



Validating LIMS in a GMP Environment

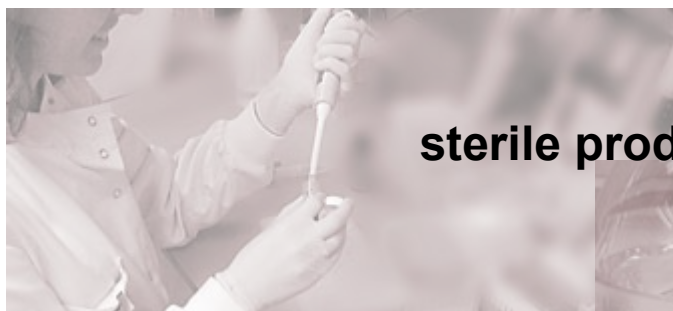


How To ...





Validating LIMS in a GMP Environment



**Support for
sterile production of solutions & lyophilizates of
peptide & protein hormones**



The Goal

Improve Data Handling and Service Quality

- Increase efficiency & productivity
- Reducing errors
- Increase regulatory compliance
- Improved process control

Long term: financial benefit



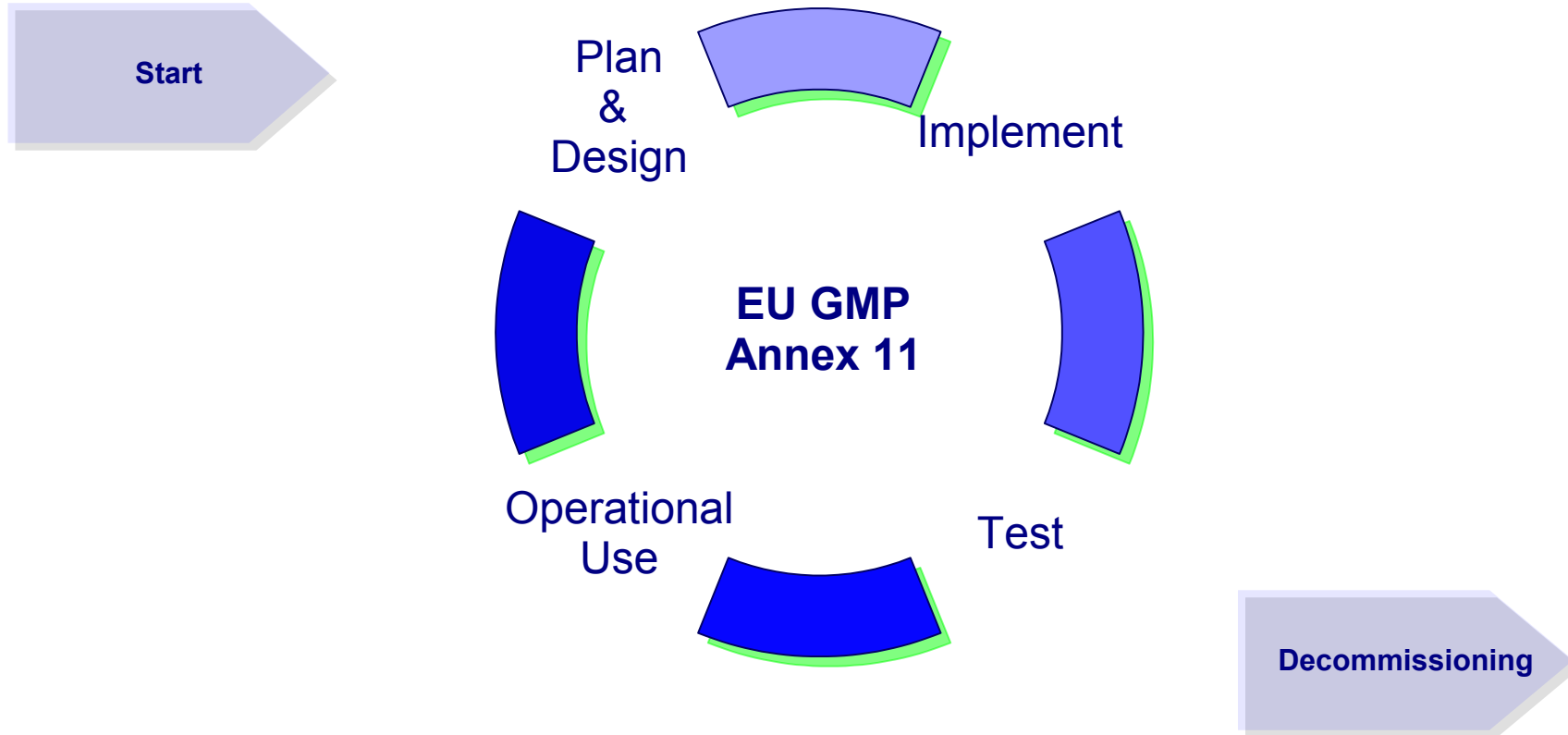
LIMS Team & Data Volume



Project Team (Core)	QA, QC, IT, Project Manager, System Owner, Vendor Implementation Consultant, Validation Consultant
Project Team (Implementation/ Validation)	System Owner: Qualified Person; Admin: QA (& Deputy); Power-User: QA, Analytics, Microbiol., Pharmacol., Production Val. Support: QA; Consult. IT-Support: local
Entry of Static & Dynamic Data & Validation	25 – 30 persons
Service Level Agreement	Internal & External, Warranty / Maintenance & Escrow
User Base 2005	~ 60 used Accounts, 15 concurrent licences
Data Volume in Production Environment 07/2005	Dynamic Data: Lots: 8920 , Samples: 34520, Tests: 74560, Results: 620510 Static Data: 3900 Product Specs, 860 Instruments, 1500 Reagents Traffic: about 2800 logins per month
	Implementation Project: July 2001 – June 2004

The Life Cycle

Software Validation (FDA): “Confirmation by examination and provision of objective evidence that software specifications conform to the user needs and intended uses, and the particular requirements implemented through software can be consistently fulfilled”



The Life Cycle

Principles of Validation:

Validation of computerized systems (EU GMP Annex 11):

Planning & Design: clear, complete, prospective!, different levels of detail, top-down approach

Implementation: systematic structured, robust, well documented
(reproducibility)

Testing: appropriate, risk based, clever

Operational Use: controlled, easy

Decommissioning: actively managed, keep accessible

FDA: “Because of its complexity, the development process for software should be even more tightly controlled than for hardware, in order to prevent problems that cannot be easily detected later in the development process” => Don't look for high degree of customized systems! Go for configuration!

Guidance



GAMP 4 (<http://www.ispe.org>)

FDA "Guidance for Industry Part 11, electronic Records; Electronic Signature – Scope and Application" (<http://www.fda.gov>)

PIC/S PI 011-1 „Good Practice for Computerized Systems in Regulated „GXP“ Environments" (<http://www.picscheme.org>)

