Our

Quality

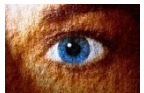
Management

System

is not used!

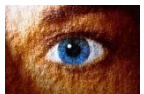
Paper summary

- It is very difficult to have an efficient and effective Quality Management System (QMS). Creating and maintaining a QMS is for organisations normally not core business.
- And what if you are a company that wants to start developing and manufacturing for instance medical devices on top of the existing portfolio of non medical systems? How can the QMS help you with serving both product lines?
- This presentation gives an overview on how to design, deploy and maintain an efficient and effective QMS:
  1. Requirements for the QMS have to be gathered also taking into account the characteristics of the organisation (size, skill level of people, product portfolio, markets, …)
  2. Next a suitable QMS design is to be chosen including how tailor towards projects can be achieved
  3. Implementation of the design has to be done resulting in the QMS items like procedures and work instructions.
  4. Verification of the implementation by means of document review and validation by piloting in a representative project.
  5. Production phase of the QMS introduces the most difficult part of the product creation: deployment. How are the employees trained in such a way that everyone in the organisation is skilled enough to work according to the QMS.
  6. While working with the QMS, the end users (employees) will submit change requests. How must an organisation act upon these requests? Deploy, deploy, deploy
- For all steps mentioned above, hints and tips are given coming from real life examples. Not only what to do but also what NOT to do.
- Examples shown will be based on the international standard ISO 9001. On top of that, requirements according to the medical device standard ISO 13485 are implemented.



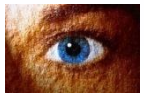
The latest version of this presentation

- Is to be found on my website www.processvision.nl



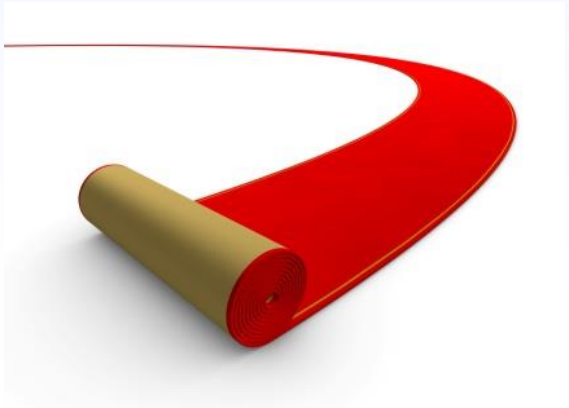
Willem van den Biggelaar

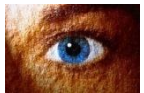
- Quality and Medical Regulatory Consultant
- Certified DEKRA auditor for ISO 9001 / ISO 13485
- Setup several ISO 9001 / 13485 certified Quality Management Systems (QMS)
- Previous jobs
  - Quality Assurance Officer (5 years)
  - System Tester (1 year)
  - Embedded software engineer (7 years)



Content

- The 6 steps of developing and maintaining a QMS
  1. Define Requirements
  2. Design
  3. Implement & Verify
  4. Validate (piloting)
  5. Deploy (initial)
  6. Under change control
- Treat it like a project
- FAQ





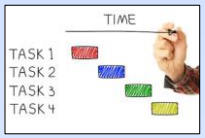
Like a V-cycle

Stakeholders wishes

Define Requirements

Quality Manual
- Quality Policy
- Management Commitment
- Organisation
- Scope

Project Plan



Change Requests

Validate (pilot run)

Deployment / Change Control

Design QMS

Quality Manual
- Processes + owners + interfaces

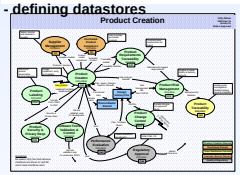
QMS Index
- procedures per process
- process owner

Procedure A process A
--- Template A
--- Work instruction A
--- Guideline A
Procedure B process B
-- Template B
-- Template C
Procedure C process A
Procedure D process C
-- Guideline B

Verify

Review Reports

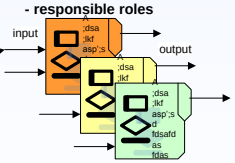
Procedure interfaces
- connecting flow charts input/outputs
- defining datastores



Annexes

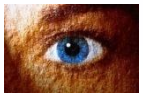
Templates | Guidelines | Work instructions

Procedures
- scope
- input / output / datastores
- activity descriptions
- responsible roles



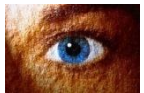
Training Material

Implement



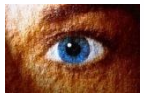
What is a Quality Management System?

- Describes a set of procedures an organization needs to follow in order to meet its business objectives.
- Small organizations:
  - mostly no official system
  - just ‘our way of doing things’
  - not written down, in the head of the staff.
- Organization gets larger
  - Written instructions about how things are done needed
  - Assure nothing is left out
  - Assure everyone knows who needs to do what, when, how.



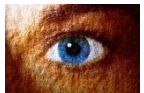
Why do you need a QMS (stakeholder wish)

- Customer demand
  - E.g. for medical devices, it is essential for a manufacturer that its suppliers are ISO 13485 certified
- Organisation communication gets complexer
  - More people in one discipline, larger projects
  - Multidisciplinair, e.g.
    - Designers <-> Testers
    - Designers <-> Engineers <-> Production
    - Sales <-> Production <-> Logistics
- Improve process / product quality
- Standard way of working in order to focus on product



1. Define requirements for a QMS

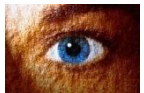
Requirements for the QMS have to be gathered also taking into account the characteristics of the organisation (size, skill level of people, product portfolio, markets, …)



1. Define: Product Market Combination (PMC)

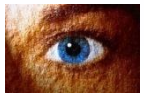
- Product examples
  - Healthcare
  - Automotive
  - Machinery
  - Consumer
  - Aerospace
  - Food
  - Pharmacy
  - Measurement instruments
  - Railway
  - Radio and telecom
- Market examples
  - Europe
  - USA
  - Canada
  - China
  - Australia
  - Russia
  - Asian pacific
  - Africa
  - South America

Each combination has its own legislation to which you must comply



1. Define: Comply with International standards

- Examples of QMS standards
  - ISO 9001 – General standard
  - ISO 13485 – Medical
  - AS 9100 – Aerospace
  - TS 16949 – Automotive
  - TL 9000 – ICT
  - ISO 27001 – IT Security
  - ISO 14001 – Environmental
  - OHSAS 18001 – Occupational health and safety
- Which to choose depends on Product Market Combination



1. Define Company Quality Policy

Pollock

QUALITY POLICY

We at Pollock Paper and Packaging are committed to a continuous process of providing innovative and quality products, services and solutions, on time and as ordered, in a professional and enthusiastic manner. This never-ending commitment will achieve our objective of total customer satisfaction, the foundation of what we call

"THE POLLOCK DIFFERENCE."

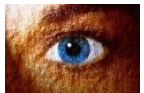
QUALITY ASSURANCE POLICY

QUALITY COMMITMENT

INECO IS COMMITTED TO TOTAL CUSTOMER SATISFACTION BY DELIVERING CONSISTENT QUALITY IN EACH OF IT'S PRODUCTS AND SERVICES

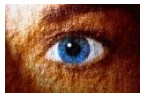
12th June, 2002

S. Bhargava
General Manager (M.E.)



1. Management must show commitment

- Let top management have quality on agenda of (yearly) personnel meeting
  - (Changes to) Quality policy
  - Metric
- Let department managers have quality on (monthly) department meeting agenda
  - (Changes to) Quality policy
  - Results internal and external audits
- Management must reserve resources for quality
  - QMS maintenance
  - Quality Audits

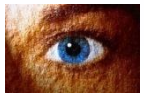


1. Define: Fit to your organisation

- Size of organisation
- Number and type of departments
- Skill and competence level of the employees
- Examples

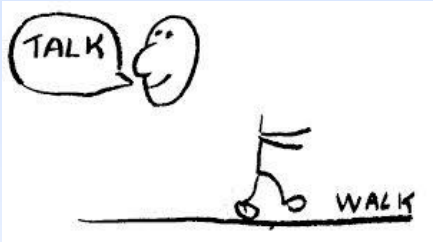
Id	Product	#Staff	Education	Key departments
A	Respiratory medical equipment	10	Academic, Large experience in product	Marketing & sales Development & Engineering Outsourced Production
B	Printed Circuit Assembly (PCA)	50	Secondary school	Purchase / Sales Mass production
C	Different types of mechatronics products	80	Academic	Development & Engineering Prototype production

