approaches
    - Specialized topic in quality methodologies
  - Development process investigation
    - Procedures for ensuring industry quality in production
    - Software development process
- Organizational investigation
  - Non-technical influence factors on system reliability

Root Cause Analysis

- What caused the fault ? - Starting point of dependability chain
  - Peeling back the layers
  - Must be performed systematically as an investigation
  - Establish sequence of events / timeline



(C) Avizienis

Root Cause Analysis

- Class of approaches and algorithms for identifying the root cause of a problem
  - Iterative process of continuous improvement
  - Can be performed in reactive or pro-active fashion
- Applied in different fields, competing definitions and understandings
  - Accident analysis in safety-critical systems, e.g. aviation industry
  - Quality control in industrial manufacturing
  - Investigation of formally described business processes
  - Hardware failure analysis
  - Risk management for huge hardware / software projects
- Assumes broken process or alterable cause (e.g. no physical faults)

RCA: 5 Whys

- Originally developed by Sakichi Toyoda for car manufacturing process
  - Limit number of dive-in‘s to avoid tracing the chain of causality
- Example: The Washington Monument was disintegrating.
  - Why? Use of harsh chemicals.
  - Why? To clean pigeon poop.
  - Why so many pigeons?
    They eat spiders and there are a lot of spiders at monument.
  - Why so many spiders? They eat gnats and lots of gnats at monument.
  - Why so many gnats? They are attracted to the light at dusk.
  - Solution: Turn on the lights at a later time.

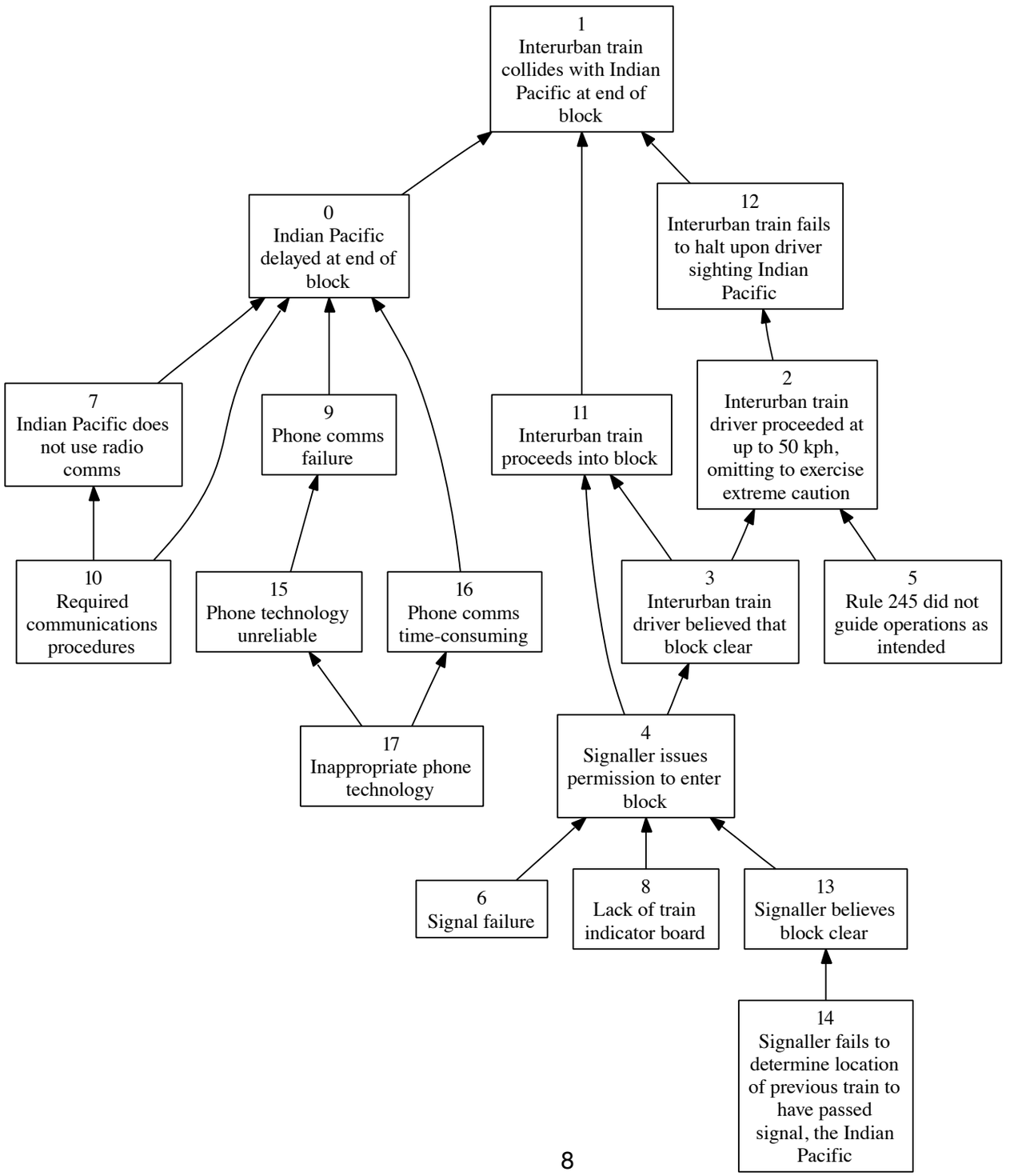
5 Whys

- Limited steps ensure that investigator moves through layers
- Has danger of stopping too early at symptoms
- Results are not reproducible
- Investigators cannot consider reasons behind their own information - need teams
- Recommendation to use observation instead of deduction
  - Deduction does not allow proper ranking of answers
- Depends on completely honest answers and complete problem statement
- Only good for low risk problems

RCA: Why-Because Analysis

- Mainly intended for accident analysis (train, bus, airplane, security, industry)
- Central notion of a causal factor
- Output is directed acyclic **Why-Because graph (WBG)**
  - Showing causal connections between all events and states of behavior
  - Nodes express causal factors, arcs express cause-effect relationship
  - Different subgraphs can be tested, results are combinable
    - **Causal sufficiency test** -
      Will an effect always happen if all parent causes happen ?
    - **Counterfactual test** (Dawid Lewis 1975, philosophical logician) -
      If the cause would not have existed, could the effect still have happened ?
      - If „no“ for two effects, then the cause is a **necessary causal factor** (NCF)

Example [Peter B. Ladkin]



RCA: Ishikawa / Fishbone Diagram

- Invented by Japanese quality control statistician for failure prevention
  - Identify (categorized) sources for variation
  - Analysis tool for systematic cause-and-effect analysis
    - List problem to be studied in the ‚head of the fish‘
    - Label main ‚fish bones‘
      - 4 M‘s: Methods (process), machines (technology), materials (raw, consumables), manpower (physical and brain work)
      - 4 P‘s: Place, procedure, people, policies
      - 4 S‘s: Surroundings, suppliers, systems, skills
    - Identify factors per category that may affecting the problem / issue
    - Repeat with sub-factors

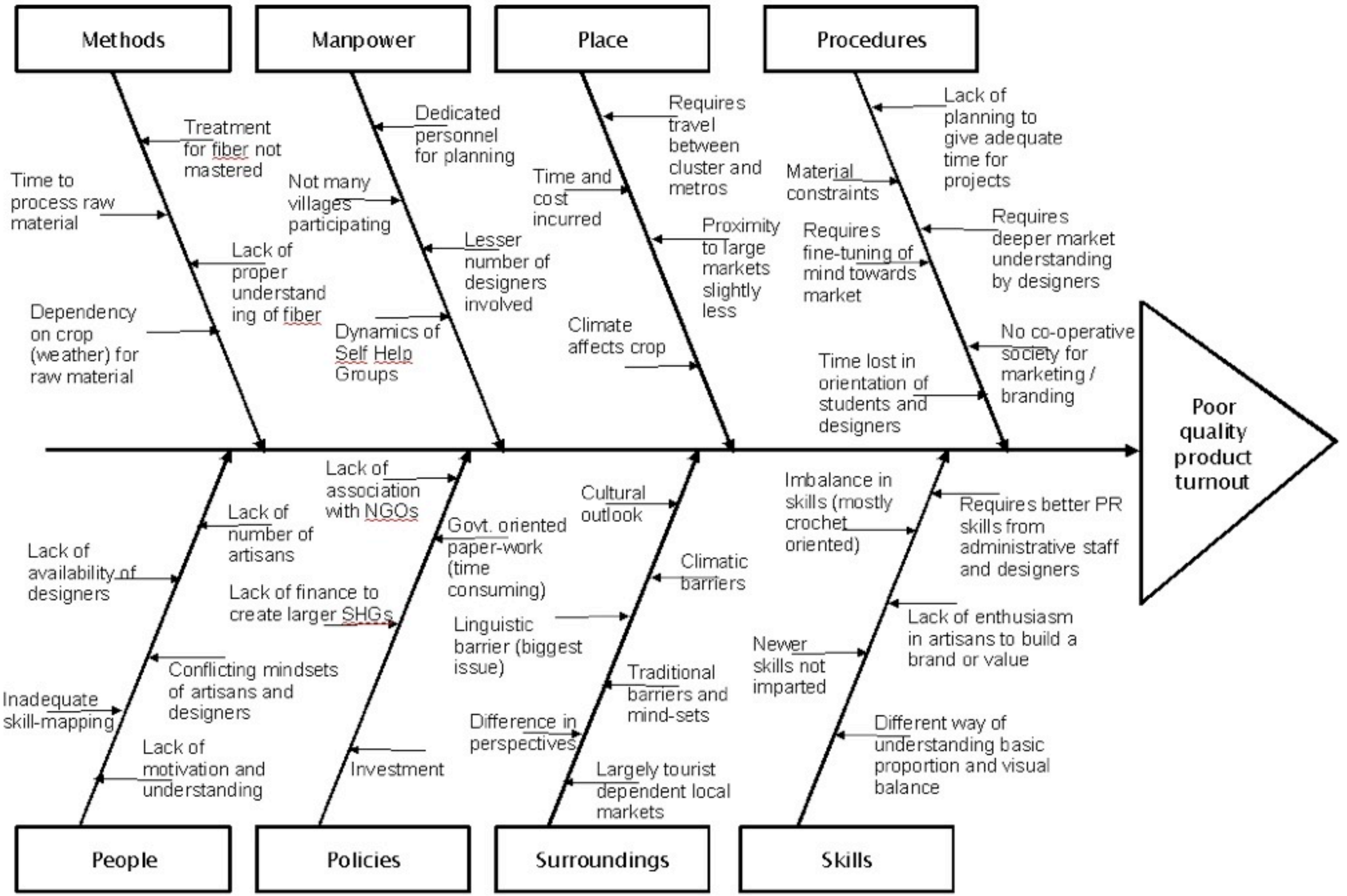
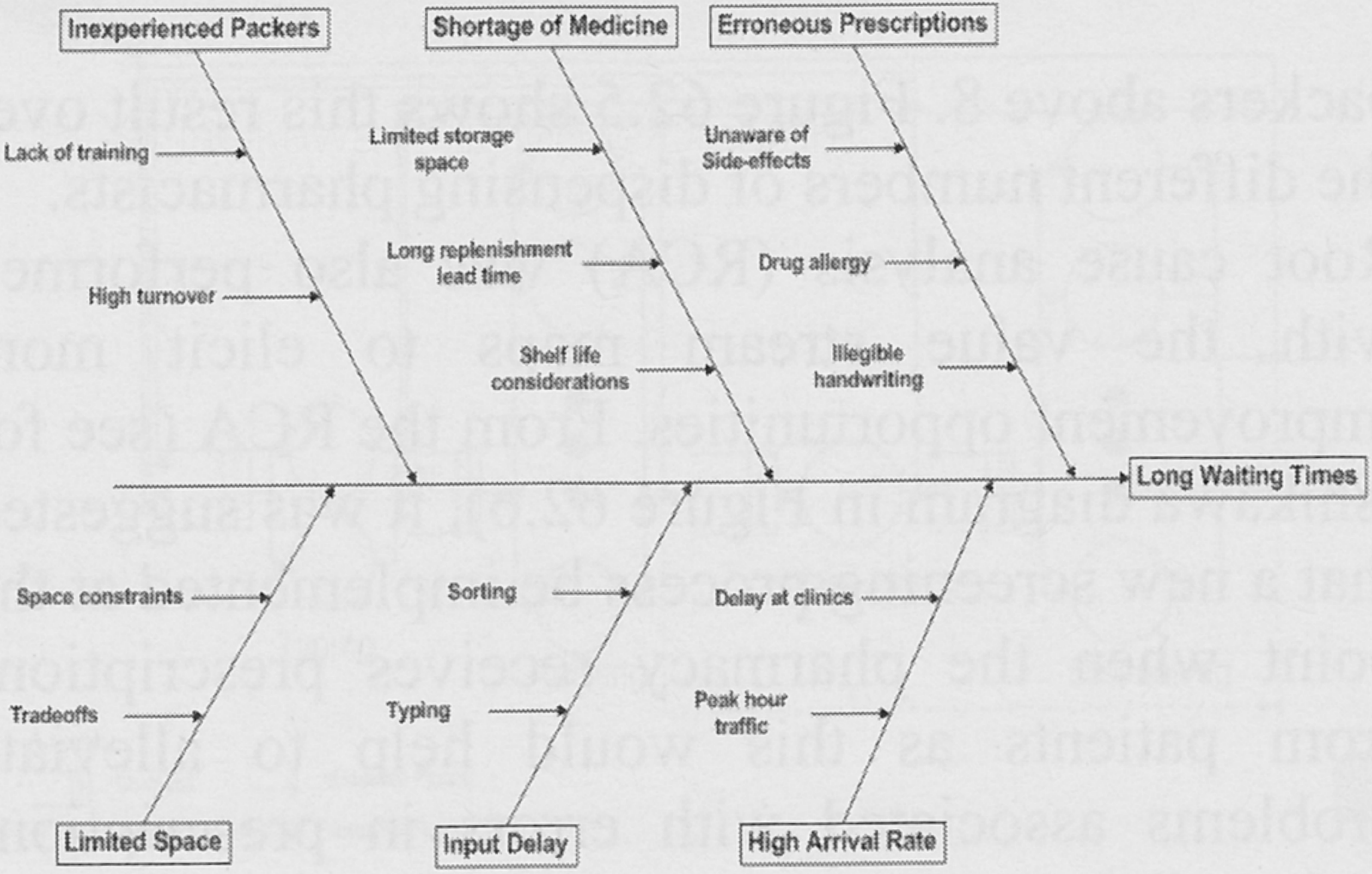