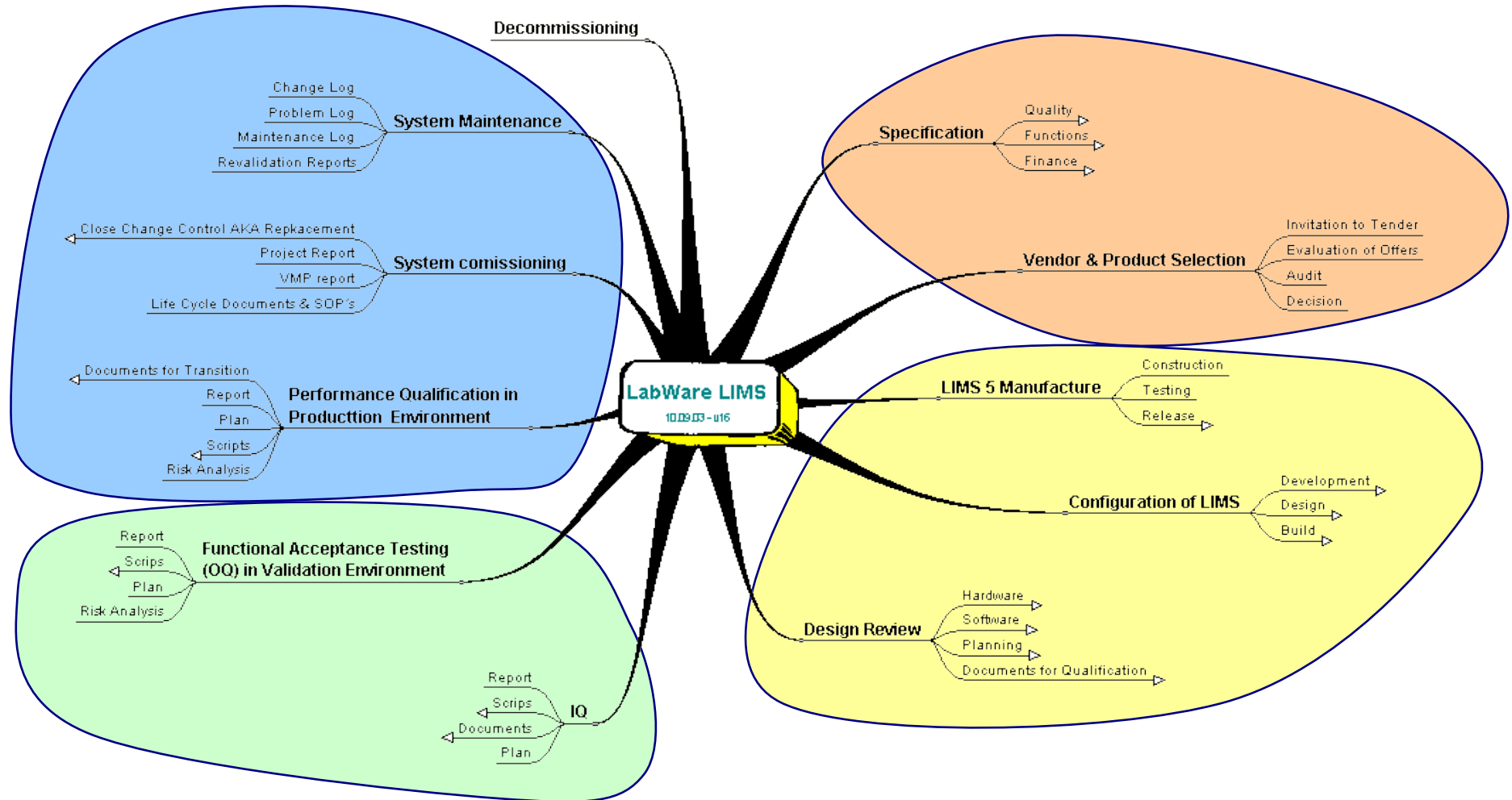
CDRH „General Principles of Software Validation; Final Guidance for Industry and FDA Staff“ (<http://www.fda.gov/cdrh/comp/guidance/938.pdf>)

ASTM E 2006-00 „Standard Guide for Validation of Laboratory Information Management Systems“ (<http://www.astm.org>)

And a lot more...

(e.g. see <http://www.bsi.de/english/gshb/manual/index.htm>)

Project & Validation Plan



Project & Validation Plan

Validation Plan: clarifies everyone's role, responsibilities, and the course of action to be followed during validation and lists documents that must be in place (also possible: in an extra item list).

You can't buy a validated system: it has to be done on site by the users!
Extent of validation to be defined by the users.

IQ/OQ may be combined: in IQ (and/or audit) check presence of proper documentation of vendor FAT (factory acceptance test).

Parts of the OQ may be within PQ! = SAT (site acceptance test).

Only validate the LIMS parts that will be used!

If possible: first implement LIMS core, then extend used functionality step by step (modular approach)

Distinguish between validation of LIMS as a whole system and validation of static data!