- A: high level procedures leaning on medical standards
- B: detailed work instructions for production
- C: high level procedures based on different standards

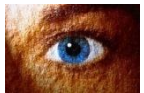


1. What I have seen gone wrong

- No or little management commitment, e.g.
  - “High quality products” is leading in the quality policy but management decides to deliver less quality if that saves money
  - Not enough resources available for QMS updates / internal audits

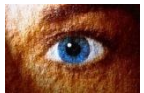


- People have no idea that what the company quality policy is
- Too heavy QMS for an organisation

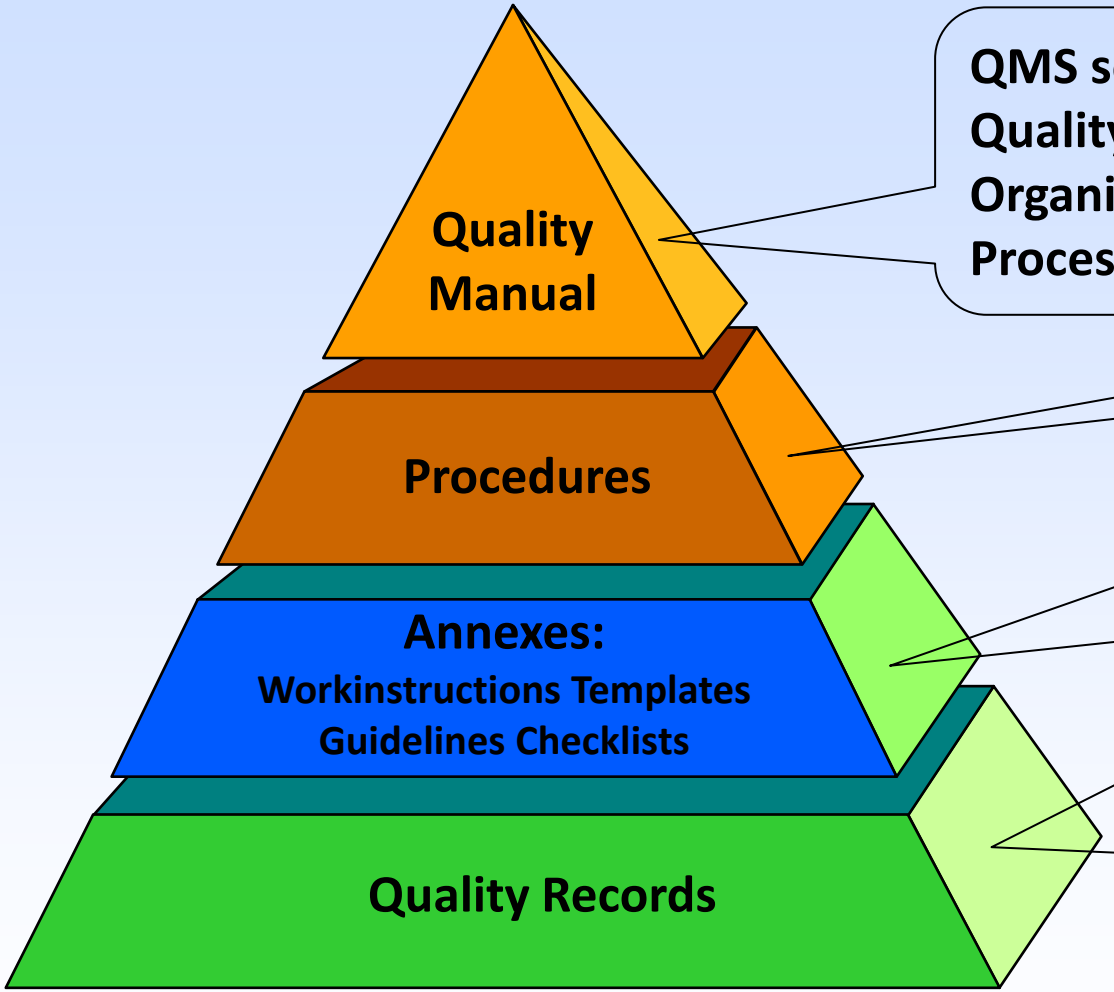


2. QMS architecture & design

Next a suitable QMS design is to be chosen including how tailor towards projects can be achieved



2. Design: Well known QMS architecture

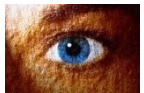


QMS scope and requirements
Quality Policy
Organisational structure
Processes including owners

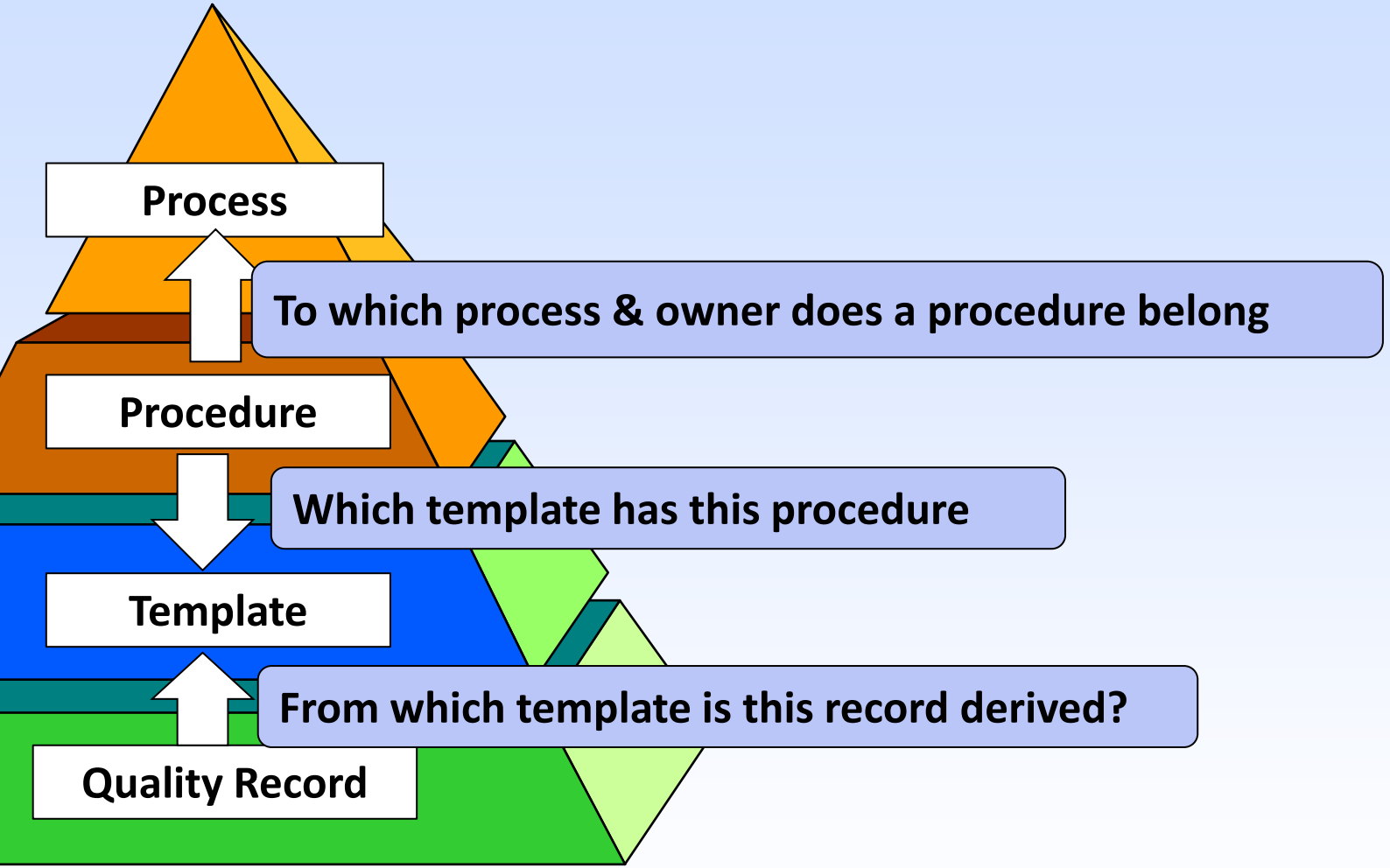
Mandatory to follow

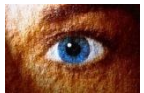
Annexes:
Extra help to be more efficient and effective