Fig: Application of Fish-bone diagram to craft based workshops in Indian design


Inexperienced Packers
Shortage of Medicine
Erroneous Prescriptions
Lack of training
Limited storage space
Unaware of Side-effects
High turnover
Long replenishment lead time
Drug allergy
Shelf life considerations
Illegible handwriting
Long Waiting Times
Space constraints
Sorting
Delay at clinics
Tradeoffs
Typing
Peak hour traffic
Limited Space
Input Delay
High Arrival Rate

FMEA

- **Failure Mode and Effects Analysis**
- Engineering quality method for early concept phase - identify and tackle weak points
- Introduced in the late 1940s for military usage (MIL-P-1629)
  - Later also used for aerospace program, automotive industry, semiconductor processing, software development, healthcare, ...
- Main goal is to identify and prevent critical failures
  - Identify activities to reduce the severity of such failures
  - Inductive analysis - starts at possible outcome, work backwards to all causes
- Used in many formal quality improvement programs

FMEA

- Main assumption: System is vulnerable to certain failure modes (== error states)
  - Examples: Electrical short-circuiting, corrosion, deformation
  - Identify relevant **failure mode** candidates based on past experience
  - **Effect analysis** for (specific) failure modes - what happens to the functionality visible to the end user ?
    - Examples: Degraded performance, noise, injury
  - Top-Down FMEA: Start with one function failure and find according failure modes
    - Typical for certification procedure, were the undesired functional problems are specified by the requirement documents
  - Bottom-Up FMEA: Find all the effects caused by all failure modes
- Failures are prioritized according to their consequences, how frequently they occur, and how easily they can be detected

FMEA Steps

1. Identify FMEA scope in cross-functional team (design, quality, testing, support, **...**)
2. Identify **functions** in the investigation scope (verb + noun)
3. Per function, identify all ways a failure could happen **- potential failure modes**
4. Identify **consequences** / **effects** per failure mode - on the system itself, related systems, product, service, customers or regulations
5. Perform **severity rating (S)** per effect - From insignificant to catastrophic, add only highest ranked effects to the further analysis
6. Identify **root causes** per failure mode, rank by **probability of occurrence (O)**
7. List tests / procedures (**process controls**) that are in place to keep the causes away
8. Determine **detection rating (D)** per control - from certain detection to no detection, based on level of process control installed
9. **Risk priority number (RPN)** = S ^{x} O ^{x} D, **Criticality (CRIT)** = S ^{x} O

Example: Bank FMEA

Assign persons during the analysis

Function	Potential Failure Mode	Potential Effects(s) of Failure	S	Potential Cause(s) of Failure	O	Current Process Controls	D	RPN	CRIT	Recommended Action(s)	Responsibility and Target Completion Date	Action Results					
												Action Taken	S	O	D	RPN	CRIT
Dispense amount of cash requested by customer	Does not dispense cash	Customer very dissatisfied Incorrect entry to demand deposit system Discrepancy in cash balancing	8	Out of cash	5	Internal low-cash alert	5	200	**40**								
				Machine jams	3	Internal jam alert	10	240	24								
				Power failure during transaction	2	None	10	160	16								
	Dispenses too much cash	Bank loses money Discrepancy in cash balancing	6	Bills stuck together	2	Loading procedure (riffle ends of stack)	7	84	12								
				Denominations in wrong trays	3	Two-person visual verification	4	72	18								
	Takes too long to dispense cash	Customer somewhat annoyed	3	Heavy computer network traffic	7	None	10	210	21								
				Power interruption during transaction	2	None	10	60	6								

RPN prioritizes differently from criticality

(C) asq.org

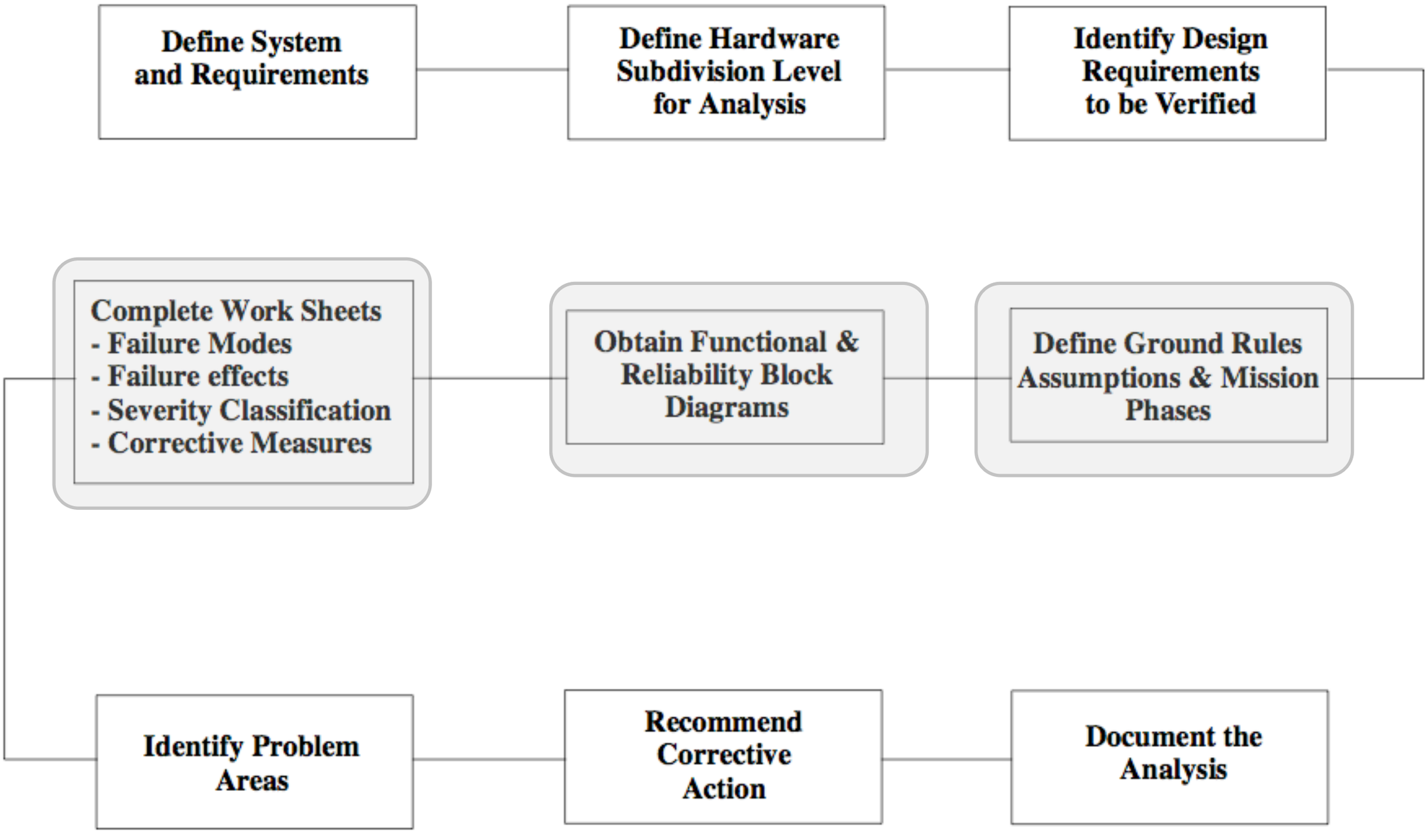
Example: NASA Spacecraft Severity Ratings (S)

- Category 1, **Catastrophic** - Failure modes that could result in serious injury or loss of life, or damage to the launch vehicle.
- Category 1R, **Catastrophic** - Failure modes of identical or equivalent redundant hardware items that, if all failed, could result in Category 1 effects.
- Category 2, **Critical** - Failure modes that could result in loss of one or more mission objectives as defined by the GSFC project office.
- Category 2R, **Critical** - Failure modes of identical or equivalent redundant hardware items that could result in Category 2 effects if all failed.
- Category 3, **Significant** - Failure modes that could cause degradation to mission objectives.
- Category 4, **Minor** - Failure modes that could result in insignificant or no loss to mission objectives.

Example: NASA Spacecraft FMEA
Ground Rules for Failure Mode Analysis

- Only one failure mode exists at a time.
- All inputs (including software commands) to the item being analyzed are present and at nominal values.
- All consumables are present in sufficient quantities.
- Nominal power is available.
- All mission phases are considered in the analysis; mission phases that prove inapplicable may be omitted.
- Connector failure modes are limited to connector disconnect.
- Special emphasis will be directed towards identification of single failures that could cause loss of two or more redundant paths.

Example: NASA Spacecraft FMEA
Flow Diagram of Overall System Analysis



FMEA Variations

- DFMEA (D: Design)
  - Focus on failure modes reasoned by design deficiencies
  - Focus on parts that can be prototyped before high volume production
- PFMEA (P: Process)
  - Analyze manufacturing and assembling processes
  - Influence design of machinery, selection of tooling and component parts
- Do not mix up design failures and causes („incorrect material specified“) with process failures and causes („incorrect material used“)
- FMEA typically makes use of fault tree modeling

Hazard & Operability Studies (HAZOPS)

- Process for identification of potential hazard & operability problems, caused by **deviations from the design intend**
  - Difference between deviation (failure) and its cause (fault)
  - Conduct intended functionality in the safest and most effective manner
  - Initially developed to investigate chemical production processes, meanwhile for petroleum, food, and water industries
  - Extended for complex (software) systems
- Qualitative technique
  - Take full description of process and systematically question every part of it
  - Assess possible deviations and their consequences
  - Based on **guide-words** and multi-disciplinary meetings

HAZOPS

- First identify system entities and their attributes (e.g. state diagrams)
- Use of keywords to focus the attention
  - **Primary keywords** - Focus attention on a particular aspect of the design intent / process condition / investigated parameter
    - Examples for plants:
      flow, pressure, react, corrode, temperature, level, mix, absorb, erode, ... isolate, vent, inspect, start-up, shutdown, purge, maintain, ...
    - Examples for Java Language Definition analysis:
      Type default value, type value range, name scope, class modifier, class name, field modifier, field type, method modifier, method name, formal parameter, ...
  - **Secondary keywords** - When combined with a primary keyword, suggest possible deviations of the system from the design intent
    - Standardized list, not all combinations are appropriate

Plant Example - Secondary Keywords

Word	Meaning
No	The design intent does not occur (e.g. Flow/No), or the operational aspect is not achievable (Isolate/No)
Less	A quantitative decrease in the design intent occurs (e.g. Pressure/Less)
More	A quantitative increase in the design intent occurs (e.g. Temperature/More)
Reverse	The opposite of the design intent occurs (e.g. Flow/Reverse)
Also	The design intent is completely fulfilled, but in addition some other related activity occurs (e.g. Flow/Also indicating contamination in a product stream)
Other	The activity occurs, but not in the way intended (e.g. Flow/Other could indicate a leak or product flowing where it should not)
Fluctuation	The design intention is achieved only part of the time (e.g. an air-lock in a pipeline might result in Flow/Fluctuation)
Early	Usually used when studying sequential operations, this would indicate that a step is started at the wrong time or done out of sequence
Late	