User Requirements Specification

URS lists what is needed

Description „As Is“ & „To Be“: flow charts for processes

Uniquely numbered requirements for ease of traceability

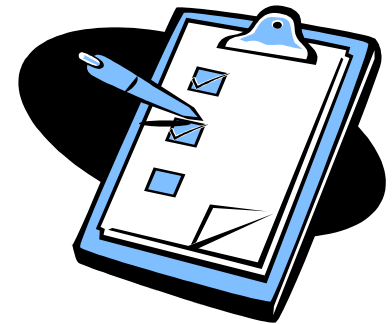
One requirement per entry

Testable, tabular format

Risk assessment

- Off the shelf LIMS:
Configurable Software = GAMP 4
(OS = 1, DBMS = 3)
Hardware (Network, Clients) = GAMP 1

Content: see GAMP



User Requirements Specification

Poor requirements management is generally considered one of the major causes for product failure. All Software development techniques will be of no use if the developer is not building the right product.

1. Understand the needs of the stakeholders: list candidates of features , relative technical development risk, estimate effort required, relative priority or importance (customers perspective)

2. Decide which features are the appropriate features to include in the software: developers estimate effort and time to satisfy the stated features until the work involved is compatible with schedule and budget but must consider also impact on acceptance etc. when removing features

3. Detailing the exact external behaviour of the system that will address the features selected: customers & all developers have same understanding of what to build, testers are testing the same qualities that are built, management applies resources to the right tasks

Specify and distinguish between 1) mandatory, 2) useful, and 3) nice to have specs

Consider measures from process risk assessment in system design



LIMS selection

Request for proposal: URS

Vendors propose solution

Evaluation

- balanced scores according to importance of requirements

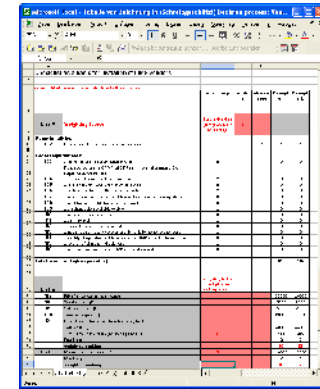
System presentation

- make use of end users feed back (interface Computer - User)

Vendor audit

- make use of shared audit reports, but watch scope!

Final selection & purchase contract



The image shows a screenshot of a spreadsheet application, likely Microsoft Excel, used for evaluating LIMS solutions. The spreadsheet has multiple columns and rows. Some cells are highlighted in red, indicating specific data points or scores. The text within the spreadsheet is small and difficult to read, but it appears to be a structured table for comparing different vendors against various criteria.



FS/DS & Prototyping

Describes how requirements will be fulfilled

Review URS with vendor

Involve users to the right level

- Prototyping
- Feedback from training
- Power users

FS/DS = final configuration – standard(vendors users guide)

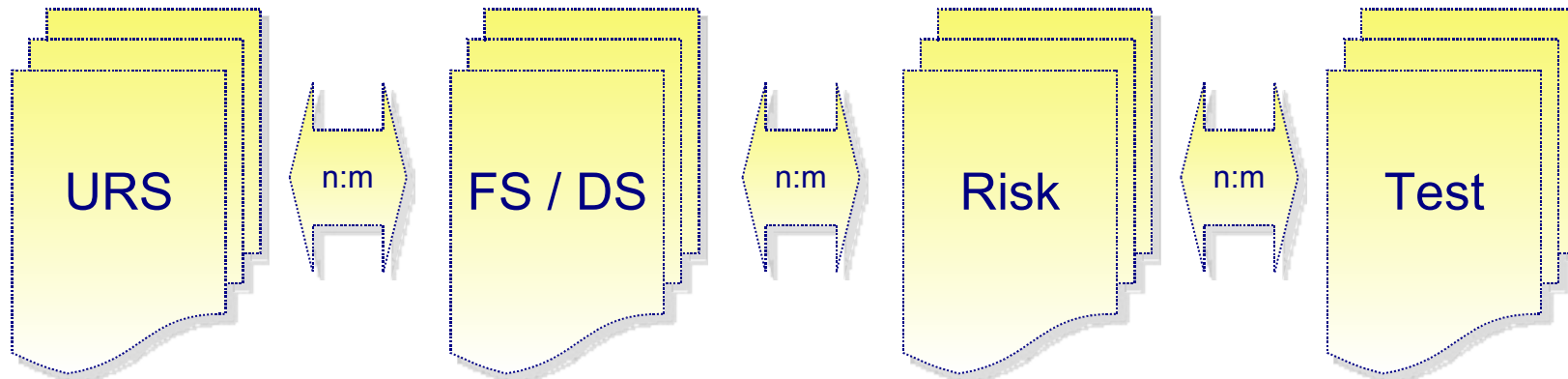
Avoid features implementation by bespoke code