All documents and records that proof that the QMS is executed conform its requirements

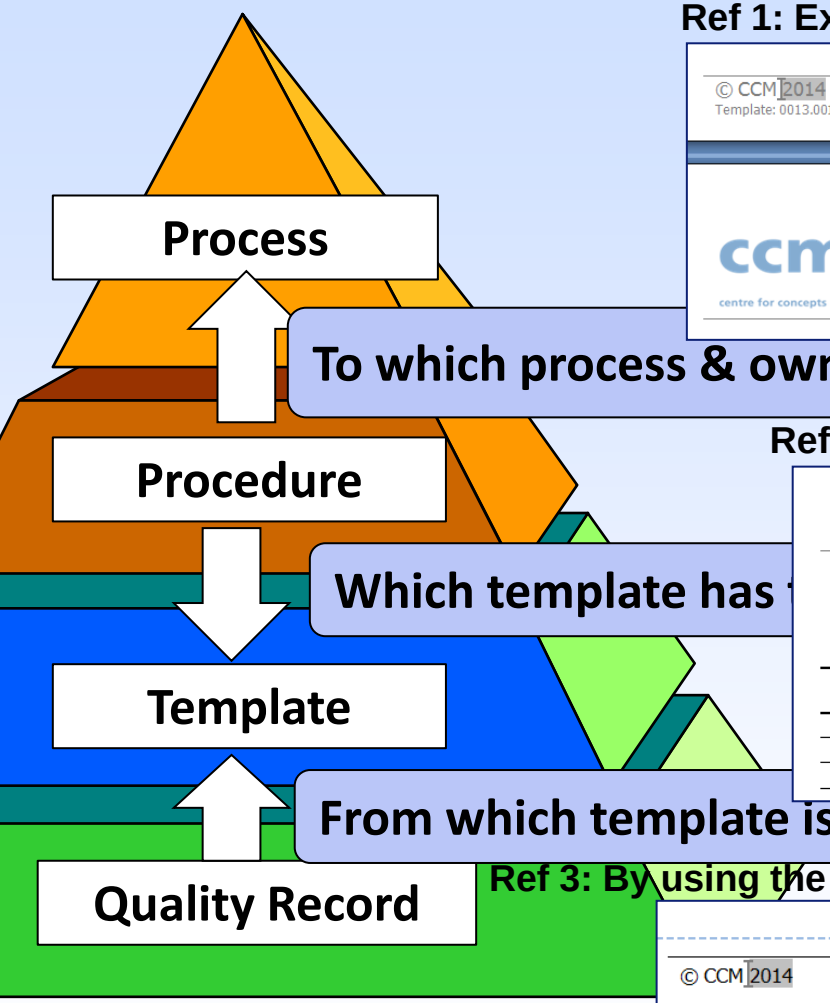


2. Design: Traceability between layers





2. Design: Traceability between layers

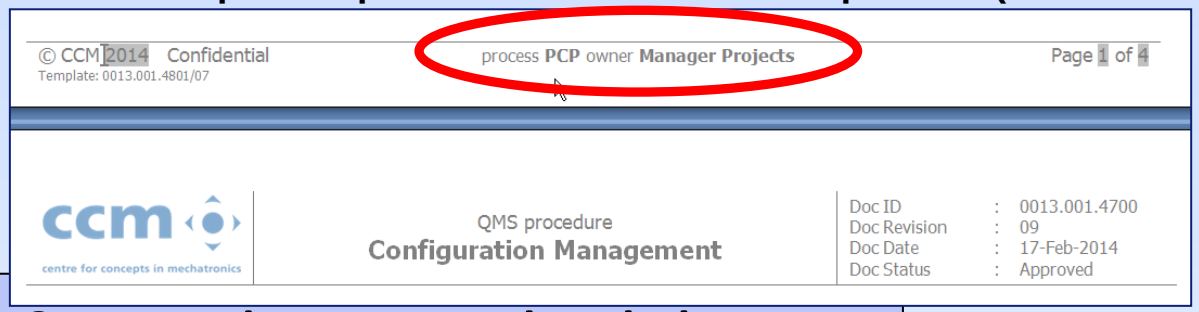


To which process & owner does a procedure belong

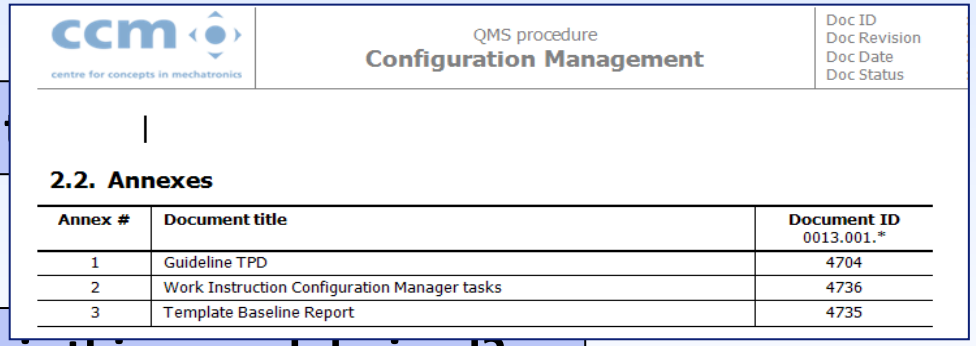
Which template has

From which template is this record derived?

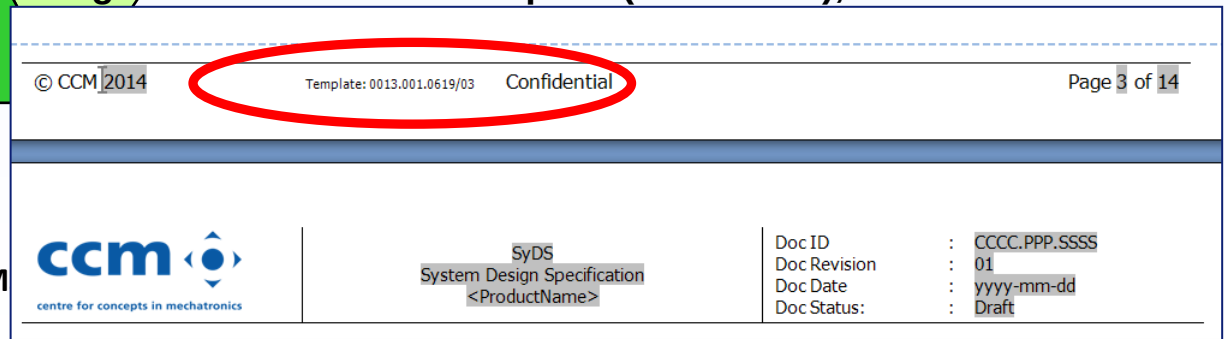
Ref 1: Example of a procedure that refers to its process (and owner)

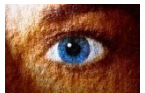


Ref 2: Example of a procedure that refers to its templates,



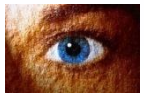
Ref 3: By using the latest released template (id+version), reference 3 is assured