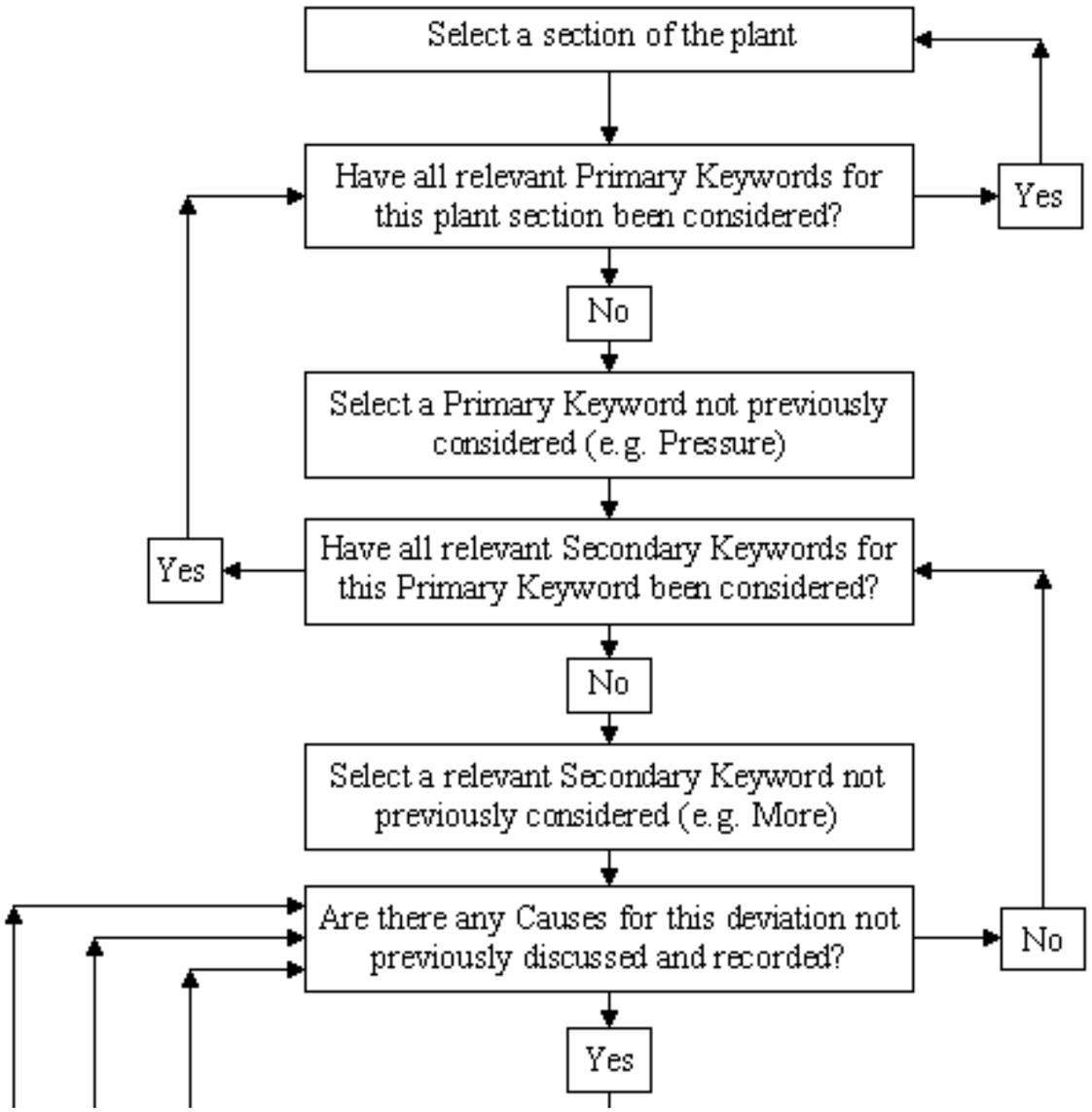
Plant Example - Combinations [Wikipedia]

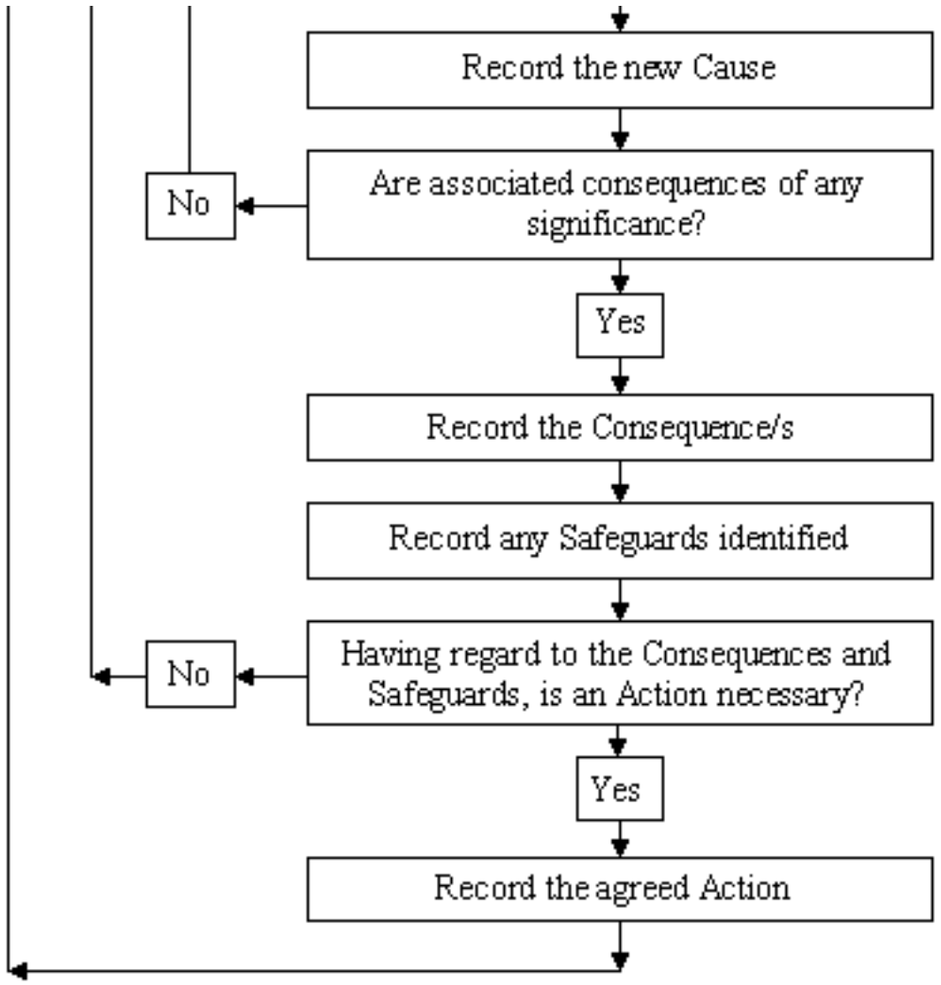
Parameter / Guide Word	More	Less	None	Reverse	As well as	Part of	Other than
Flow	high flow	low flow	no flow	reverse flow	deviating concentration	contamination	deviating material
Pressure	high pressure	low pressure	vacuum		delta-p		explosion
Temperature	high temperature	low temperature					
Level	high level	low level	no level		different level		
Time	too long / too late	too short / too soon	sequence step skipped	backwards	missing actions	extra actions	wrong time
Agitation	fast mixing	slow mixing	no mixing				
Reaction	fast reaction / runaway	slow reaction	no reaction				unwanted reaction
Start-up / Shut-down	too fast	too slow			actions missed		wrong recipe
Draining / Venting	too long	too short	none		deviating pressure	wrong timing	
Inertising	high pressure	low pressure	none			contamination	wrong material
Utility failure (instrument air, power)			failure				

HAZOPS for Java Language Specification [Kim, Clark, McDermid] - Secondary Keywords

Word	Meaning
No	No part of the intention is achieved. No use of syntactic components
More	A quantitative increase, the data value is too high (within or out of bounds)
Less	A quantitative decrease, the data value is too low (within or out of bounds)
As Well As	Specific design intent is achieved but with additional results
Part Of	Only some of the intention is achieved, incomplete
Reverse	Reverse flow - flow of information in wrong direction, iteration count modified in wrong direction, logical negation of condition
Other Than	A result other than the original intention is achieved, complete but incorrect
Narrowing	Scope or accessibility is narrower than intended.
Widening	Scope or accessibility is enlarged
Equivalent	The same design intent is achieved in a different way (without any side effect)

HAZOPS Procedure





(C) www.lihoutech.com

HAZOPS Documentation

- Apply in a systematic way all relevant keyword combinations to the design
- Per combination, record
  - Deviation (keyword combination)
  - Cause (potential causes for the deviation)
  - Consequence (from both the deviation and the cause)
    - Should not take credit for protective systems or instruments in the design, since not all operational conditions are clarified at this point
  - Safeguards (which prevents the cause or safeguards the system against it)
    - Can be hardware, software, or procedures
  - Actions (which either remove the cause or mitigate the consequences)

HAZOPS for Java Language Specification [Kim, Clark, McDermid] - Deviants Example

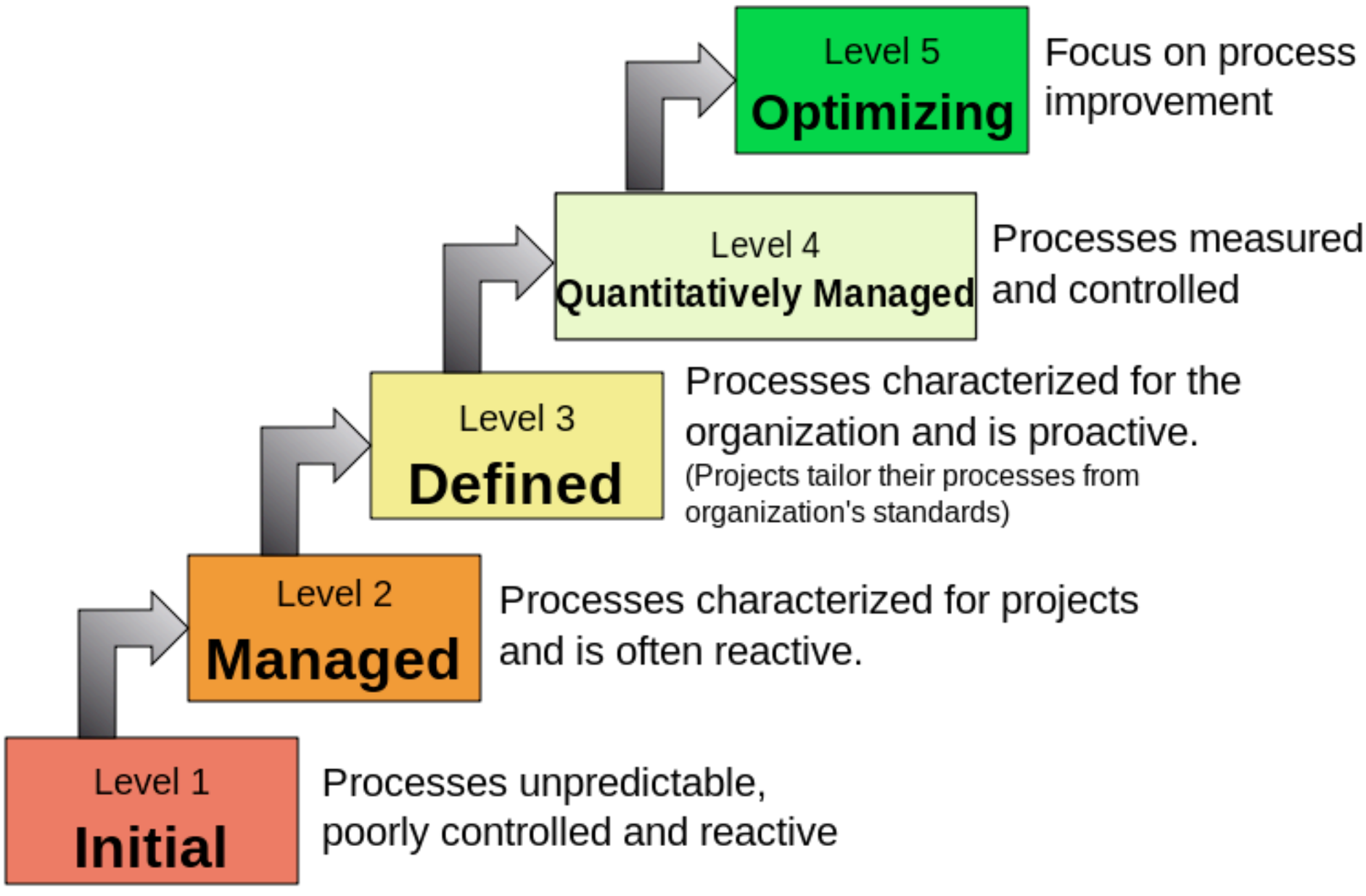
FieldDeclaration:
FieldModifiers $_{opt}$ ***Type VariableDeclarators ;***

Attribute: Field modifiers	Guideword: OTHER_THAN
Causes	• The modifier `static` is specified where the field should be an instance variable. • The modifier `static` is not specified where the field should be a class (static) variable.
Consequences	A field becomes a class variable instead of an instance variable (or vice versa), causing different results or compile errors.
Attribute: Field modifiers	Guideword: MORE
Causes	The number of the specified modifiers increases.
Consequences	The behaviour of a field is changed/restricted or compile-time error occurs.
Attribute: Field modifiers	Guideword: LESS
Causes	The number of the specified modifiers decreases. (e.g. from 2 modifiers to 1 or no modifier)
Consequences	The behaviour of a field is changed or compile-time error occurs.
Attribute: Accessibility	Guideword: WIDENING
Causes	An access modifier is changed from `protected` to `public`, or from `private` to `public`.
Consequences	The fields that were not accessible become accessible. For example when a field is changed from `private` to default access, any entities within the same package can access the field either by ***SimpleName*** or ***QualifiedName***.
Attribute: Type compatibility of a field type	Guideword: AS_WELL_AS
Causes	• Class type T is declared instead of class type S, provided that S is a subclass of T. • Interface type K is used instead of interface type J, provided that J is a subinterface of K.
Consequences	Widening reference conversion regards a reference as having some other type.
Attribute: Type compatibility of a field type	Guideword: PART_OF
Causes	Class type S instead of class type T is declared, provided that S is a subclass of T.
Consequences	`ClassCastException` may arise if the actual reference value is not a legitimate value of the declared type at run time.

