



26th Annual ASQ Audit Conference
Blueprint for a Successful Audit

October 9 - 11, 2017 • Pre-conference Courses
October 12 - 13, 2017 • Conference

ASQ

The Global Voice of Quality™ The InterContinental Hotel | Dallas, Texas



ASQ®

Audit
Division

The Global Voice of Quality™

Comparing Software Quality Auditing and Software Quality Engineering

Mark Neal

Stuart Yarost

12 October 2017

Objective

- Describe the approaches for quality oversight of software development projects: process auditing, product auditing and quality engineering
- Compare the attributes for each approach
- Share the benefits of software quality engineering embedded in development teams
- Describe how to obtain management support to establish software quality engineering and based on pilot project success, expand quality engineering involvement in development teams
- Identify the role of process and product auditing when quality engineering is applied in software development projects

Note: The principles described in this presentation may be applied to any product development environment (mechanical, electrical, etc.) by removing the occurrences of “software” in the text.

Software Process Auditing

- To ensure the product development process is compliant with established procedures, work instructions, standards and any applicable regulatory requirements
- Usually performed on a periodic basis (e.g., yearly) independent of project schedule/milestones
- Samples from key deliverables for several recent development projects, in-flight or historical, to establish evidence of procedural compliance
- Personnel performing process audits may or may not be technically competent in software development or the environment of the project (coding language, development tools, etc.)
- Usually results in audit findings which may trigger rework of the audited deliverables and/or revisions to procedures and in the worst case trigger a field action to revise the fielded software

Software Product Auditing

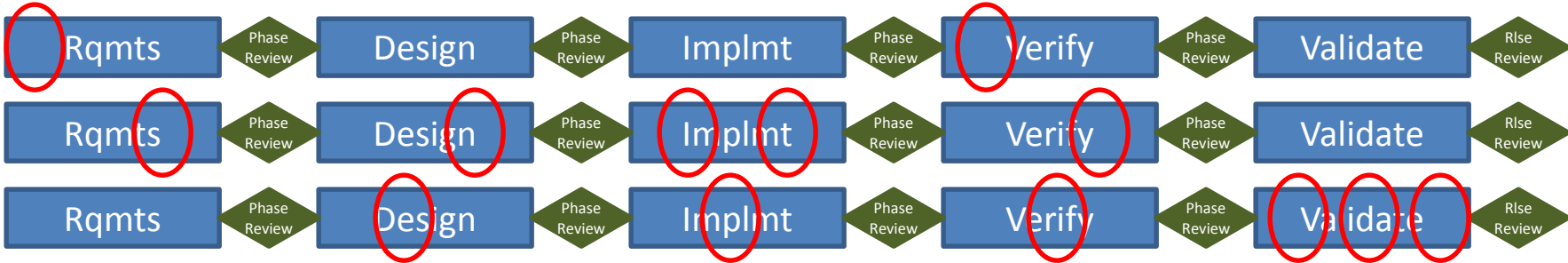
- To ensure the product development process is compliant with established procedures, work instructions, standards and any applicable regulatory requirements and to ensure technical content is supported by documented rationale
- Usually performed on a per-project basis at key project milestones or development phases to determine readiness to move to subsequent development phases or readiness for product release
- Reviews deliverables developed during the phase(s) being audited to establish evidence of procedural compliance and adequacy and completeness of technical content
- Personnel performing product audits must be familiar with software development and the environment of the project (coding language, development tools, etc.)
- Usually results in audit findings which may trigger rework of the audited deliverables and/or revisions to procedures and in the worst case may delay progression to subsequent phases or product launch