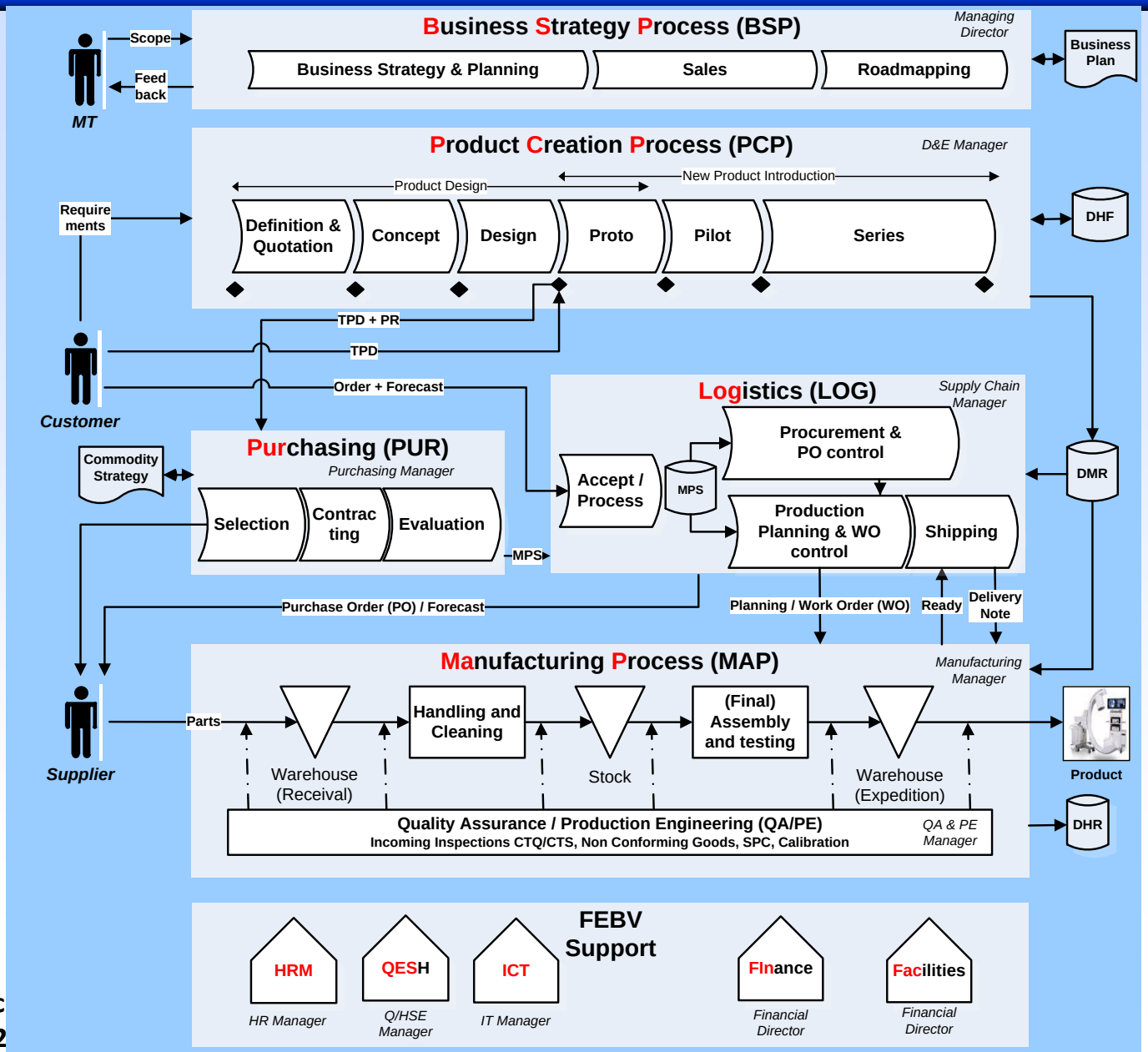


2. Design: the processes

- Divide in main processes and supporting processes
- Assign process owners:
  - Design/Own the process
  - Deploy/Advocate the process
  - Measure the process
  - Maintain/Improve the process
- Do NOT make the QA manager owner of everything
- Processes ≠ Departments!
  - A product creation process is a divided responsibility not only owned by the development department
- Design the interfaces between the processes