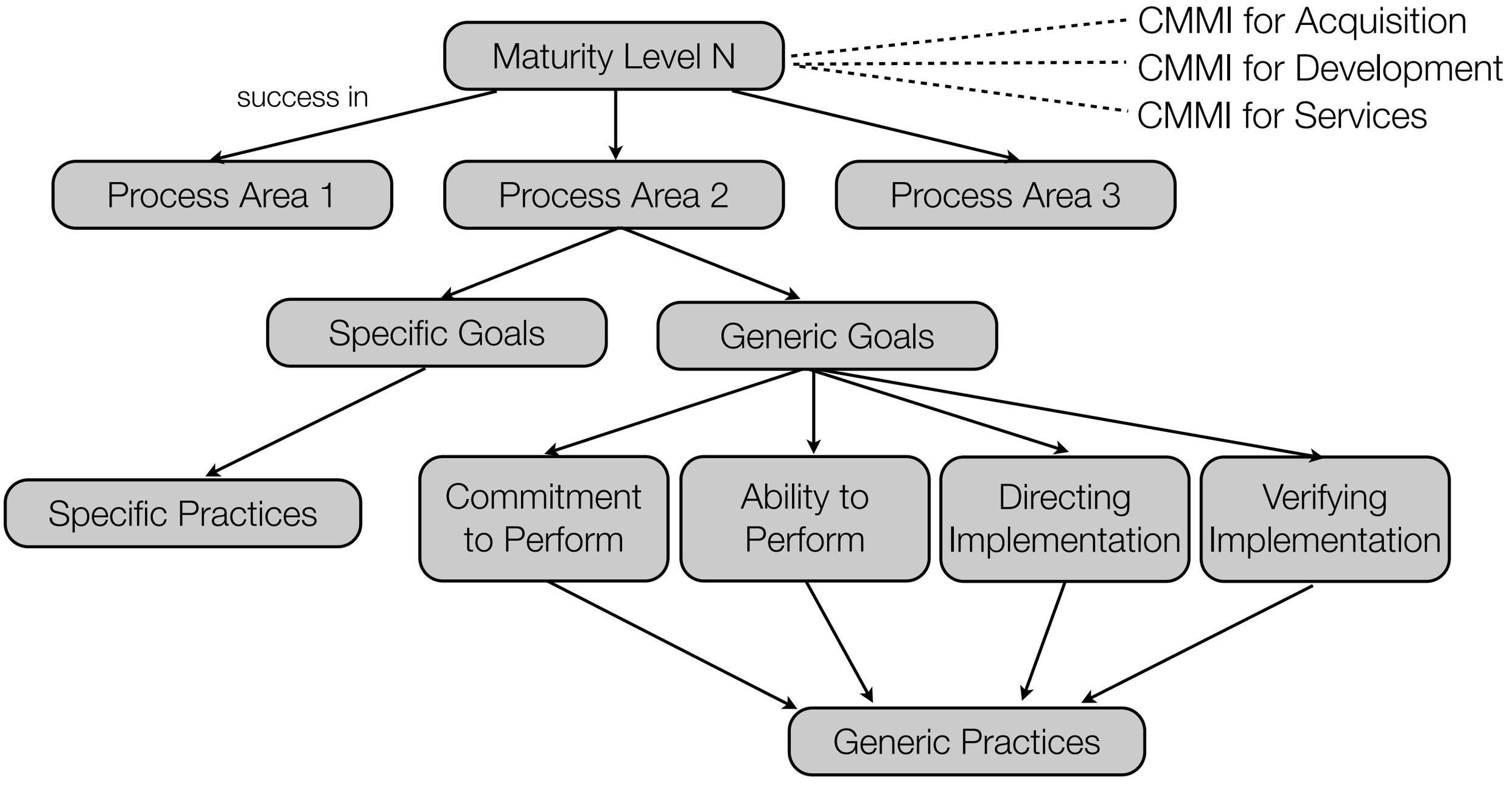
Software Process Evaluation and Improvement

- Software reliability can be derived from level of trust into the development process
- Relevant for software supplier evaluation in public bidding
- Most famous approach is the **Capability Maturity Model (CMM)** by CMU Software Engineering Institute (SEI), extended to **CMMI**
  - Customer wants: Completion in time, budget, and functionality with high quality
  - Management wants: High customer satisfaction and productivity, control
  - Application areas for CMMI
    - System engineering
    - Software engineering
    - Integrated product and process development
    - Supplier sourcing

CMMI Maturity Levels



Capability Maturity Model Integration - Structural Overview

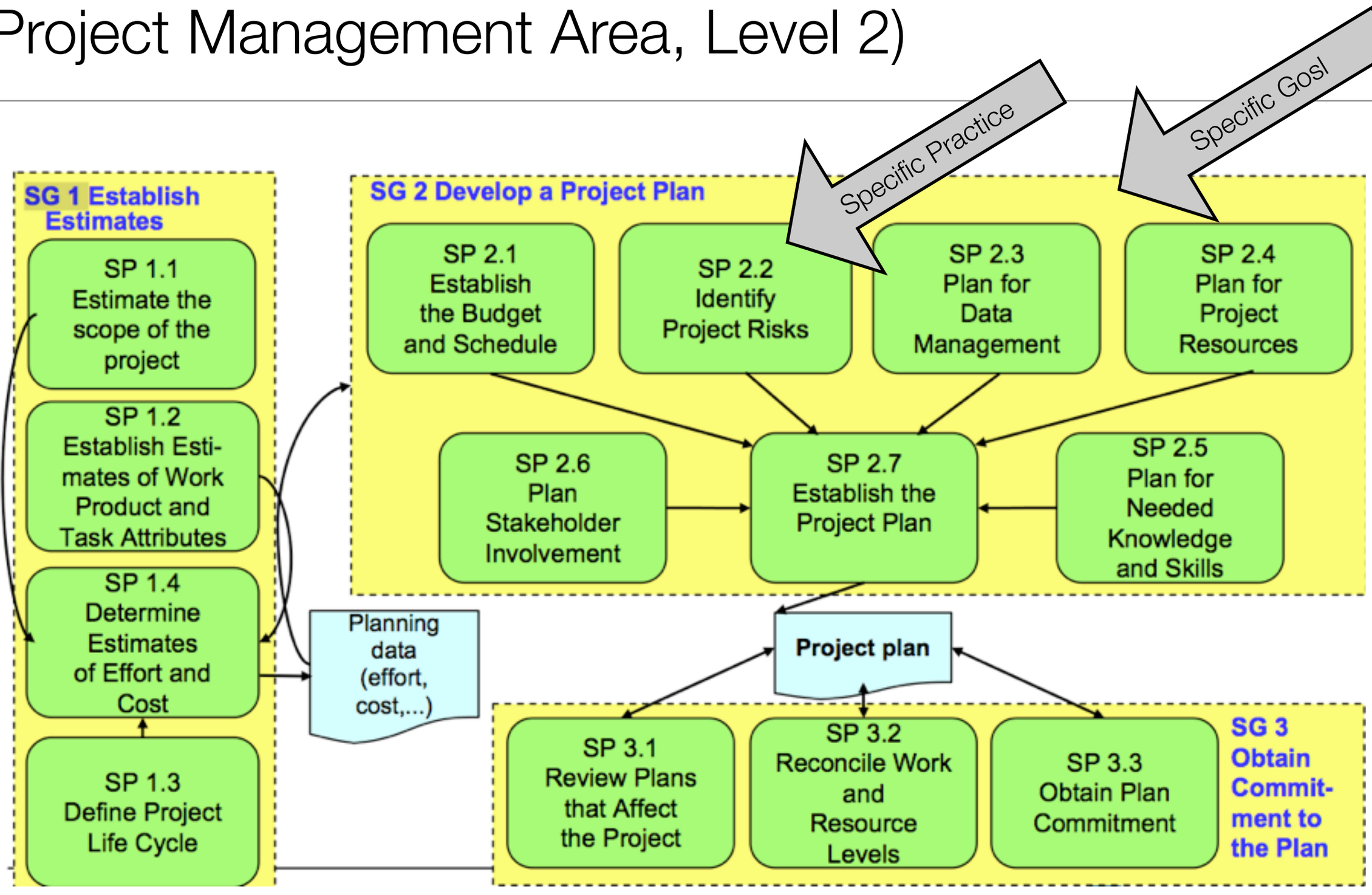


Capability Maturity Model Integration

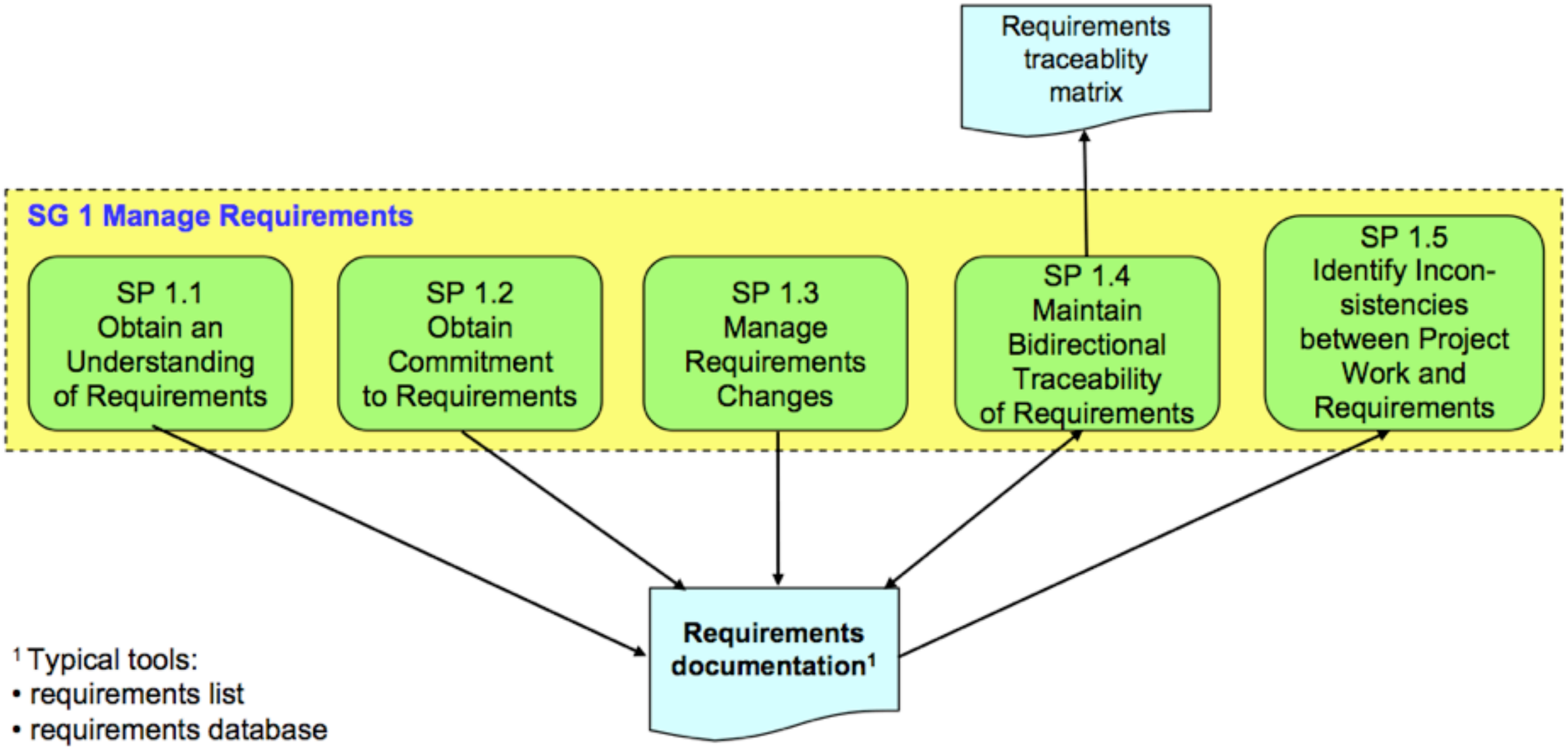
- Maturity level consists of the results in multiple process areas
  - Latest version 1.3 contains 22 process areas

	Process Mgmt.	Project Management	Engineering	Support
2		Project Planning (PP) Project Monitoring and Control (PMC) Supplier Agreement Management (SAM)	Requirements Management (REQM)	Configuration Management (CM) Process & Product Quality Assurance (PPQA) Measurement and Analysis (MA)
3	Organizational Process Focus (OPF) Organizational Process Definition (OPD) Organizational Training (OT)	Integrated Project Management (IPM) Risk Management (RSKM)	Requirements Development (RD) Technical Solution (TS) Product Integration (PI) Verification (VER) Validation (VAL)	Decision Analysis and Resolution (DAR)
4	Organizational Process Performance (OPP)	Quantitative Project Management (QPM)		
5	Organizational Innovation and Deployment (OID)			Causal Analysis and Resolution (CAR)