Trace Matrix

Completeness, Accuracy

A relational DB may be used to map the connectivities



Risk Assessment

Early general GMP risk assessment

- Electronic records & electronic signature
- General business risk assessment

URS review risk assessment

- Tabular high level functional assessment

Process & system FMEA

- Impact weighting
- Setting limit
 - all functions with impact high to be tested
 - others are handled when errors occur



Risk Assessment

It is known that software testing has limitations and that it cannot be exhaustively tested. Because of the very large number of possible paths through a software program, 100% path coverage is generally not achievable. The amount of testing is established based on risk assessment!

FMEA:

scenario and failure mode / reason for failure:
impact (patient risk, data integrity, business)

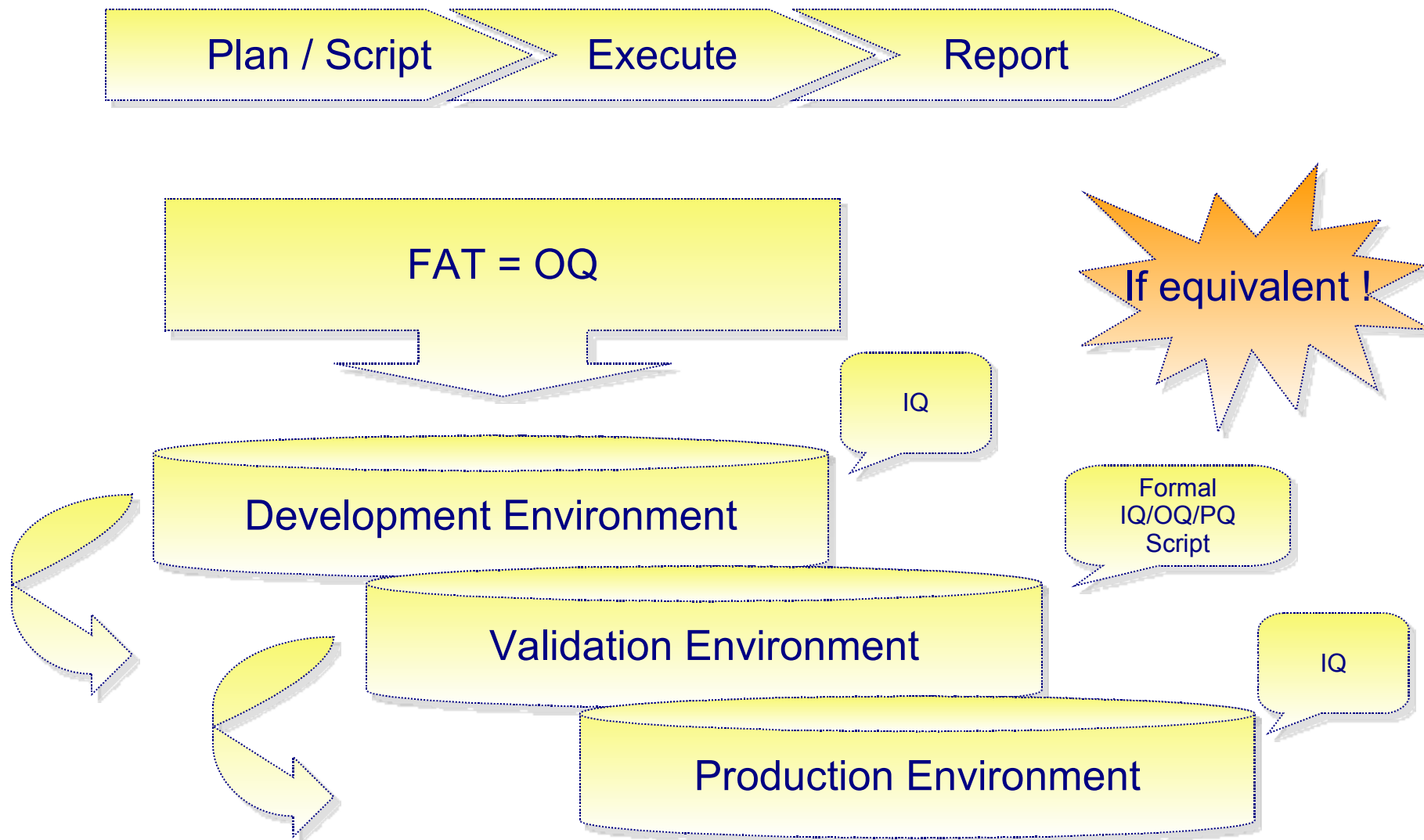
of error*likelihood of occurrence*detection probability = risk priority number