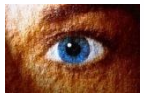


2. Design: process overview in one picture

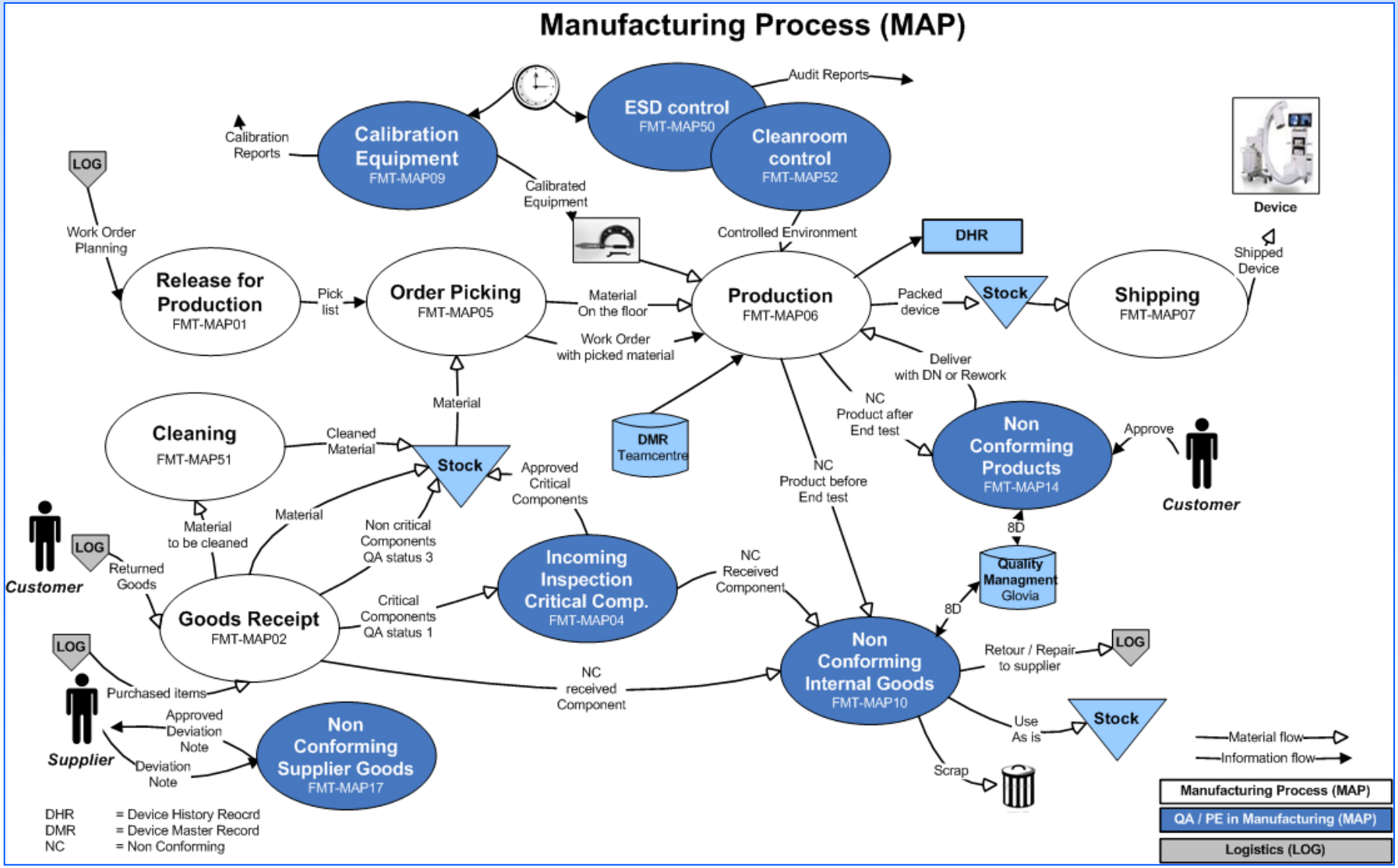


2. Design the procedure interfaces

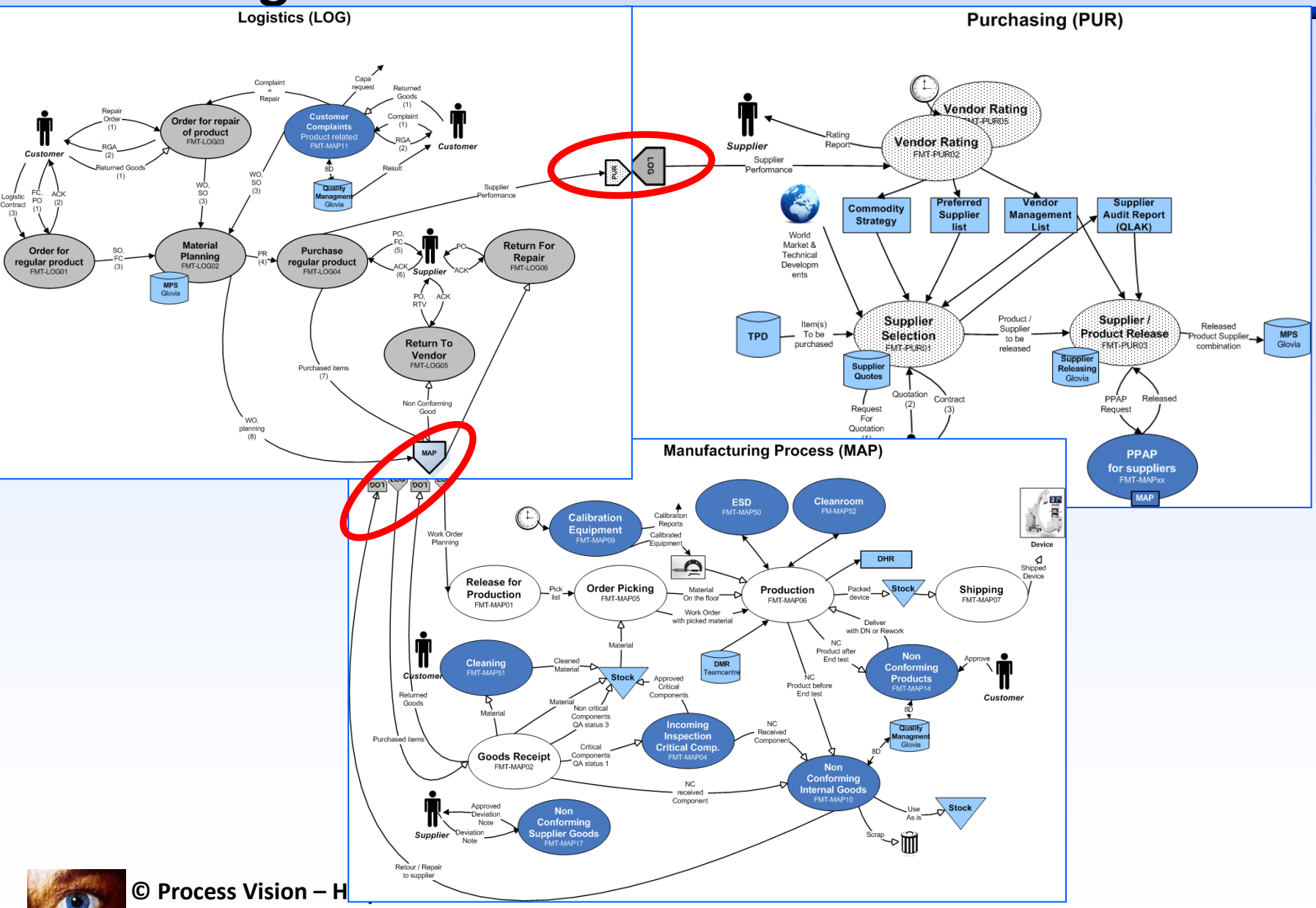
- Allocate to each process the neccessary procedures
- Guard the interfaces between the processes
- Again, put it all in a picture
- Tip: you can use these pictures for training purposes



2. Design: Procedure interface per process



2: Design: Guard the interfaces



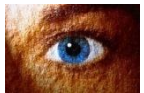
2. What I have seen gone wrong

- No process owners, QA does all the work
- Interfaces between procedures not aligned

- No overview of processes or procedures just a bunch of “how-to’s”.
- No strict division between QMS items (Q manual, procedures and annexes) and Quality Records

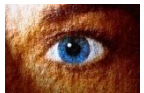
3. Implementation & Verification

Implementation of the design has to be done resulting in the QMS items like procedures and work instructions.
Verification of the implementation by means of document review



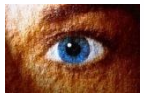
3. Write the procedures

- Don’t let key users be authors
  - Most of the time they don’t have the time for it
  - Most of the time they are not keen on doing it
  - Most of the time they are not capable of doing it
- Let stakeholders review (walkthrough) them:
  - Process owner (if not author)
  - Process owners for which their process interface
  - Key users – can I use it? Is it complete?
  - Quality Manager – is it compliant with regulatory



3. Tailoring (deviation of) the QMS

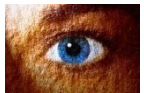
- Always allow projects to tailor towards own needs
- Let tailoring be documented in the project plan
- Let tailoring be approved by the QA manager
  - Tailoring must be compliant with regulations
- If much tailoring occurs, its a signal that perhaps the QMS is not suitable to the needs of the organisation



3. Write the templates and forms

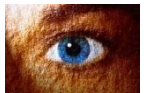
- Basis for the quality records so be precise
- First make mother template(s)/form(s)
  - Layout
  - Default content
- Next, make templates/forms for all possible document types
- Input
  - Mother template/form
  - Best practices
  - Procedures
- Again let stakeholders review (walkthrough) them

Mother template



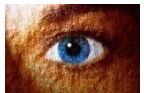
3. Write the other annexes

- If needed, guidelines
- If needed, checklists
- If needed, work instructions
- In native language if users are native speakers only
- Again let stakeholders review (walkthrough) them
- Checklists can evolve during history of the organisation, if same things go wrong over and over again, put them in checklists



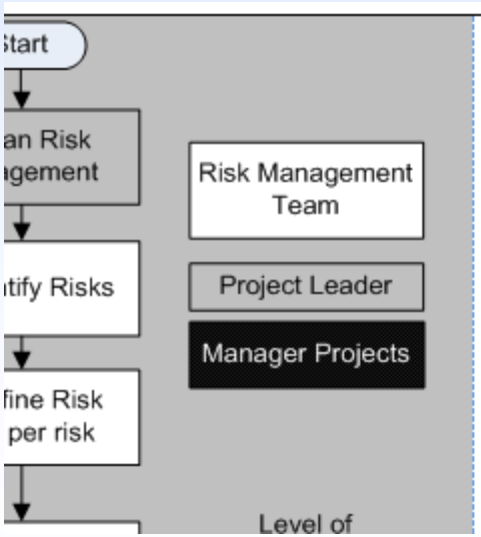
3. Write training material

- While writing you already have to replace yourself in the minds of the users of the QMS
- And maybe find out
  - A procedure has become much too complex
  - A template is not usable
  - ……



3. Deal with different regulations

- Suppose your organisation makes products for both medical and non medical industry
- Either write different procedures for each industry
- Or address the difference within the procedures & templates. E.g.



1. **Plan Risk Management (Project Leader)**

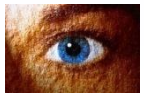
At start of project activities as described below are planned *and docum*
RMF.

2. **Identify Risks (Risk Management Team)**

For each risk type, identify and document the risks *(safety and securit*
should be traceable in the specification documents):

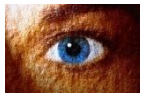
- Project Risks: [Annex 1], inputs are CRS, SRS, resources, time, b
- Product Risks: [Annex 2], inputs are CRS, SRS, intended use, mis
- Product risk regarding Safety for Operator and Patient, Machine s handled in the FMEA during the Product Creation Process.

Annex C of [Ref 7] contains useful safety questions to be asked.



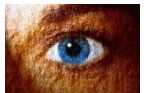
3. What I have seen gone wrong

- Local department templates and work instructions not under QMS control
- No mother templates so no escape if a template is not available leading to non compliant project documentation
- Procedures long (lot of text, no pictures) and tedious



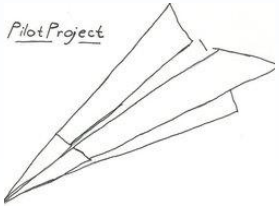
4. Validate the QMS

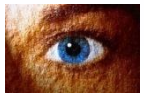
validation by piloting in a representative project.



4. Validate the QMS: choose pilot projects

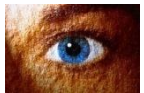
- Try to validate QMS on all its possible intended uses
- Choose (if possible)
  - Different type of projects. E.g:
    - 1 Medical project
    - 1 Machinery project
  - Different size of projects. E.g.:
    - 1 very small projects (1 – 10 FTE)
    - 1 very large project (50+ FTE)
    - 1 medium project (10 – 50 FTE)
- Choose projects that are just started
  - Each project leader will have a “reason” not to be chosen
  - Projects with high time pressure are reluctant to be chosen





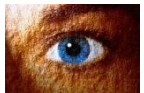
4. Validate the QMS: process the results

- Let projects log all issues they come across
- Implement the issues in the QMS
- Review changes with stakeholders



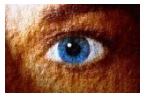
4. What I have seen gone wrong

- No piloting at all, simply “dump” the new process into live projects
- Only piloting with a small (easy) non typical project

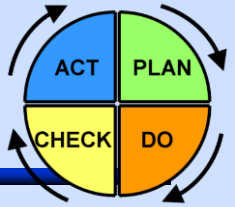


5. Initial deployment

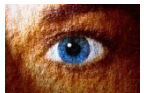
Production phase of the QMS introduces the most difficult part of the product creation: deployment. How are the employees trained in such a way that everyone in the organisation is skilled enough to work according to the QMS.