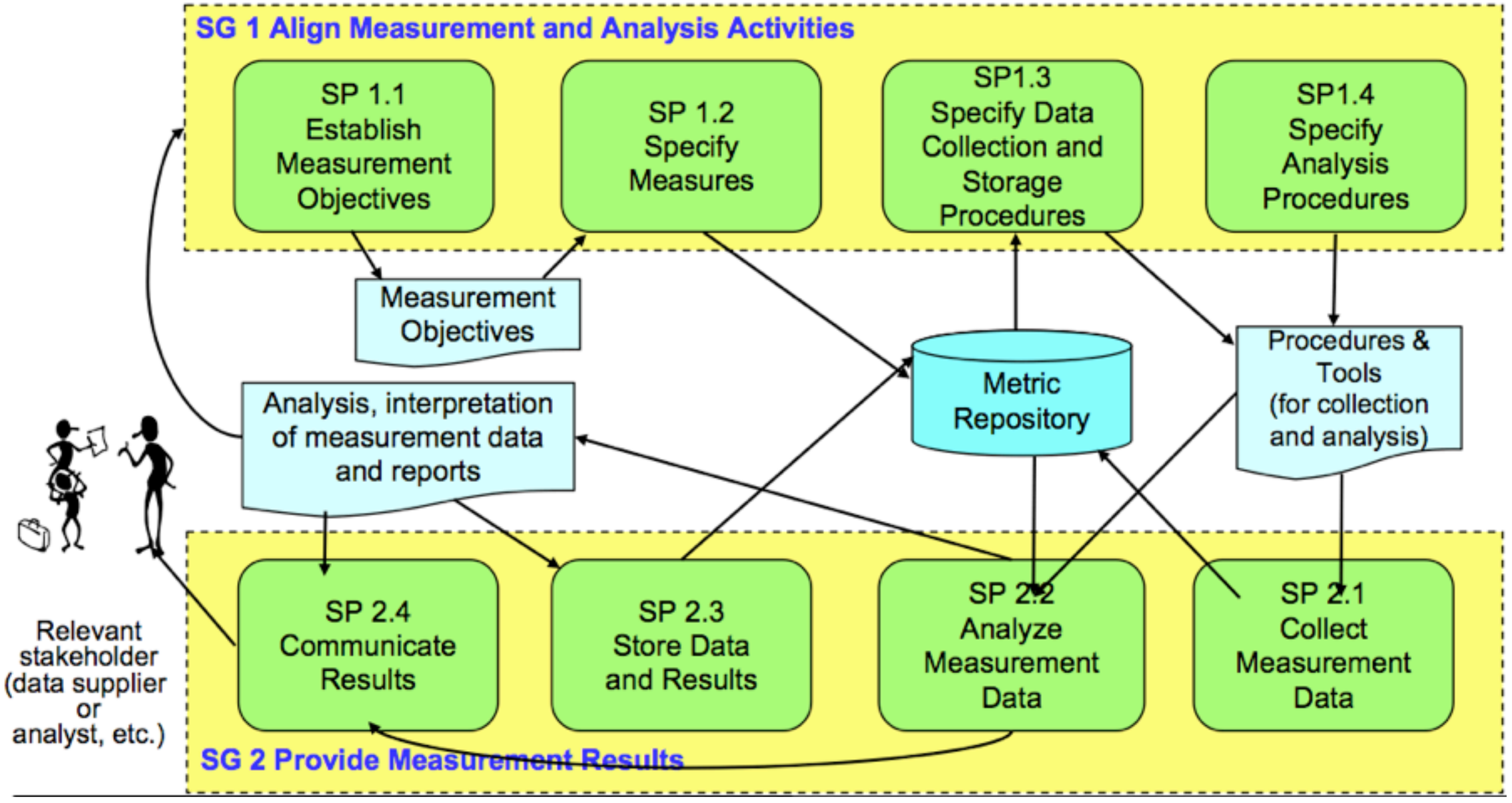
Example: Specific Goals in Project Planning (Project Management Area, Level 2)



Example: Specific Goals in Requirements Management (Engineering Area, Level 2)



Example: Specific Goals in Measurement and Analysis (Support Area, Level 2)



Reliability Models for IT Infrastructures

- System reliability in a commercial environment is determined by many factors:
  - Software and hardware reliability
  - Training of maintenance personnel
  - „Business processes“ how maintenance is handled
  - The way the IT department is organized
  - ...
- Impact of management organization on reliability is an emerging research field
- Standards for IT organization, based on best practices
  - Describe which processes have to be established in an IT department
  - Provide reference models for organization of IT department

Standards

- ISO/IEC 20000: IT Service Management
- ISO/IEC 27001: Information Security Management
- Capability Maturity Model Integration (CMMI®)
- Control Objectives for Information and related Technology (COBIT®)
- Projects in Controlled Environments (PRINCE2®)
- Project Management Body of Knowledge (PMBOK®)
- Management of Risk (M_o_R®)
- eSourcing Capability Model for Service Providers (eSCM-SPTM)
- Telecom Operations Map (eTOM®)
- Six Sigma™.

Six Sigma

- Coined by Motorola engineer (1986), meanwhile Motorola trademark
  - Started as simple statistical technique to reduce defects in manufacturing
  - Improve quality by improving manufacturing processes
- Embraced as company quality improvement strategy (Sony, Honda, TI, Canon, ...)

- Lower and upper boundary level for process results
  -> should be far away from mean quality of the process or service

- Prior to Six Sigma, upper and lower limits were defined by standard three-sigma rule
  - For a normal distribution, 99.73% of the values lie within three standard deviations
  - Used as rough probability estimate for values (e.g. outlier detection)

$$\mu + 6\sigma \text{ and } \mu - 6\sigma$$

μ: process mean
σ: standard deviation of the process

Six Sigma

- Management framework for processes, techniques, and training
- Data-driven systematic evaluation approach
  - Asks tougher and tougher questions until quantifiable answers are received
  - Measure of process capability, related to defect rate and complexity of a product
    - *Defect:* Any process output that does not meet customer expectation
- Advantage: Well defined targets by specification of statistical metrics
  - Popular among manufacturing and service organizations
- Quality improvement does not automatically lead to financial benefit
  - Mathematical models for chosing an optimal improvement strategy
  - Might demand process tweaking or replacement

Six Sigma Metrics

- *Yield (Y)*: Measure of process capability
  - Ratio of defect free units vs. units produced (‚opportunities‘)

$$Y = \frac{Defect\ free\ units}{Number\ of\ opportunities} \times 100\%$$

- *Defects per Million Opportunities (DPMO)*
  - Typically estimated from a sample of units - defects per unit (DPO)

$$DPMO = DPU \times 10^6 = \frac{Number\ of\ defects}{Number\ of\ opportunities} \times 10^6$$

- *Sigma Quality Level:* Measure of quality for the output produced by an organization
  - Level directly related to DPMO metric

Six Sigma Levels

- Level 6 allows less that four (3.4) *defects per million opportunities (DPMO)*
- Practice: Mean expected to move up to 1.5 standard devations over time
  - Experience that processes get worser in the long term
  - Assumption that initial 6-sigma process will degrade to 4.5-sigma
  - Designed to prevent under-estimation of defect levels
- Does not reflect product specifics (pace maker vs. mass mail advertiser)

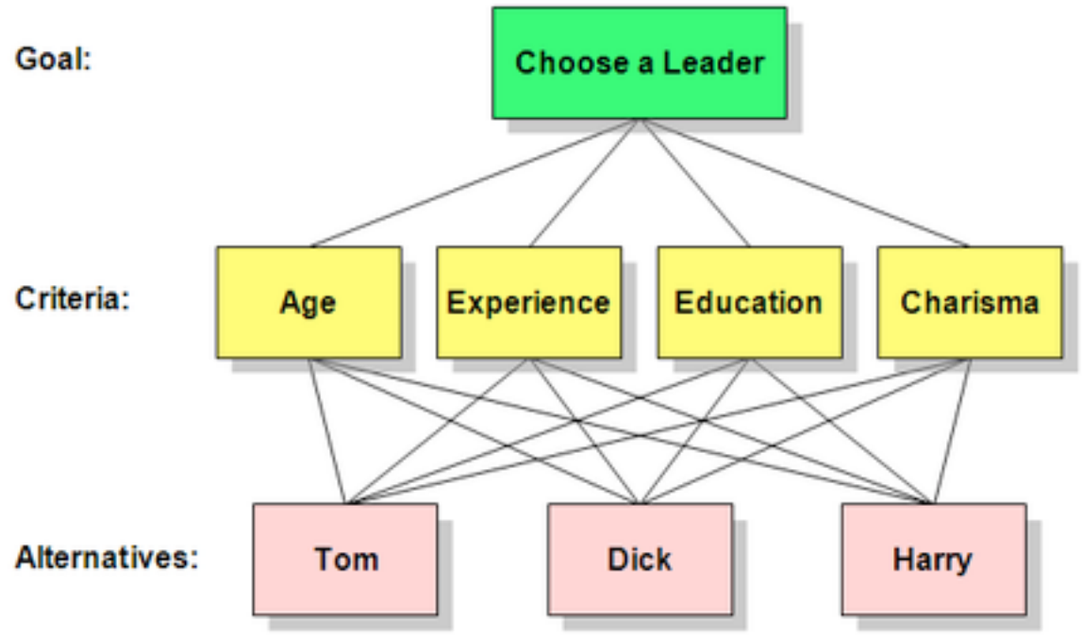
Level	DPMO	Defective	Yield
1	691462	69 %	31 %
2	308538	31 %	69 %
3	66807	6,7 %	93,3 %
4	6210	0,62 %	99,38 %
5	233	0,023 %	99,977 %
6	3,4	0,00034 %	99,99966 %
7	0,019	0,0000019 %	99,9999981 %

Six Sigma Project Selection

- Each process improvement opportunity is treated as *project*
- Criterias for choice [Pande]
  - *Business benefits:* Impact on customers, business strategy, finances; urgency
  - *Feasibility:* Resources needed, expertise available, complexity, success probability
  - *Organization impact:* Learning and cross-functional benefits
  - Also: Top management commitment
- Project selection itself should follow a strategy
- Once project is chosen, walk through methodology for improvement

Six Sigma Project Selection

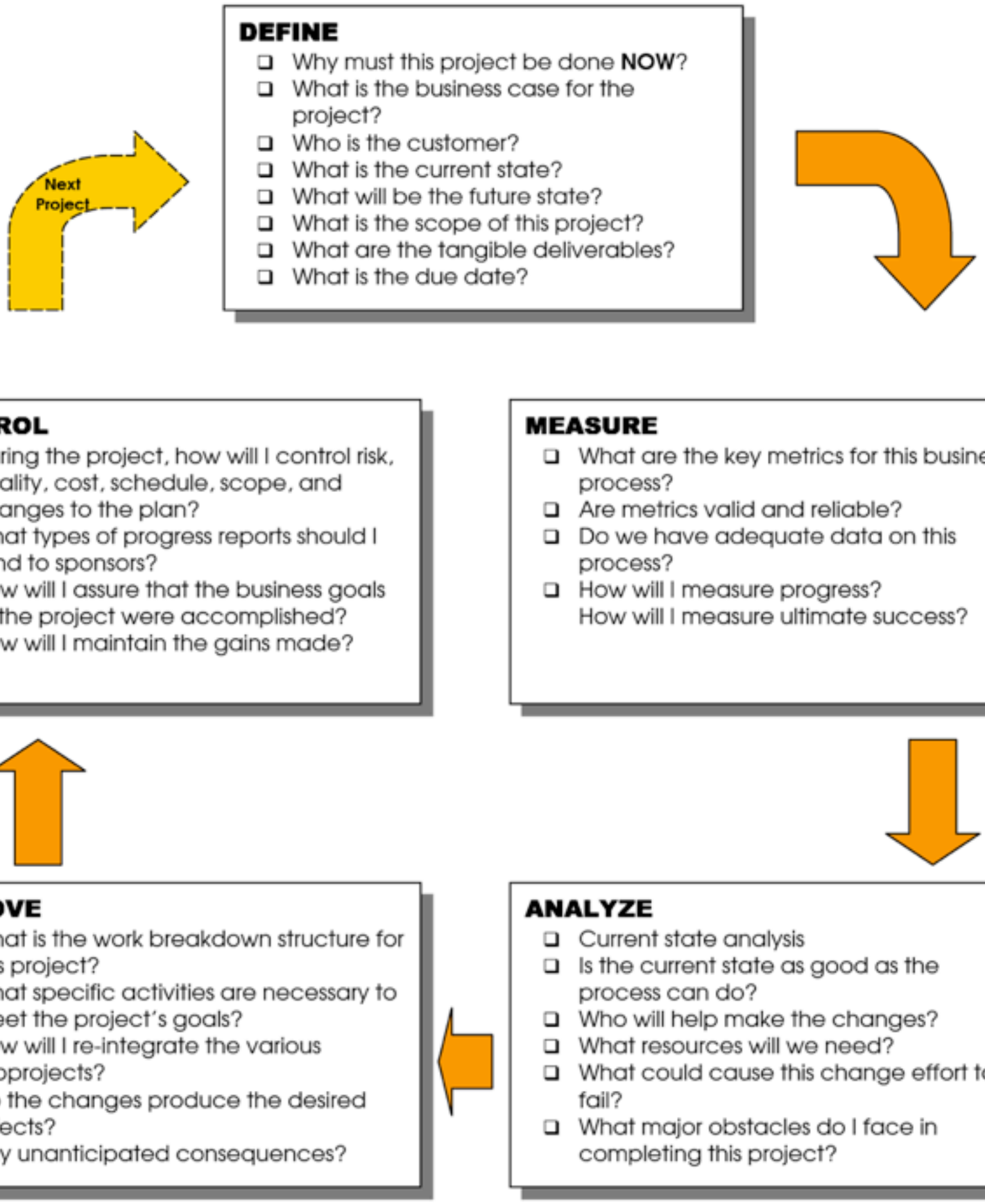
- *Analytic hierarchy process (AHP)* project selection strategy [Kumar]
- Assign weights to *criterias* for projects based on pair-wise comparison:
  - Duration for completion
  - Costs of project
  - Probability of success
  - Strategic fit of the project
  - Increase in customer satisfaction
  - Increase in Six Sigma quality level
  - Reduction of *cost of poor quality (CoPQ)*
  - Manpower requirement (green belts and black belts)



AHP Explained [Wikipedia]

DMAIC Methodology

- Aimed at improving existing process by using Six Sigma (‚project‘)
- 5 stages in a cycle
  - **Define**: Identify the problem in terms of deficiencies in the *critical to quality (CTQ)* parameter
  - **Measure**: Define and use metrics to rank process capability, get gap to target
  - **Analyze**: Determine cause-effect relationship between process performance gaps (measured in CTQ) and process inputs
    - Examples: 5 Whys, data mining, experiment design, FMEA, ...
  - **Improve**: Implement solution for identified problem from ‚define‘ stage
  - **Control**: Sustain the improvement
- Strength in tool box due to mathematical foundation


Next Project
DEFINE
❑ Why must this project be done NOW?
❑ What is the business case for the project?
❑ Who is the customer?
❑ What is the current state?
❑ What will be the future state?
❑ What is the scope of this project?
❑ What are the tangible deliverables?
❑ What is the due date?
MEASURE
❑ What are the key metrics for this business process?
❑ Are metrics valid and reliable?
❑ Do we have adequate data on this process?
❑ How will I measure progress?
How will I measure ultimate success?
ANALYZE
❑ Current state analysis
❑ Is the current state as good as the process can do?
❑ Who will help make the changes?
❑ What resources will we need?
❑ What could cause this change effort to fail?
❑ What major obstacles do I face in completing this project?
IMPROVE
❑ What is the work breakdown structure for this project?
❑ What specific activities are necessary to meet the project's goals?
❑ How will I re-integrate the various subprojects?
❑ Do the changes produce the desired effects?
❑ Any unanticipated consequences?
CONTROL
❑ During the project, how will I control risk, quality, cost, schedule, scope, and changes to the plan?
❑ What types of progress reports should I send to sponsors?
❑ How will I assure that the business goals of the project were accomplished?
❑ How will I maintain the gains made?