Software Quality Engineering

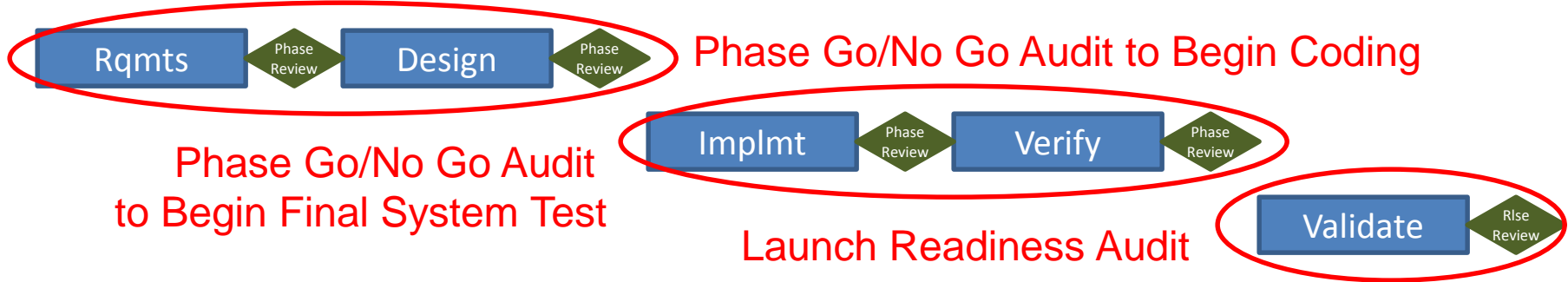
- To ensure the product development process is compliant with established procedures, work instructions, standards and any applicable regulatory requirements, to ensure technical content is supported by documented rationale and ensure “ilities” (reliability, maintainability, quality, etc.)
- Performed continuously as an embedded core team member on a per-project basis and integrated throughout the project activities and deliverables
- Participates in project activities, reviews and deliverables throughout the project timeline which ensures procedural compliance and adequacy and completeness of technical content
- Personnel performing Software Quality Engineering function must be technically competent in software development, the environment of the project (coding language, development tools, etc.) and respected by the development team
- Reviews of work products and quality oversight feedback for corrections are performed in real-time as the work is performed

Quality Oversight of Software Product Development - A Graphical View

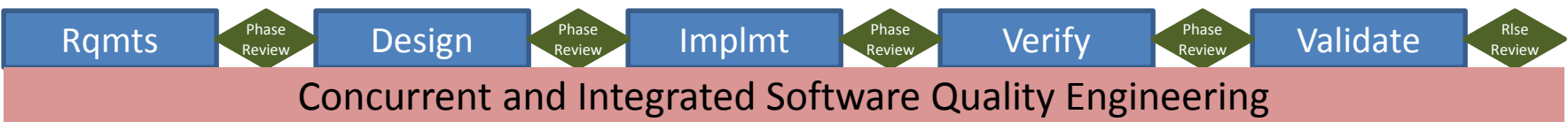
Software Process Auditing Example



Software Product Auditing Example



Software Quality Engineering



Software Quality Engineering Benefits

- **Acts As Process “Shepherds”** – Software Quality Engineers ensure compliance by participating in software product development activities and deliverables and improves process understanding by Development
- **Provides Independence** – Reporting structures outside the project team and usually outside the Product Development organization ensures that issues can be raised and addressed without fear of repercussion
- **Adds Value** - Perceived by the team as “value-added” contributors versus “policemen” – Software Quality Engineers bring “Voice of the Customer”, human factors and other quality principles to the development team
- **Gives Timely Feedback** - Issues are raised in real-time and addressed before significant downstream work would be affected
- **Ensures Quality Requirements Planned Up Front** – Project plans or Quality Plans are established up front to define all compliance and quality requirements, resulting in less rework due to missing activities or deliverables late in the project

Software Quality Engineering Benefits (cont)

- ***Prevents “Last Minute” Surprises*** - Software Quality Engineers involvement throughout software development results in few, if any, “showstoppers” at the time of product release to the field
- ***Implements Process Continuous Improvement*** - Fewer defects escape to the field through Software Quality Engineer oversight and integrated review of requirements, design, coding and testing activities and deliverables and management of in-process defects identified
- ***Continuously Mentors Engineers on Process*** – Continuous interaction with Software Quality Engineers provides greater understanding of the “Why” of the processes, reducing defects due to misunderstanding in new situations
- ***Reduces “Us vs Them” Attitude*** – Working together reduces the barriers between Quality and Engineers, allowing earlier and more friendly interactions, leading to fewer defects escaping to the field
- ***Supports Agile/SCRUM Environments*** – Software Quality Engineers can help modify the process and deliverables to support any selected methodology

Implementing Software Quality Engineering

- Senior management responds to \$\$s, time to develop/launch products and impact to customers after launch
- Review recent product releases and estimate time, wasted labor costs and/or revenue losses or delays due to process or product audit findings identified late in the project, high defect counts, etc.
- For products regulated or subject to international standards, review recent actions by regulatory authorities, ISO registrars, Notified Bodies, etc. and quantify the \$\$ or time impacts to resolve
- Software Quality Engineering pays for itself as a return on reduced rework due to audit findings and reduced defects in the field which may result in rapid unplanned releases to correct issues