5. Deployment - training



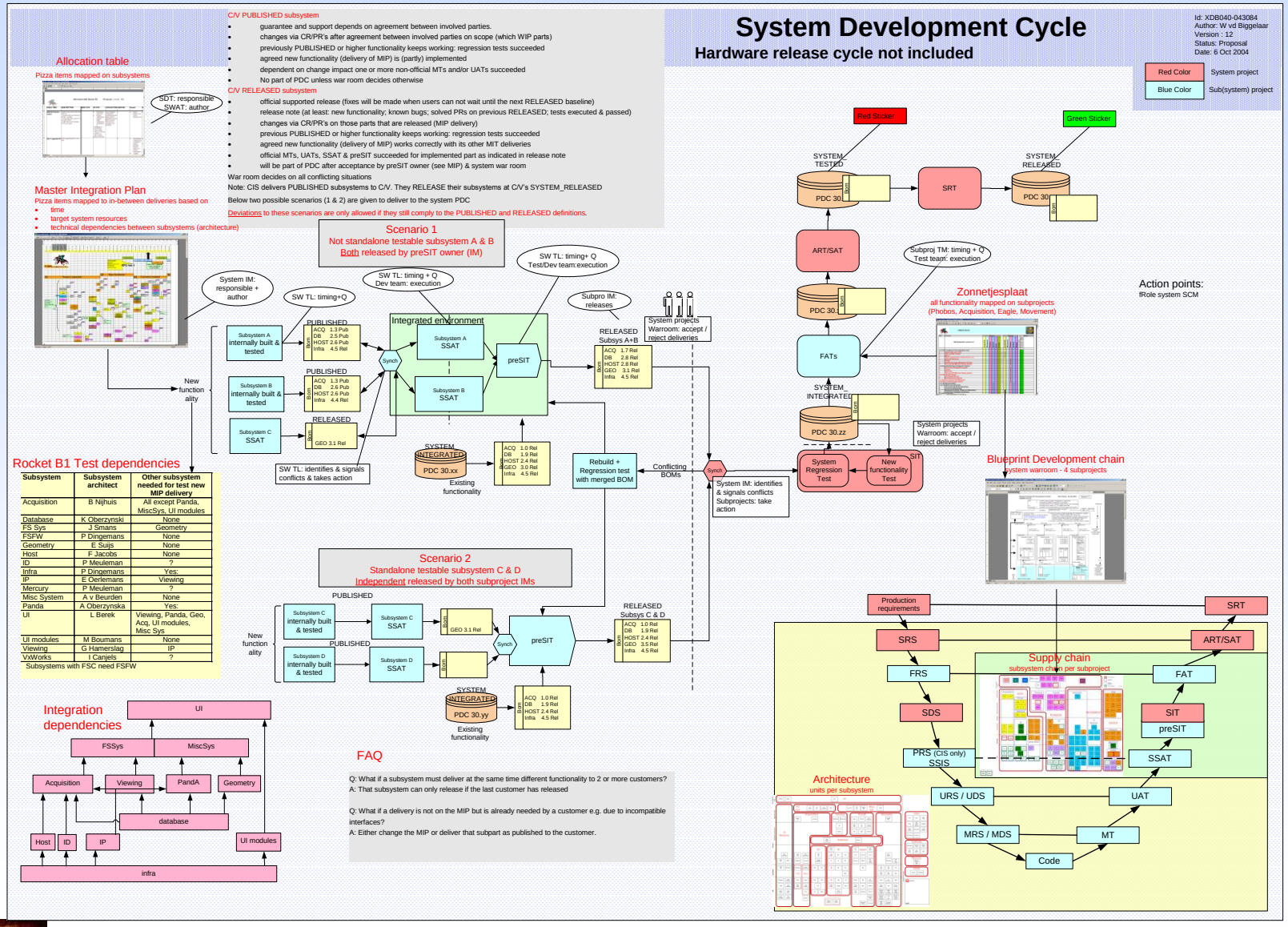
- Create training plan: who must be trained in what and when (input is QMS index)
- Initial training of people
- Re-train people pro-active (regular basis) or re-active (if needed)
- Don’t forget training program for new employees
- Let QA officer / Process Owner check by
  - Sanity checks (coffee corner, project meetings)
  - Official audits



5. Deployment – explain via wall posters

- Use posters on walls, coffee corners, team rooms, etcetera
- Also put in FAQ's, contact persons
- Explain them to the team
- If changes are due, hang new posters
- And re-explain
- The team can never say again "we didn't know"

An example

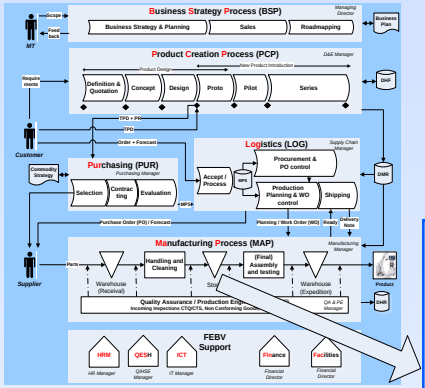

System Development Cycle
Hardware release cycle not included
Id: XDB040-043084
Author: W vd Biggelaar
Version : 12
Status: Proposal
Date: 6 Oct 2004
Red Color
System project
Blue Color
Sub(system) project
C/V PUBLISHED subsystem
guarantee and support depends on agreement between involved parties.
changes via CR/PR's after agreement between involved parties on scope (which WIP parts)
previously PUBLISHED or higher functionality keeps working: regression tests succeeded
agreed new functionality (delivery of MIP) is (partly) implemented
dependent on change impact one or more non-official MTs and/or UATs succeeded
No part of PDC unless war room decides otherwise
C/V RELEASED subsystem
official supported release (fixes will be made when users can not wait until the next RELEASED baseline)
release note (at least: new functionality; known bugs; solved PRs on previous RELEASED; tests executed & passed)
changes via CR/PR's on those parts that are released (MIP delivery)
previous PUBLISHED or higher functionality keeps working: regression tests succeeded
agreed new functionality (delivery of MIP) works correctly with its other MIT deliveries
official MTs, UATs, SSAT & preSIT succeeded for implemented part as indicated in release note
will be part of PDC after acceptance by preSIT owner (see MIP) & system war room
War room decides on all conflicting situations
Note: CIS delivers PUBLISHED subsystems to C/V. They RELEASE their subsystems at C/V's SYSTEM_RELEASED
Below two possible scenarios (1 & 2) are given to deliver to the system PDC
Deviations to these scenarios are only allowed if they still comply to the PUBLISHED and RELEASED definitions.
Allocation table
Pizza items mapped on subsystems
SDT: responsible
SWAT: author
Master Integration Plan
Pizza items mapped to in-between deliveries based on
time
target system resources
technical dependencies between subsystems (architecture)
Scenario 1
Not standalone testable subsystem A & B
Both released by preSIT owner (IM)
System IM: responsible + author
SW TL: timing+Q
SW TL: timing + Q Dev team: execution
SW TL: timing+ Q Test/Dev team:execution
Subpro IM: releases
System projects Warroom: accept / reject deliveries
New function ality
Subsystem A internally built & tested
Subsystem B internally built & tested
Subsystem C SSAT
PUBLISHED
RELEASED
Integrated environment
Subsystem A SSAT
Subsystem B SSAT
preSIT
Synchr
RELEASED Subsys A+B
SYSTEM INTEGRATED
PDC 30.xx
Existing functionality
SW TL: identifies & signals conflicts & takes action
Rebuild + Regression test with merged BOM
Conflicting BOMs
System IM: identifies & signals conflicts
Subprojects: take action
Rocket B1 Test dependencies
Subsystem | Subsystem architect | Other subsystem needed for test new MIP delivery
Acquisition | B Nijhuis | All except Panda, MiscSys, UI modules
Database | K Oberzynski | None
FS Sys | J Smans | Geometry
FSFW | P Dingemans | None
Geometry | E Suijs | None
Host | F Jacobs | None
ID | P Meuleman | ?
Infra | P Dingemans | Yes:
IP | E Oerlemans | Viewing
Mercury | P Meuleman | ?
Misc System | A v Beurden | None
Panda | A Oberzynska | Yes:
UI | L Berek | Viewing, Panda, Geo, Acq, UI modules, Misc Sys
UI modules | M Boumans | None
Viewing | G Hamerslag | IP
VxWorks | I Canjels | ?
Subsystems with FSC need FSFW
Scenario 2
Standalone testable subsystem C & D
Independent released by both subproject IMs
Subsystem C internally built & tested
Subsystem D internally built & tested
Subsystem C SSAT
Subsystem D SSAT
GEO 3.1 Rel
RELEASED Subsys C & D
Integration dependencies
UI
FSSys
MiscSys
Acquisition
Viewing
PandA
Geometry
database
Host
ID
IP
UI modules
infra
FAQ
Q: What if a subsystem must deliver at the same time different functionality to 2 or more customers?
A: That subsystem can only release if the last customer has released
Q: What if a delivery is not on the MIP but is already needed by a customer e.g. due to incompatible interfaces?
A: Either change the MIP or deliver that subpart as published to the customer.
Red Sticker
Green Sticker
SYSTEM TESTED
SYSTEM RELEASED
PDC 30
SRT
ART/SAT
Subproj TM: timing + Q Test team: execution
Zonnetjesplaat
all functionality mapped on subprojects (Phobos, Acquisition, Eagle, Movement)
Action points:
fRole system SCM
FATs
SYSTEM INTEGRATED
PDC 30.zz
System projects Warroom: accept / reject deliveries
System Regression Test
New functionality Test
SIT
Blueprint Development chain
system warroom - 4 subprojects
Production requirements
SRS
FRS
SDS
PRS (CIS only)
SSIS
Supply chain
subsystem chain per subproject
FAT
SIT
preSIT
SSAT
Architecture
units per subsystem
URS / UDS
UAT
MRS / MDS
MT
Code