Six Sigma Roles

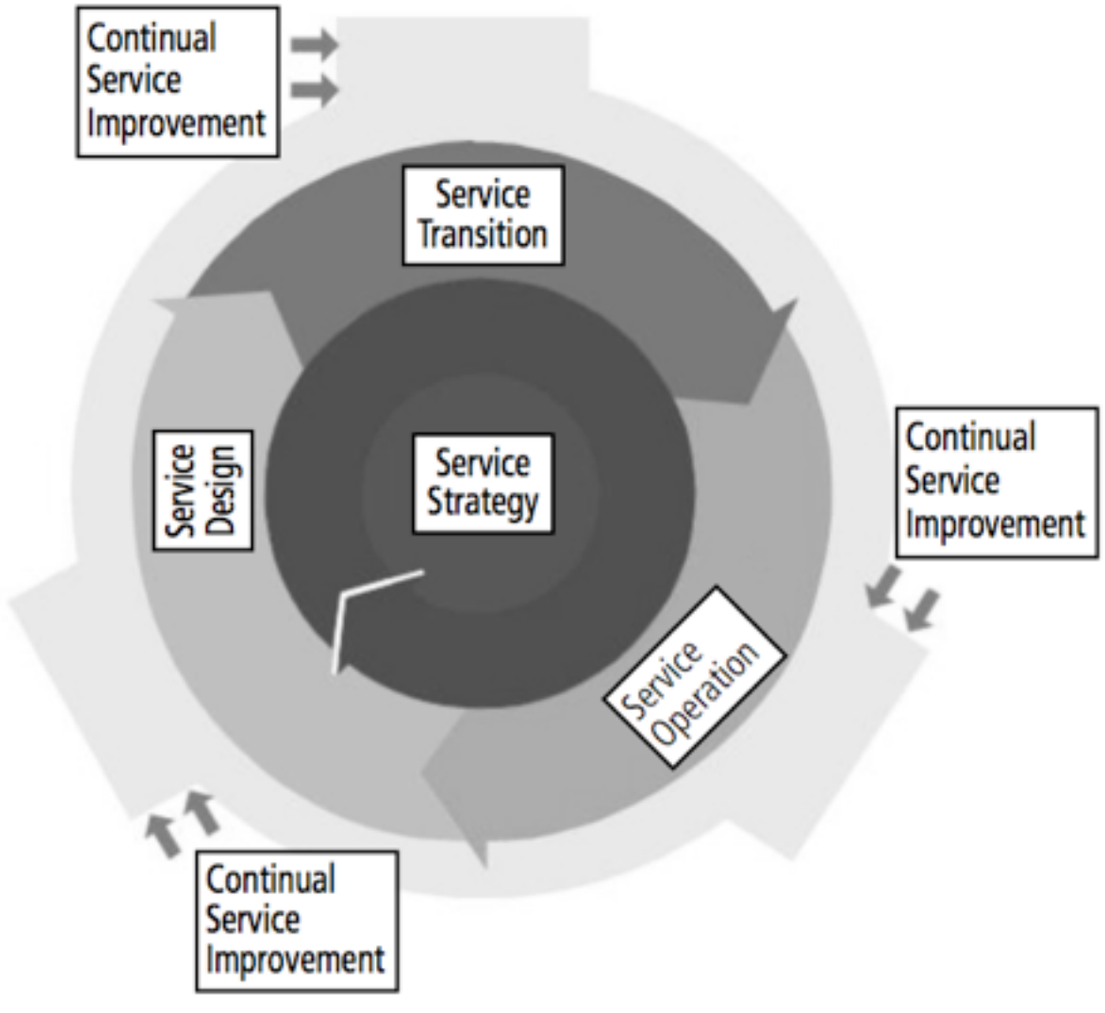
- *Executive leadership*: Empowers the other role holders with freedom and resources
  - Responsible to drive new ideas for improvement
- *Champions*: Drawn from upper management, responsible for Six Sigma deployment
  - Act as mentors for black belts
- *Master black belts*: Full-time in-house experts
  - Coach and trainer for the cross-department implementation of Six Sigma
- *Black belts*: Apply Six Sigma methodology full-time to specific projects
  - Rule of thumb: One black belt per 100 employees (1% rule)
- *Green belts*: Common employees helping the black belts along with their normal job
- Rising number of participating employees is expected to increase profitability

Six Sigma Trends

- *Design for Six Sigma (DFSS / DMADV)* approach
  - Optimize both customer needs and organizational objectives in engineering work
  - Aims at early design phase, not at improving existing process
- 5 stages in a cycle
  - **Define**: Develop *new product development (NPD*) strategy
  - **Measure**: Understand customer requirements
  - **Analyze**: Develop conceptual design after analyzing design options
  - **Design**: Develop product or process design
  - **Verify**: Develop prototype, evaluate effectivness of the design
- Design process or service „with the end in mind“

ITIL

- Information Technology Infrastructure Library (ITIL), latest version v3
  - Started as set of recommendations by the UK Government
  - Concepts and guides for **IT service management**
  - Supports to deliver business-oriented quality IT services
- Core publications: **Service Strategy**, **Service Design**, **Service Transition**, **Service Operation**, **Continual Service Improvement**
- Broad tool support from vendors
- High costs for certification and training
- Methodology sometimes over-respected at the expense of pragmatism



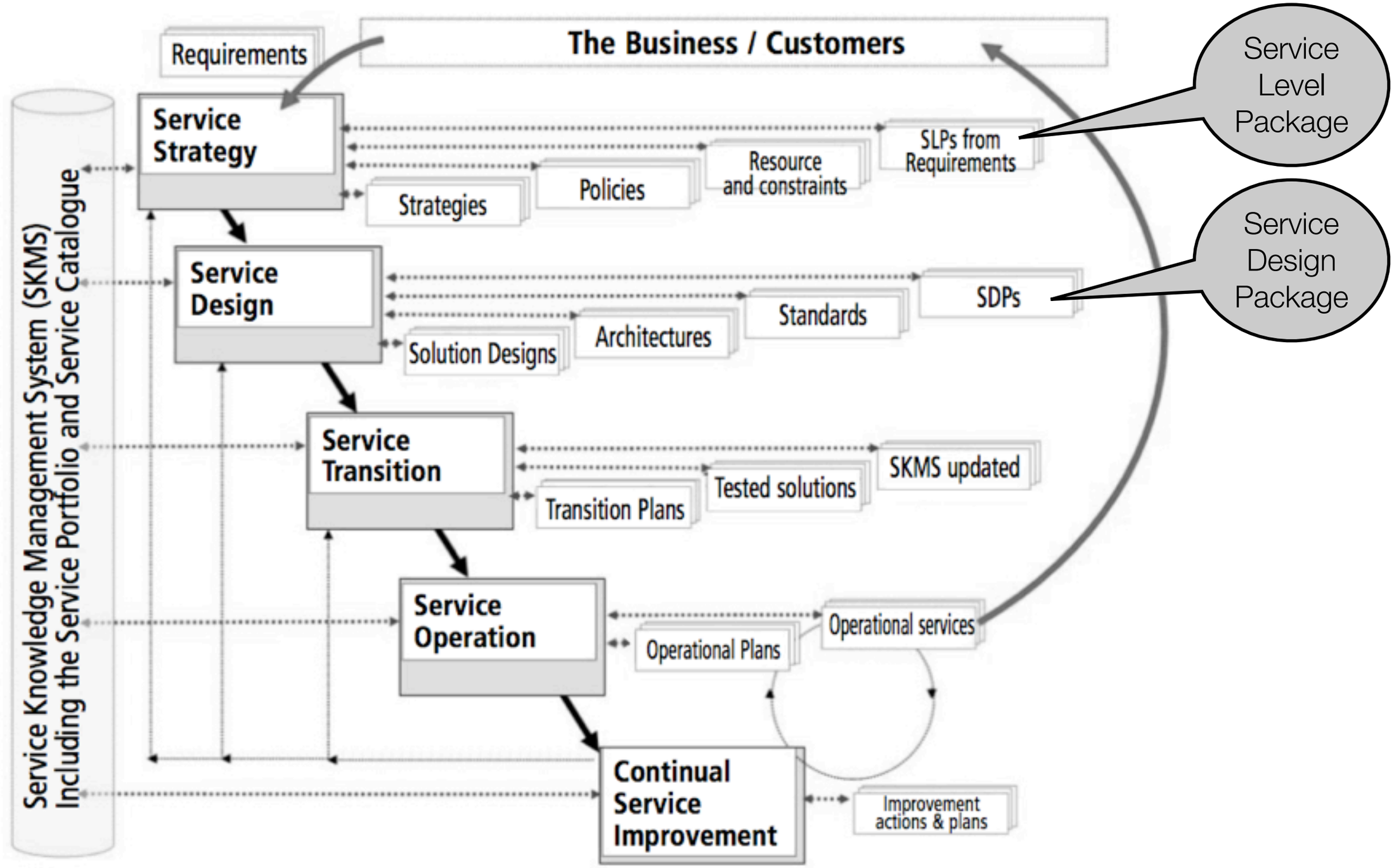
ITIL

- Replaces technology-oriented view on IT service management with an „end-to-end“ approach
- Get rid of „technology silos“ and „islands of excellence“
- Management system, expected to be ...
  - ... more focused on business needs and related to business processes
  - ... less dependent on specific technology, service thinking
  - ... more integrated with other management approaches and tools

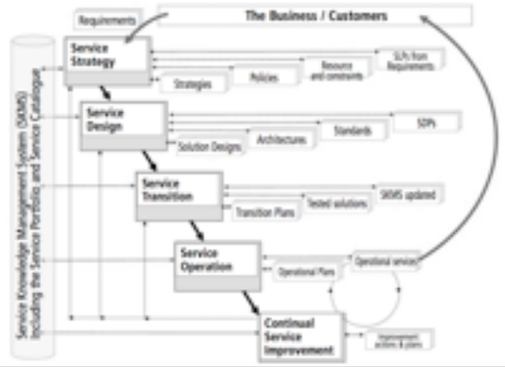
ITIL v3 - Service

- *„A service is a means of delivering value to customers by facilitating outcomes customers want to achieve without the ownership of specific costs and risks.“*
- Supports the customer / user in achieving his defined goals
- **Service value** for the customer = **utility** (what) and **warranty** (how)
- **Service level**, the service quantity and quality demands, should be measurable
- Current and future demands on service level are specified in a **service-level agreement**
- **Service assets** (resources, capabilities to control them) support **business assets**
- **Service providers**:
  - Type 1 (internally for one business unit)
  - Type 2 (for multiple business units in the same organization)
  - Type 3 (serving multiple external customers)

ITIL v3 - Service Lifecycle
Initiated from a change in business requirements



ITIL v3 - Service Strategy



- Service strategy has to rely on acknowledgment that customer buys satisfaction of needs, not a physical product
  - What customers ? What needs ? When and why do they occur ? Current and potential market places ? Competitors ? How to achieve differentiation ?
- The **four Ps of strategy**
  - **Perspective**: Distinct vision and direction (e.g. what services to provide)
  - **Position**: The basis on which the provider will compete (e.g. market check)
  - **Plan**: How the provider will achieve the vision
  - **Pattern**: Distinct fundamental patterns in decisions and actions of time

ITIL v3 - Service Strategy

- **Service Provisioning Models**
  - Selected by customers, used by providers
  - **Managed Service**: Business unit requiring a service fully funds the service provision for itself
  - **Shared Service**: Provisioning of multiple services through shared resources and infrastructure, targeting one ore more business units
  - **Utility**: Provisioning based on customer requirements (how much, how often, what time)
- **Organization Design and Development**
  - Development stages of the organization itself, interfaces to services, service analytics, strategies for service sources (internal, shared, outsourcing, ...)