Reduce amount of testing by not testing combinations below a certain risk priority number (i.e. of low risk)

Better understanding of where are the weak points



Testing



Testing

Write test plan with acceptance criteria, submit for approval, testers execute it and document results, check results and write report and submit for approval.

Test protocols should cover all aspects of the system (also disaster recovery!).
FAT = release of standard product by vendor: have a look at this documentation during the vendor audit

IQ (checks deliverables including documents and files etc., and basic “work together” of components = HW & SW): has to be done for all GxP relevant environments!

OQ = Factory Acceptance Test (vendor); but check if test environment (vendor) is equivalent to customer site environment !

PQ = Site Acceptance Test (customer)

List assumptions (e.g. validated operation system running, work acc. SOPs), exceptions (what does not have to be tested why?) and limitations (tests done with special test case data but not with real data in a test environment) in plan!

Testing – continued -

Development environment: for informal testing and training and development of test scripts;

Validation or Test environment: used for formal testing according to script

Production Environment: correct installation / transfer has to be documented

Prerequisite for the approach: no relevant difference between the environments!

Coding & Customization: Code Review / White box / Unit / Integration Testing necessary; have to have coding guidelines etc.

Black Box / Stress in OQ/PQ: test approach = along business processes; tree; risk based; all in OQ, selected functions risk based in PQ; normal/boundary + robust testing

Regression (executing same test script again): after changes e.g. adding bug fixes

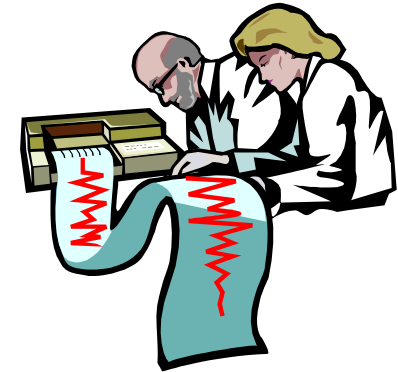
Traceable: URS – PQ, FS – OQ, DS – IQ

Use similar approach for Static Data Entry

(Test driven development; RTCA/DO-178B : Test tree methodology)

Static Data

Validation of Analysis & Specifications
(see also Change Control and PQ-Plan)



- Informal testing (development)
- Entry of real data & check of calculation results (val.)
- Approval routing (productive use)
- Keying in critical data (results, lot numbers) needs a second check (by person or automatic validation function)!



Transition Phase & Going live Operation & Maintenance

Configuration Management / Item List (list all files, modules, ...)