5. Deployment – How to find the QMS

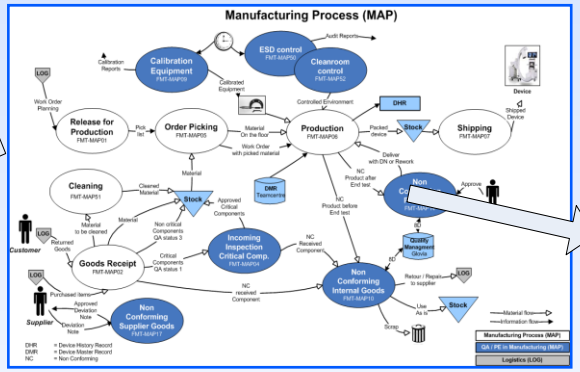
- Make several search possibilities
  - On terms, e.g. via google like search
  - On lists, e.g via the QMS index list
  - On pictures – e.g. via a clickable process
  - Via different views

5. Find the QMS – via clickable process

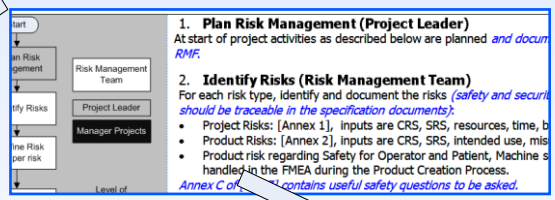
ALL PROCESSES



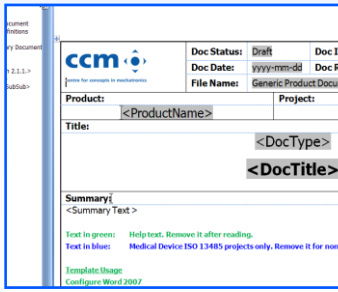
ONE PROCES

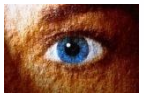


ONE PROCEDURE



ONE TEMPLATE

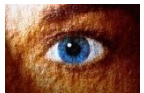




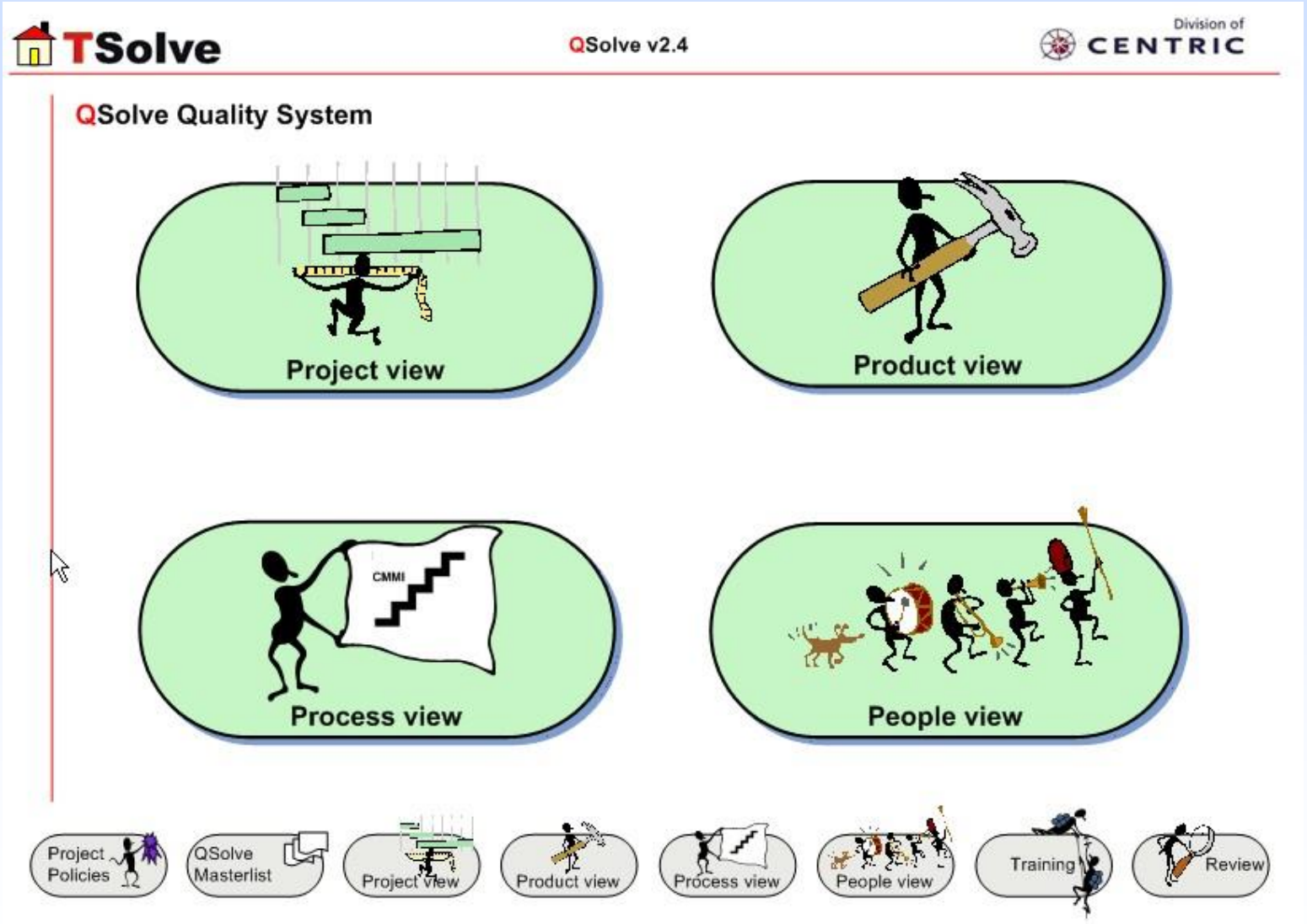
5. Find the QMS – template one click away

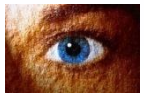
Project Acquisition	Templates	Project Financials
Orientation and Quotation 0400		Project Cost Control 8000
Order Acceptation 0500		Sales Invoicing 8300
		Financial Project Closing 8100

Product Creation (CCM)		Product Creation (MediSpirit)
Product Creation 0600		Quality Manual MediSpirit 0110
ESD handling 0630		Function descriptions MediSpirit 3
Clean room instruction 0631		New Product Introduction 7500
Software Development 0900		Guideline Labeling - Translation
Market Clearance 4900		Critical Supplier Control 6100
Guideline Machine Directive 4911		Clinical Evaluation 1200
Risk Management 0800		Market Clearance 4900
Customer Satisfaction 2000		Guideline Medical Device Directive
Nonconforming Outgoing Products 2100		Manufacturing 9000
Configuration Management		Nonconforming Products in Producti