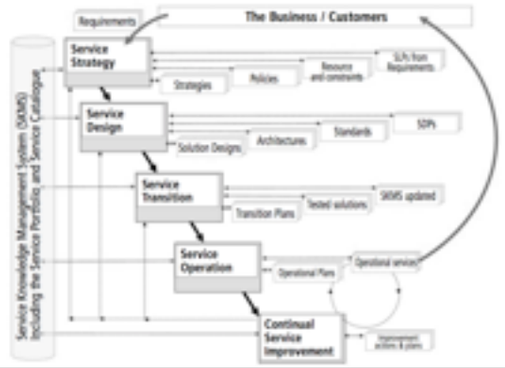
ITIL v3 - Service Strategy

- Service strategy process also contains side activities
  - **Financial management** for quantification of the money value of services - budgeting, accounting, charging
  - **Service portfolio management (SPM)** as continuous process for controlling investment across the service lifecycle
    - Planned services, live services, retired services
    - **Define, Analyze, Approve, Charter** the portfolio
  - **Demand management**
    - Example: Excess capacity generates costs without generating value
    - Based on utility and warranty terms for service value

ITIL v3 - Service Strategy

- Roles and responsibilities
  - **Business Relationship Manager (BRM)** - Close contact to customer
  - **Product Manager (PM)** - Development and management of service lifecycle
  - **Chief Sourcing Officer (CS)** - Leading sourcing strategy development

ITIL v3 - Service Design



- Design IT services to meet current and future agreed business outcomes
- Service design has to consider
  - Agreed business outcomes
  - Support for service lifecycle, risk management, security, resiliency
  - Definition of measurement methods and metrics, skill development
- The **four Ps of design**
  - **People:** Peoples, skill and competencies involved in the provisioning
  - **Products**: Technology and management systems used in service delivery
  - **Processes**: Processes, roles and activities involved in service provisioning
  - **Partners**: Vendors, manufacturers, and suppliers used to assist the provisioning

ITIL v3 - Service Design

- **Service Design Package (SDP)**: Defines all aspects and requirements of the service
  - Produced for each new IT service, major change, or IT service retirement
- **Service Catalogue Management (SCM)**: Maintaining the central source of service information for business areas, describing available IT services
- **Service Level Management (SLM):** Negotiates, agrees and documents IT service targets with the business (e.g. SLA)
- **Capacity Management**: Management of capacity and performance-related issues for services and resources, to match IT to business demands
  - **Capacity Management Information System (CMIS)**
- **Availability Management**: Reactive activities (monitoring, measuring, analysis, event management, incident management) vs. proactive activities (planning, design, recommendations)
  - **Availability Management Information System (AMIS)**

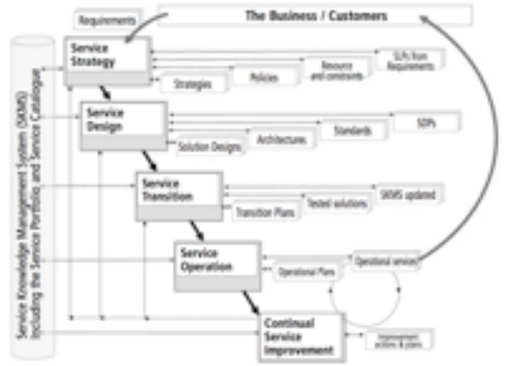
ITIL v3 - Service Design

- **IT Service Continuity Management (ITSCM)**
  - Maintain appropriate risk reduction measures and recovery options
- **Information Security Management (ISM)**
  - Considered within the corporate governance framework
  - Align IT security with business security, ensure information security
    - *Information availability* - should be usable when required
    - *Information confidentiality* - observed or disclosed to only those who have a right to know
    - *Information integrity* - Completeness, accuracy, no unauthorized modification
    - *Information authenticity and non-repudiation* - Business transactions and information exchange can be trusted

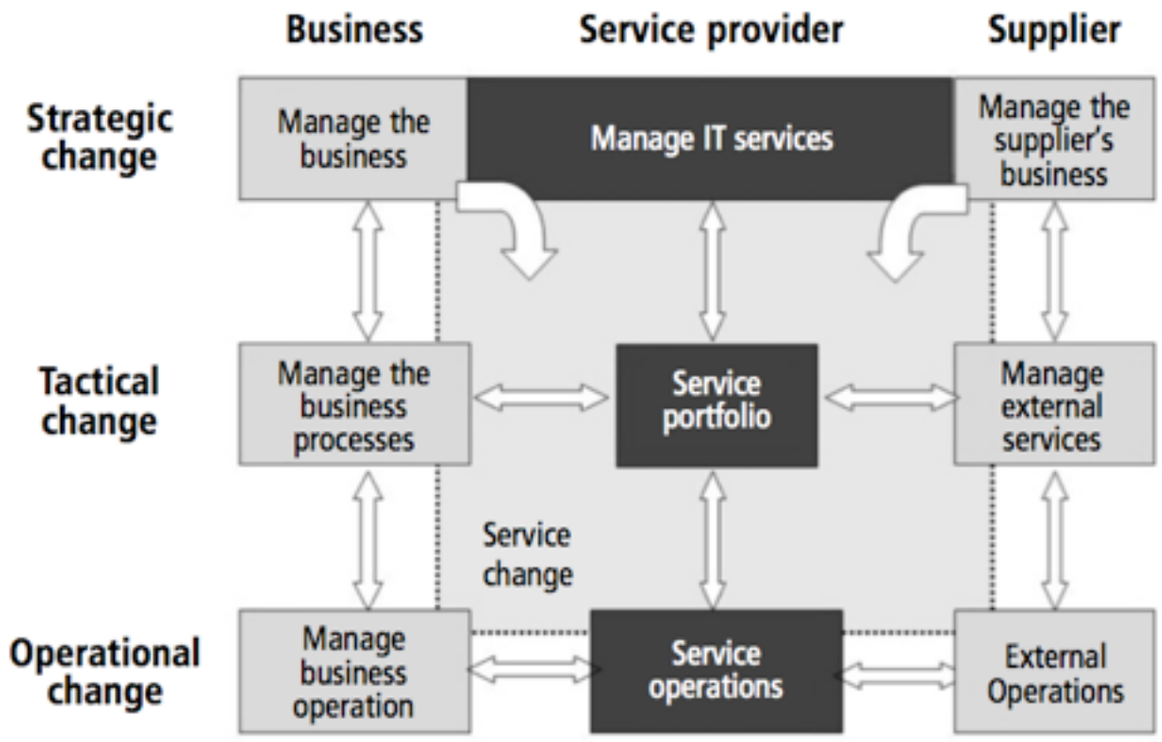
ITIL v3 - Service Design

- **Supplier Management**
  - Obtain value for money from suppliers, while meeting the business targets
  - **Supplier and Contract Database (SCD)**
- Roles and responsibilities
  - **Service Design Manager** - Coordination and deployment of quality solution designs for services and processes
  - **IT Designer / Architect** - Coordination and design of required technologies, architectures, strategies, designs, plans
  - **Service Catalogue Manager, Service Level Manager, Availability Manager, IT Service Continuity Manager, Capacity Manager, Security Manager, Supplier Manager ...**

ITIL v3 - Service Transition



- Deliver services into operational use
  - Receive SDP‘s from design stage, deliver into operational stage every element required for ongoing operation
  - Needs to ensure that service can operate in foreseeable extreme or abnormal circumstances



- **Change management**
  - Can apply to different scopes of change and release management
  - Ensures that service changes are recorded, evaluated, authorized, prioritized, planned, tested, implemented, documented, reviewed
  - Optimize business risk of changes

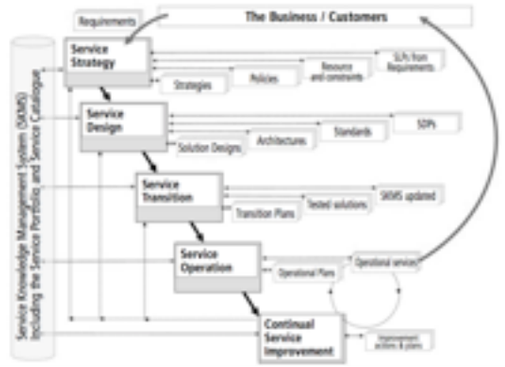
ITIL v3 - Service Transition

- **Service Asset and Configuration Management (SACM)**
  - Identify, control and account for service assets and configuration items (CI)
  - **Support by Configuration Management System (CMS)**
- **Knowledge Management**
  - *Data-Information-Knowledge-Wisdom* structure
  - Make information available to the right person at the right point in time
- **Transition Planning and Support**
  - Identify, manage, and control risks of failure across transition activities
  - Improves ability to handle high volume of change and releases

ITIL v3 - Service Transition

- **Release and Deployment Management**
  - Covers whole assembly and implementation of new/changed services for operational use
  - From release planning to early life support
  - Deliver changes at optimized speed, risk and cost
- **Service Validation and Testing, Evaluation**
  - Provide objective evidence that the new / changes service supports the business requirements, including agreed SLAs
- Roles and responsibilities
  - Not anticipated as separate group of people, ‚flow of experience and skills‘

ITIL v3 - Service Operation



- Deliver agreed levels of service to customers, manage applications and infrastructure that support the service delivery
- Only in this stage, services actually deliver business value
- Balancing of conflicting goals
  - Internal IT view vs. external business view
  - Stability vs. responsiveness
  - Quality of service vs. cost of service
  - Reactive vs. proactive activities
- *Operational health* of services summarizes relevant indicators relevant for execution of **vital business functions**

ITIL v3 - Service Operation

- **Event Management Process**
  - Handle change of state that has significance for the management of a configuration item or IT service
  - May indicate correct or incorrect functioning (tape change vs. hardware outage incident)
  - Depends on monitoring, but always generates / consumes notifications
  - Response to an event can be automated or may require manual intervention
- **Incident Management Process**
  - Handle unplanned service interruption, quality reduction, or configuration item error state not impacting the service
  - Restore normal service as quickly as possible
  - Functional vs. hierarchical escalation

ITIL v3 - Service Operation

- **Request Fulfillment Process**
  - Source and deliver services
  - Provide information about services and procedures for obtaining them
- **Access Management Process**
  - Manage confidentiality, availability and integrity of data and intellectual property
  - Manage service access rights and user identity
- **Problem Management Process**
  - Problem is a cause of one or more incidents
  - Responsible for investigation of incidents for identification of the problem
  - Understand causes, document workarounds and request changes to permanently resolve the problems

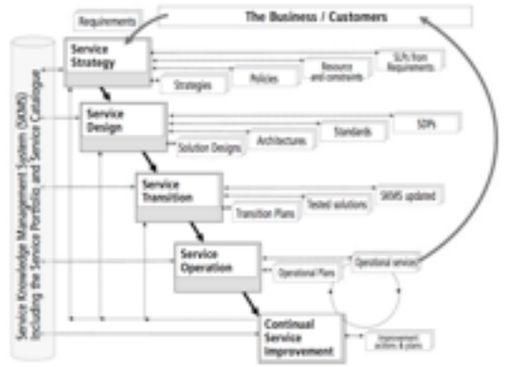
ITIL v3 - Service Operation

- **Service Desk Function**
  - Single central point of contact for all users of the IT
  - Logs and manages all incidents, service requests and access requests
  - First-line investigation and diagnosis, keeping users informed
  - Different organization approaches:
    - Local service desk (close to the user)
    - Centralized service desk (higher volume with fewer stuff)
    - Virtual service desk (staff in many locations, appear as one team)
    - Follow the sun (24h coverage, automated call passing)