SOP's

- Security (grant & revoke access rights, ...)
- Maintenance
 - Service level agreement (SLA)
- Backup & restore, disaster recovery
- Change control / error handling
- Operation & usage (URS and FS/DS may be of use!)
 - Establish system management group / power users
 - Own manuals on top of vendor users guide
- Training plan
- Archiving

Training & User Base Structure

Training according to plan

- \geq 3 levels of users
 - Admin / power user / end user
- Online user manual (Vendor +)
- Online information base (Lotus Notes database)
 - Discussion Forum
 - Knowledge Base
 - Wish List
- Training groups according to application area.
- Perform training not too long before skills can be applied and used.
- Step by step: Theory & background training – practical training (sandbox) – use system – refresher and expanded training if needed.
- Encourage knowledge exchange and publish information and training material (e.g. on intranet).



Access Control & Access Rights

User: application form

- assisted by power user / admin

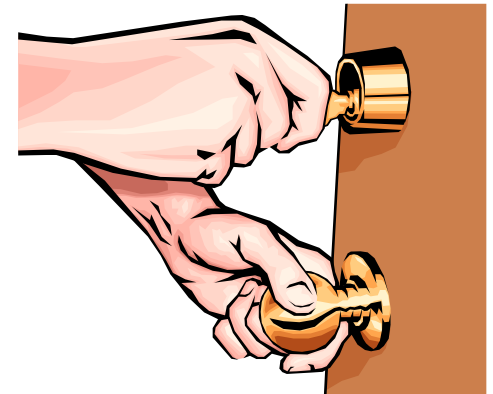
Supervisor approves

LIMS owner approves

Implementation

- IT responsible adds user to software deployment list
- LIMS admin implements LIMS rights & hands over password

SOP how to handle passwords, password loss, attempts of unauthorized access, and who is responsible for what



Error Handling

Register errors & aberrations

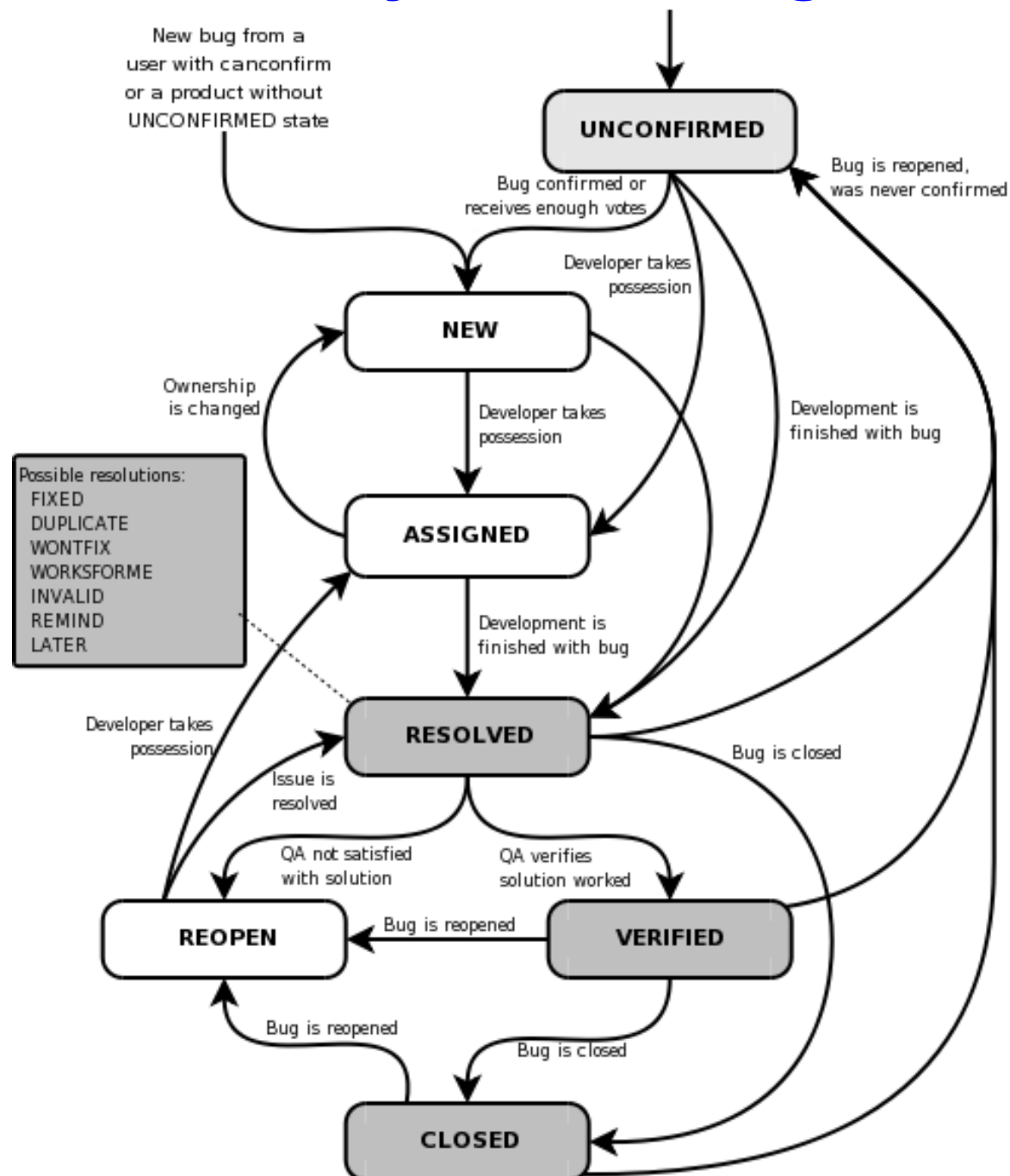
- eMail / by phone / Lotus Notes...