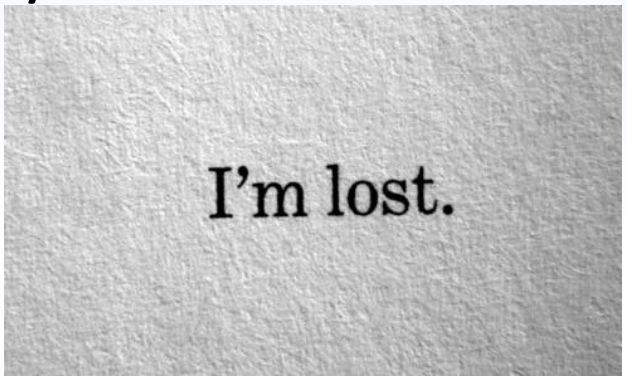
5. Find the QMS – clickable with different views

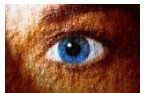




5. What I have seen gone wrong

- No deployment at all, just issue changes, nobody knows or notices it
- Deployment by saying ”Everybody can read the procedure“
- Deployment by going through the procedure step by step in a detailled manner
- People have no clue where to find the QMS
- And if they have found it, you cannot find what you search for



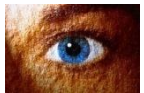


6. Change Control

While working with the QMS, the end users (employees) will submit change requests. How must an organisation act upon these requests? Deploy, deploy, deploy

6. Change Management

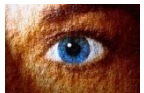
- Setup a change request tool
- Let users submit change requests in a easy way
- Form a Change Control Board
- Process the change requests on weekly basis
- If backlog occurs, you will loose user commitment
- Change QMS according to previous mentioned steps
- Release on regular basis, e.g. Quarterly and keep deploying
- **In the end, users must “own” the QMS not QA manager or process owner**



6. What I have seen gone wrong

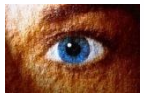
- No change control / configuration management / version control leading to
  - draft procedures being published
  - Templates being deleted by personell

- Large backlog on change requests leading to unmotivated personell to issue new requests



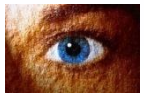
Putting it all together

Start a project if you want to create/maintain the QMS



Putting it all together – start a project!

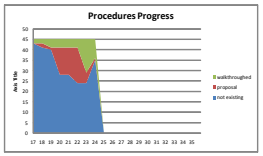
- Updating or making a QMS is a normal project
- Appoint a project leader
- Create a project plan and track progress
- Have regular meetings with stakeholders



Example of QMS development plan

ISO 13485 project Way of Working and Global Plan

Tracking



Walkthroughs of flow charts and proc interfaces at weekly progress meeting leading to informal updates (invite key users)

QMS publication board For pilot project(s)

Publish After walkthrough

Electronic Reviews (no meetings): should only reveal minor issues.

QMS publication board For pilot project(s)

Publish After E-reviews Or annex written

- ☐ Stage 1
- ☐ Training Done
- ☐ Management Review done
- ☐ Process audits done

☐ Stage 2 Certification

☑ Start Project — 01-2010

04-2010 Week 18 — Walkthroughs

08-2010 Week 35 — Official e-reviews Q Manual & procedures

☐ QMS approved — End Q4 2010

End Q1 2011

Define
- Q Manual
- Q Index
- Pilot project(s)

☑ Q Manual / Index walkthroughed
☑ Pilot projects defined

Process Owners:
Write (buddy with WBig)
- Parts of Procedure Interfaces
- Flow charts

WBig
- Total procedure interfaces
- Design webinterface

☑ All red and orange procedures synchronised with each other (proposal)

- Write annexes, check with 1 peer and publish
- Xtra text flowcharts (front, back)

Write / Executing Training. At least: Quality Policy

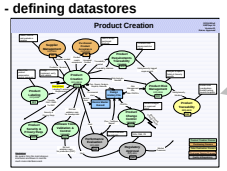
Gather evidence In pilot Project(s)

Annexes
- default English
- in Dutch if users are Dutch speakers only

Templates
Guidelines
Work instructions

Quality Manual
- Quality Policy
- Organisation
- Scope
- Processes
- Process descriptions
- Process Onwers

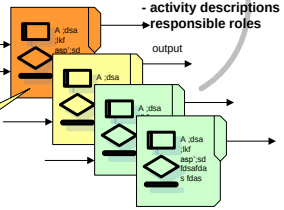
Procedure interfaces
- connecting flow charts input/outputs
- defining datastores



Update of A&D file by all owners

Abbreviation & Definitions file

Flow charts *(incl. blue italic)*
- scope
- input / output / datastores
- activity descriptions
- responsible roles

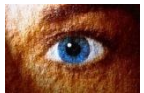


1 A4
Hints:
- merge sequential activities same owner
- activities in imperative sense

Quality Manual Index File
- procedures per process
- process owner

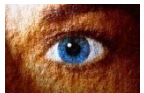
Procedure A process A
--- Template A
--- Work instruction A
--- Guideline A
Procedure B process B
-- Template B
-- Template C
Procedure C process A
Procedure D process C
-- Guideline B

Priorities:
Red = High (ISO, needed for project))
Orange = Medium (ISO)
Green = Low (non ISO, CCM wish)

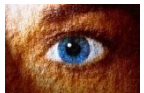


Summary

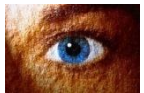
- Did this help to answer the question “What to do if our QMS is not used?”
- In short
  - Treat the QMS as a normal product to be developed, implemented, verified and validated, released, put under change control / version control / configuration management
  - Treat the way you develop the QMS as a normal project
  - Deploy, deploy, deploy
  - **The users must feel they “own” the QMS**



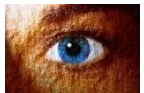
- InfoQ did and interview with Willem about the benefits of having a QMS, dealing with multiple regulations, assuring adherence to a QMS, how a QMS can support agility and deploying a QMS in an agile way
- A summary is to be found in the upcoming sheets.
- The complete interview can be found on http://www.infoq.com/news/2014/10/getting-QMS-used



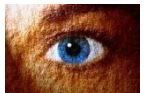
- In your view what is a QMS? What are the benefits of having one?
  - A QMS is the documented set of agreements about the way of working within a company
  - Without this agreement each design project and/or production team works according its own insights which can lead to a wide variation in the quality of products and inefficient work
- Can you name some of things that often go wrong when organizations are deploying a QMS?
  - Problems that I often see in organizations with Quality Management Systems are:
    - The idea that writing it down is enough
    - Only performing initial deployment
    - All deployment by the quality person feeding the idea that the QMS is his/hers responsibility



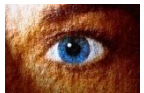
- From a governance point of view a QMS must assure that applicable regulations are being met. Do you have good practices when dealing with multiple overlapping or conflicting regulations or regulations from different standardization bodies?
  - You can make a cross matrix to check if all regulations are “in”
  - I always try not to split up (make separate) procedures but sometimes it can be more convenient: an example is a work instruction to bring a medical device to the market and a work instruction to bring a machinery to the market.
  - I have not seen any conflicting regulations yet as in principal they all want the same: are you in control



- What can you do to assure that a QMS is followed and adhered to in an organization?
  - First you need to do proper deployment
  - Next, check if QMS is followed by official audits and non official “audits” (like listening to people talking in the coffee corner)
  - You need to re-deploy if issues are found
- How can you validate a QMS to see if it supports an organization in reaching it's goals?
  - You can’t by validating as I described it (piloting) but you can by measuring on the processes. E.g. By means of GQM (Goal Question Metric) method



- If an organization is adopting agile or lean processes would they still need a QMS?
  - Yes! You always have output so you need templates. You always want a common way of working so you need procedures or work instructions
- What kind of changes are needed to a QMS to support agility?
  - Agility is just a way of working, so write down how you do it and put it in your QMS. I have seen organizations who keep agile development outside their QMS. It became a work instruction in each of their projects. That is not needed and can be ineffective.
- Could you develop and implement a QMS using an agile approach, e.g. in iterations in stead of a big bang deployment?
  - Of course, certainly if you don't know the exact requirements at the start. You could also develop it while the first project is already running, e.g. the first increment should deliver the requirements and the project management processes




The
End