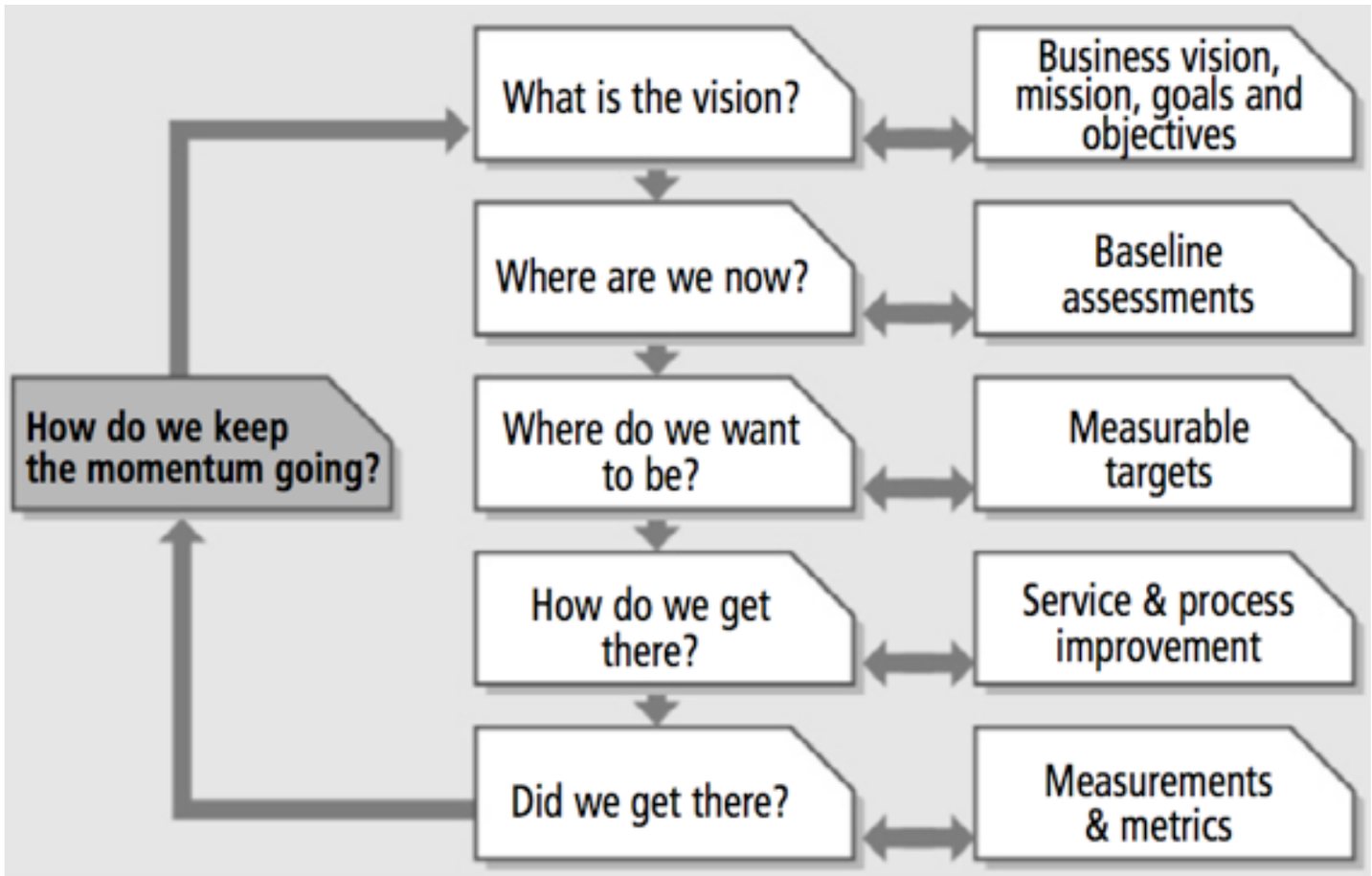
ITIL v3 - Service Operation

- **Technical Management Function**
  - People providing technical expertise and management of IT infrastructure
  - Mainly assistance function
- **Application Management Function**
  - Focus on software rather than infrastructure
  - Applications treated as one component of service
- **IT Operations Management Function**
  - Operations control - routine tasks, centralized monitoring from cockpit
  - Facilities management - data centres, computer rooms, recovery sites

ITIL v3 - Continual Service Improvement

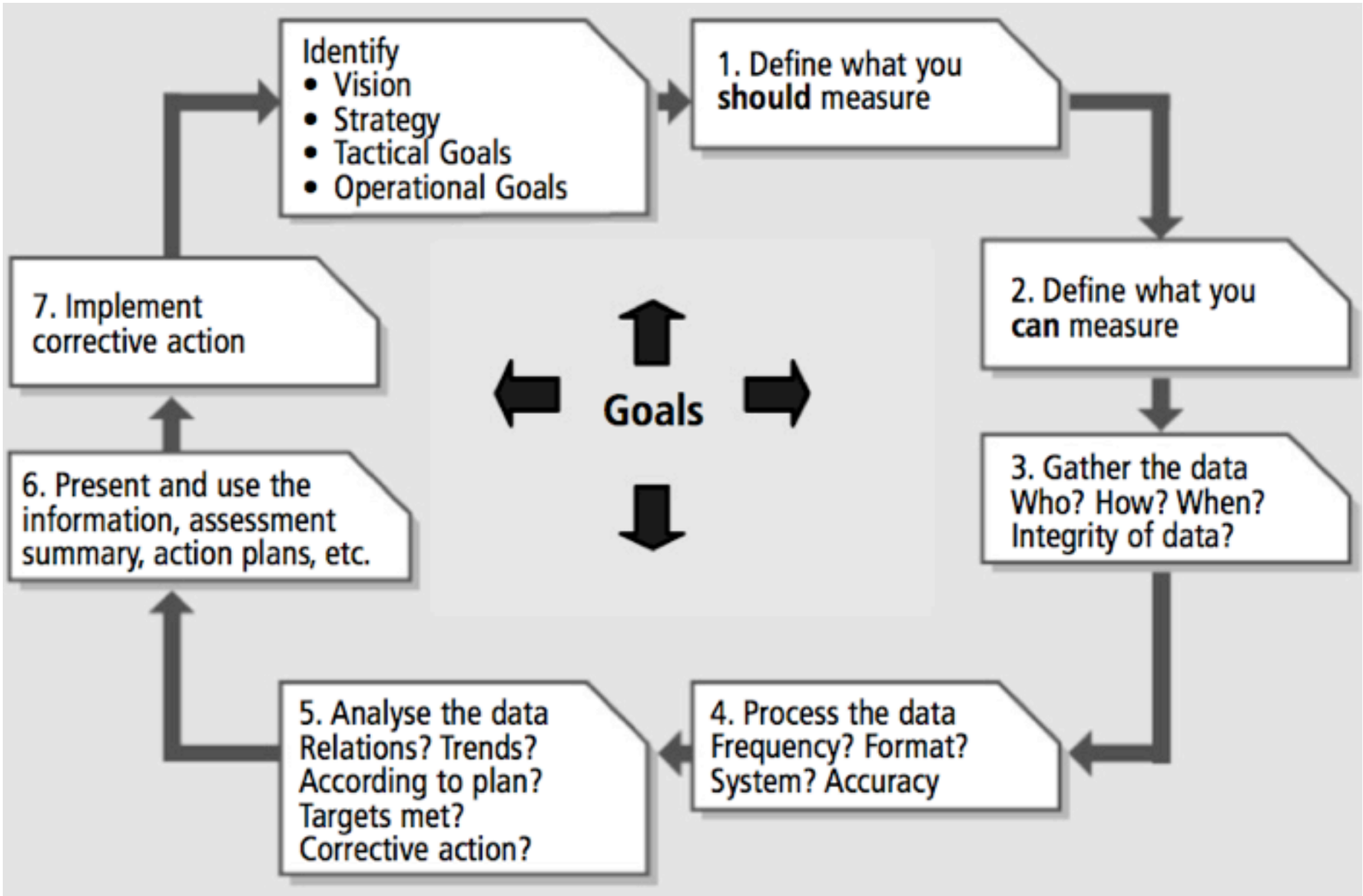


- Continual Service Improvement (CSI) maintains value for customers
  - Should be embedded in organization, instead of ad-hoc approaches after failure
- CSI model - Contrast current position with long-term goals and objectives



ITIL v3 - Continual Service Improvement

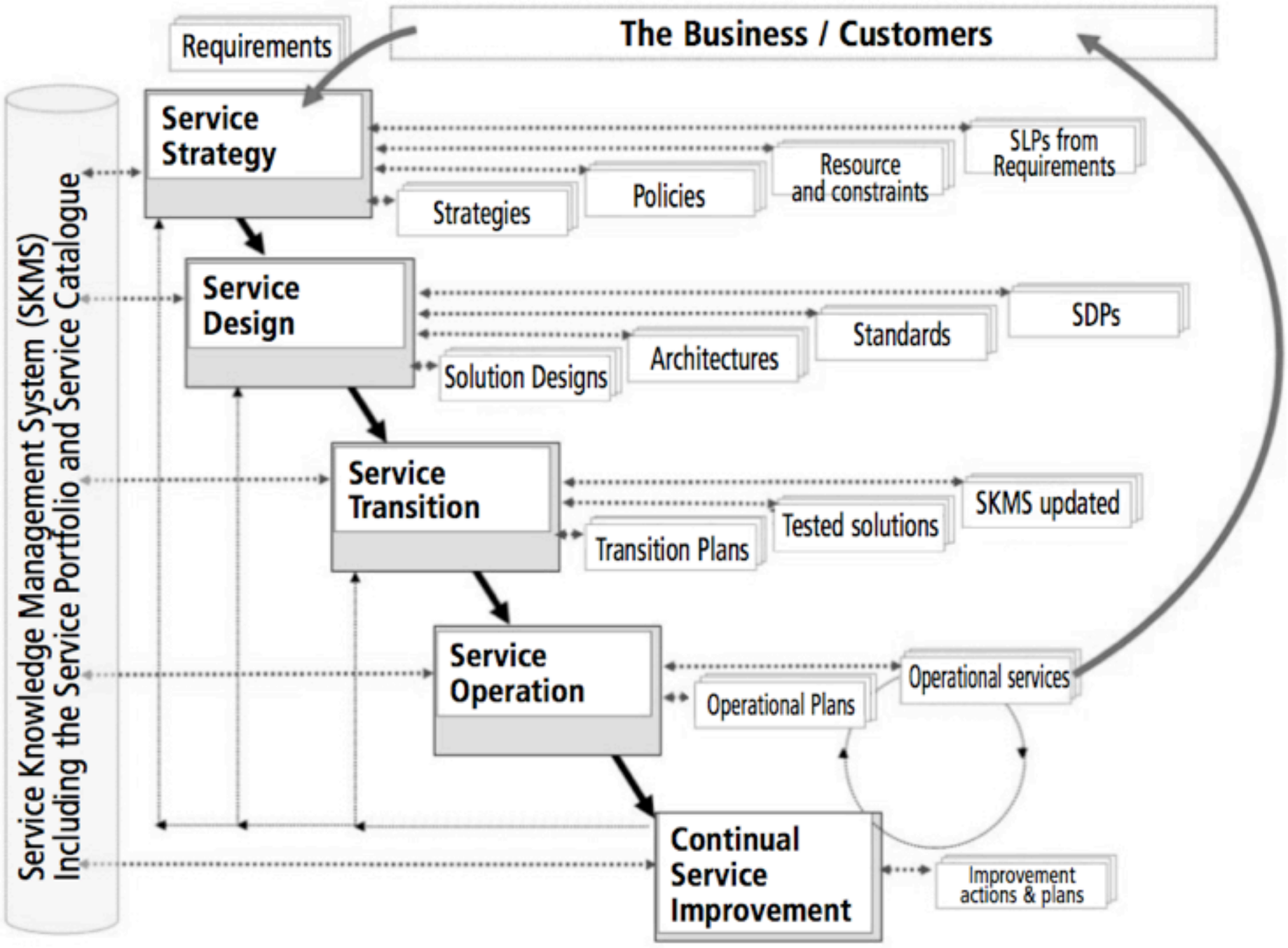
- 7-Step Improvement Process



ITIL v3 - Continual Service Improvement

- **Service Measurement**
  - Metric types:
    - **Technology metrics** - components, applications; performance, availability
    - **Process metrics** - critical success factors (CSFs), key performance indicators (KPIs)
    - **Service metrics** - results of service, computed from technology metrics
  - Validate earlier decisions, help in deciding for corrective actions
- **Service Reporting**
  - Summarize historical development of data collected
- Dedicated **CSI manager role** recommended, all other activities as part of the other lifecycle stages

ITILv3 - Service Lifecycle



ITILv3 - Service Lifecycle

Continual Service Improvement (CSI)
7-Step Improvement Process
Service Measurement
Service Reporting

Service Strategy (SS)
Strategy Generation
Financial Management
Service Portfolio Management
Demand Management

Service Operation (SO)
Event Management
Incident Management
Request Fulfilment
Problem Management
Access Management

Service Transition (ST)
Transition Planning and Support
Change Management
Service Asset & Configuration Mgmt
Release and Deployment Mgmt
Service Validation and Testing
Evaluation
Knowledge Management

Service Design (SD)
Service Catalogue Management
Service Level Management
Capacity Management
Availability Management
IT Service Continuity Management
Information Security Management
Supplier Management

CoBiT

- *Control Objectives for Information and related Technology (COBIT)*
- International model for defining control and audit goals for IT, started in 1996
  - Maintained by *Information Systems Audit and Control Association (ISACA)*
  - Association of auditors, reaction on business impact of IT
- Collection of best practices for auditing in IT, no standardization on its own
- Considers:
  - Technical standards (EDIFACT, ISO, ...)
  - Codes of conduct for business (EU, OECD, ...)
  - Qualification criteria for IT systems and processes (ISO 9000, common criteria, ...)
  - Professional standards from the domain
  - Documented industry practices

CoBiT

- Describes how to rate the maturity level of *IT processes*
  - Considers IT quality (efficiency, robustness), IT security (integrity, trustability), and IT fiduciary (compliance to financial and lawful rules)
- Processes are defined to rely on *IT resources*, structured into:
  - *Data* - structured and non-structures information elements
  - *Application systems* - collection of manual and programmed processes
  - *Technologies* - Hardware, operating systems, databases, networks, ...
  - *Facilities* - Resources for housing and operation of IT systems
  - *People* - Human part in IT system operation
  - All these factors must be planned, developed, implemented, operated and monitored in a coordinated fashion - meta framework for compliance checks

CoBiT

- Metrics: *Key Goal Indicators (KGI), Critical Success Factors (CSF), Key Performance Indicators (KPI)*
- Example: Change Management Prozess
  - KGI‘s: Number of successful changes, reduction of service interruptions due to changes (goals of the process itself)
  - CSF‘s: Useful software and hardware inventory (rating of applied improvements)
  - KPI‘s: Number of changes leading to failures, number of delayed changes (rating of applied improvements)
- Using COBIT in an organization
  - Auditing of IT processes according to COBIT
  - Using COBIT as internal evaluation tool
- KGI ranks resulting output, KPI ranks performance of producing these outputs

CoBiT

- ITIL covers one part of COBIT scope (IT service management processes)
- Maturity of IT processes can be ranked with COBIT
  - Becomes relevant for BASEL II or Sarbanes-Oxley rating of organizations
- Top-down approach in CoBiT analysis
  - Derive IT architecture (resources) from IT goals (processes) from business goals (requirements)