empowerment
quality engineering

Cross Pollination Benefits Computerised Systems

Cross pollination can be defined as: "To use processes, procedures and tools from other businesses in order to gain an increased benefit". Why would the pharmaceutical industry want to look at the activities from other vertical industries?

The traditional pharmaceutical project manager and validation expert are very apt at qualifying and delivering pharmaceutical projects (production lines, manufacturing API's, building factories, cleaning down rooms). When applied to computerised systems, particularly GAMP category 4 and 5 systems, it is often the case that problems occur: long schedules; major cost overruns, initial poor quality and dissatisfied users. In the end, a validated state is achieved but with some added pain. Computerised systems are inherently very complex to produce, (even a simple system can take a number of months) and require some additional assistance in order to ease the burden and increase confidence. This can be seen by the support provided by the guidance bodies, such as GAMP.

Very often, the computerised system is only fully challenged by validation experts and end user when it reaches the OQ stage. Any issues found at this point results in high over heads: defect raising; root cause analysis; documentation rework, review and sign off; code updates; independent verification; release; IQ; regression tests; OQ; identification of ripple bugs (bugs that arise as result of the fix to related piece of functionality); more documentation updates and further overheads. Compounded to this is the complexity of software – although complete coverage of the user requirements have been met, have all of the underlying technical risks been identified and catered for?

Software development and maintenance became major corporate concerns in the latter half of the 20th century as most companies could not compete successfully without software. Although there are numerous examples of companies being beset by the troubles of poor software, there are also examples of companies being very successful in bringing software under control. The best example, the telecommunications manufacturing industry has been one of the leaders in managing software production. They have realised long ago that a successful business depends upon quality and reliability of the equipment they produce. After all, their hardware is built to run at over 99.99% of the time and their systems must allow emergency calls through at all times. We see them as the pioneers of complexity and also quality within the wider software engineering community.

The number of procedures, tools and activities available that can be of benefit is staggering. For example, studies have identified that there are over 50 software methodologies in use in 2008. So if other industries have been successful in providing quality computerised systems then why not take aspects of their approaches and apply them to the pharmaceutical sector?

Examples of Cross Pollination:

Accurate defect reporting is a key element in improving quality. Analysis of the defect has led to some innovative defect prevention and defect removal operations in many companies. The careful measurement of defects and the subsequent analysis of data is one of the most cost effective activities a company can perform and is generally one of the primary areas of focus for process improvement.

Front loading: This is the term to describe the re-alignment of the test focus from after the code effort has occurred to the whole computerised system project, e.g.

cont....

Sample of Procedures and Processes for Cross Pollination
Methodologies
Extended V-Model
DSDM
Rapid Application Development
Agile
Prototyping
...
Management
Prince II
PMBOK
Scrum
Test
Quality
...
Requirements
Analysis
JRD
Risk Analysis
Use Cases
User Stories
Traceability
...
Design
Unified Modelling Language
Data Flow Diagrams
Entity Relationship Diagrams
Normalisation
High Level Design
Low level Design
Design reviews
...
Code
Reuse
Package purchase
Code Inspections
Test Driven Development
Pair programming
Compiler outputs
Static Analysis
Standards
Object Oriented guidelines
...
Testing
Formal Testing
Functional Testing
Non Functional Testing
Unit Testing
Integration Testing
System Testing
Field testing
Acceptance Testing
Independent Testing
Static Testing
Reviews and Inspections
Automation
...
Distinct Roles and Skills
Design
Independent Verification
Independent Validation

Sample of Procedures and Processes for Cross Pollination
Independent Validation
...
Installation and Commissioning
Maintenance and Support
Support Processes
Configuration Management
Defect Management
Version Control
Release Management
Branching & Merge Management
Project Change Control
Process Change Control
Training
Meetings
Risk And Issue Management
Customer Satisfaction
Team Assignment Guidelines
Project Estimation Guidelines
...
Quality
Test Process Improvement
SPICE
CMMi
ISO 9001:2008
Six Sigma
Software Project Control Audits
Process Improvement
Metrics
Defect removal Efficiency
Running Tested Features
Defect Reporting Analysis
...

static testing of requirements; static testing of design; incremental software development; applying technical test design techniques; deploying static analysis tools; testing the installation approach from the first code delivery, ensuring independent and skilled technical testers. Basically, testing early and testing often so that defects are detected (or even prevented) sooner. Front loading is a key attribute in achieving Right First Time Validation.

Quality by Design: Thinking in software engineering is moving from "how can I develop a solution within time and budget" to "how can I get the solution properly tested and within time and budget". The attributes that are recognised as good coding (low coupling, code reuse, normalised database tables, designed software interfaces, trace and logging statements), are also things that facilitate easier testing. If the testing is easier, then it will be more thorough. If it is more thorough then the software will have a better coverage and a better statement understanding of its level of quality. This is what the software engineering community mean by "is the requirement testable" at the design and code stage. Some observers believe that this is single most important thing that can be done to achieve quality software is to design the quality in – from the start.

Computerised System Project Control Audits: How effective are the processes and procedures being used by the computerised system development team? How effective is the team at working towards them. Performing a computerised system control audit that is based on a wider software engineering audit can pay dividends to the outcome of the Validation effort by identifying gaps in the internal/external suppliers approach. Decisions can be made whether to continue as planned, instruct on additional Quality Assurance and Control activities to be performed by the supplier or the Validation team.

By combining the wider software engineering community approaches with the regulatory compliance needs of pharmaceuticals, the costly over runs of computerised system projects can be consigned to the past. The benefits to Validation is the leverage of independent expertise, a faster Validation cycle, no high priority issues, less documentation and a faster delivery of the compliant, business critical computerised system.

Why are computerised systems so complex?

The building of an acute hospital of 500,000 sq ft is a complex undertaking. There are dozens of different kinds of codes and inspections, such as electrical codes, environmental codes and plumbing codes that must be adhered to. Many types of experts will be required for such a large building as generalists are not skilled enough to install specialised items such as electronic transformers and wiring or radiation equipment. Independent verifiers are required to ensure that the equipment has been properly installed. The façade of the building and its interior design may be what most visitors see, but there is much more hidden complexity to the hospital. Software is the same: the User Interface hides a multitude of complex inner workings that requires specialist skills to design, build and maintain .

Empowerment Quality Engineering origins lie in the telecommunication manufacturing industry and we have successfully combined the wider software engineering processes and approaches with the regulatory needs of the Pharmaceutical industry. By doing so, we produced immediate benefits: reduced costs, on time schedule, end user satisfaction, regulatory compliance and enhanced quality. Our longer term benefits include increased computerised system development, maintenance and Validation through put.

Empowerment Quality Engineering Ltd.
Omagh Business Complex | Great Northern Road | Omagh
| N. Ireland |BT78 5LU

w. www.empowermentqe.com/pharma
t. +44 (0) 28 82 25 19 50
m. +44 (0) 77 80 43 29 53
e. ciara.mcmanus@empowermentqe.com