Log errors & assess

- Unique number & register in form
- Classification
- Contact Vendor support if necessary
- List of unresolved bugs and open issues has to be public!



Life Cycle of a Bug



<http://www.bugzilla.org/docs/>

Changes

Implementing bug fixes

Adoption to external changes

Extensions: new modules / functions, interfaces

Revalidation strategy

1) Design

2) Risk Assessment

3) Testing

4) Implementation

5) Use



Changes

Development

Informal testing by author / users

Set up Change Log

Risk assessment: what is affected? (trace matrix may be useful)

Set up of test plan

Migrate into Validation Environment

Validation

Perform Tests according to plan

Document Test Result (Change Log)

Migrate into Production

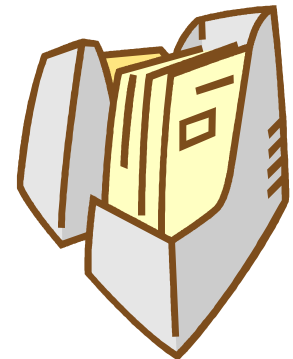
Production

Use after Approval



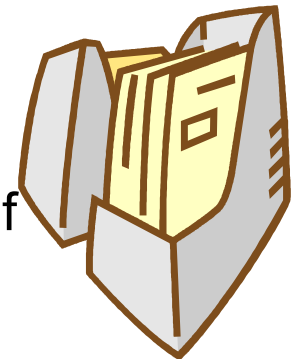
System Retirement & Archiving

- Predefine retention period
- Only authorized access to archived data (for offline and online archives!)
- Data format: readability problem (long term storage): validated convert / export in common or flat file format (e.g. XML, PDF, ASCII, CSV, ...)
- Stability of storage media: long term data integrity preserved by copying (MO to MO, server to server,...)
- Protect against change without audit trail: use read only media
- Decision to delete data must be documented!



System Retirement & Archiving

- Availability: Data may be still needed (e.g. for audits; see statutory obligations and business needs!).
- Active archive management and maintenance is needed: convert / copy / migrate and check data integrity on a regular basis
- Paper reports rarely represent all meta data; risk assessment as basis & justification for decision of what not to archive
- Define: what is raw data?! Binary form, human readable form
- For general information see also http://www.glp.admin.ch/legis/ArchElectRawData1_0.pdf



Everything under Control