Implementing Software Quality Engineering (cont)

- Pick a project or program as a “pilot” to assign a full-time Software Quality Engineer:
 - Key project large enough to warrant a full-time SQE
 - Project planned for less than 6 months to allow significant impact, but provides timely feedback to senior management
- Assign a technically competent SQE to the project; should be ASQ Certified Software Quality Engineer
- Develop and track metrics on the affect of Software Quality Engineering
- After 1 or 2 successes with metrics showing improved product release experiences, senior management will be more willing to expand the approach to other projects
- After some time where software quality engineering is applied to all projects, procedures and work instruction should be updated to define roles and responsibilities

Role of Auditing With Software Quality Engineering

- Process Auditing
 - Process auditing will likely always exist due to:
 - Internal audit requirements driven from Corporate/Division policies, regulations or standards
 - Regulatory audits, if applicable, will continue
 - ISO Registrar/Notified Body audits, if applicable, will continue
 - Embedded Software Quality Engineers will periodically audit all processes, and audits should be of shorter duration, since the SQE is familiar with the programs.
 - The time to prepare for the official audits should be significantly less, since the in-process audits have been occurring at regular intervals.
 - Process audit findings should be significantly reduced and lower severity from prior to implementing concurrent and integrated software quality engineering

Role of Auditing With Software Quality Engineering (cont)

- Product Auditing
 - In general, internal product audits can be eliminated when concurrent and integrated software quality engineering is implemented
 - Product audits may still be applicable:
 - Regulatory product pre-approval audits, if applicable
 - If requested by senior management for critical programs
 - Note: In this scenario there is a Quality function evaluating another Quality function – usually not something you want to do
 - When requested by the customer as part of the contractual agreement

Quality Oversight Comparison

Practice	Process Auditing	Product Auditing	Quality Engineering
Personnel Training and Skill Set Focus	Auditing	Auditing and Familiarity with Software Development	Deep Technical Competence in Software Development, Language, Tools, etc.
When Performed	Periodic, Not Associated With Project Milestones	Associated with One or More Project Phases	Continuous, Concurrent and Integrated
Review Approach	Select Samples from Recent Projects	Select key deliverables and documents from project activities for the phases audited	Participates in the development and review of key deliverables and documented project activities

Quality Oversight Comparison (cont)

Practice	Process Auditing	Product Auditing	Quality Engineering
Responsibility for the Project or Product	None; not responsible for project or product	None; not responsible for project or product	Member of the project team and equally responsible for project timeline and product success
Independence	Independent	Independent	Independent
Impact on Product Quality At Launch	Little Impact - primarily ensuring process compliance	Some Impact - ensure required deliverables are present and activities performed	Significant Impact - involvement in reqmts, design, coding and testing throughout project
Impact on Project Timeline	Depends - due to nature of findings and point in project schedule or past project	Negative - can delay progress to next phases if findings are severe	Positive - Keeps timeline from slipping; compliance and technical content is reviewed in real-time; few surprises

Quality Oversight Comparison (cont)

Practice	Process Auditing	Product Auditing	Quality Engineering
Impact on Fielded Products	Severe findings may trigger rework of deliverables, or worst case, unplanned release to correct deficiencies	Little impact as product audits are conducted prior to product launch	Little impact as software quality engineering is implemented concurrently and integrated with project plan
Impact on Development Culture	Sporadic; long term culture change difficult	Punitive; culture change not likely long lasting	Collaborative; process understanding increases with each development cycle; can lead to “we wouldn't do it any other way”

Conclusion

- Concurrent and integrated software quality engineering is much more effective than product audits at ensuring compliance with procedural and regulatory requirements throughout the product development process
- While concurrent and integrated software quality engineering requires investment in SQE resources, it will pay for itself through benefits of reduced timeline impact of rework to address continuing process audits, reduced defects at launch, reduced issues in the field and improved customer experiences with the product
- If your company is currently performing product audits for quality oversight, consider shifting to a concurrent and integrated software quality engineering approach
- Software Quality Engineering may have a significant impact on the development culture of an R&D team and thereby reducing defect density, schedule, cost and potential for recalls

Questions?