

Information Quality Assurance

A PASSION FOR PREVENTION

Based on the information that we've been able to gather about QA project success stories, We have a good foundation for something more than just a self-serving case study. We can talk about the importance of injecting a "quality culture" throughout the IT organization, and provide 5 recommendations for turning QA from a dirty word to rallying cry. I don't like the word "injecting" by the way.

This mini white paper is a "just-in-time" learning document. The idea is that we can showcase our knowledge leadership in 3-5 minute segments with these 2-page, bulleted documents that educate rather than sell. The great thing about these is that the content can be re-used for other purposes such as trade publication articles, seminars, etc.

INJECTING A QUALITY CULTURE WITHIN YOUR IT ORGANIZATION 5 RECOMMENDATIONS

1. Make requirements Testable by practicing Static Testing.
2. Generating ROI based Metrics should be a part of DNA.
3. Budget & Implement Early Automation
4. Engage Testing early and Graduate from "Corrective to Preventive" Validation.
5. Expose applications to derive Quality at NO Additional Cost.

1. Make requirements Testable by practicing Static Testing.

- Start testing with the requirements phase
- 20-40% of defects are directly attributed to requirements, why wait and only test after development? Start at the source.
- Static testing documents that requirements are robust, stable, and free from ambiguity or perception issues. That prevents defects from occurring in the first place. Use SMART Principle:
 - o S - Specific
 - o M - Measureable
 - o A – Achievable
 - o R – Relevant
 - o T –Testable
- Example – user needs to log in. Where? How? Both username and password?
- Why hasn't this caught on?
 - o Focus is still to test AFTER development
 - o Feeling that this is a waste of money
- Reality
 - o Testing team – builds better test cases, in less time, lower cost money
 - o Development team – writes stronger code with fewer defects, more efficient, lower cost.
- Log all defects, not just ones that have a development impact
 - o Lessons learned for future projects
 - o Helps show ROI on defects prevented vs detected and repaired later

2. Generating ROI based Metrics should be a part of DNA.

- Shift focus from “what are you doing” to “what can you do better” through definition and use of metrics
- 3-level dashboard – speak to each audience differently, relevant to their roles. Does the CEO/CIO really care about the number of defects? Hardly.
 - o Sr. Management – Total cost vs budget, value realized
 - o Unit Management – milestone measures such as schedule adherence, overall results, etc.
 - o Project Leadership – productivity measures such as number of test cases, defects, hours, etc.

3. Budget & Implement Early Test Automation

- Everyone focuses on trying to automate all testing, but it may not always be the right solution. Probably more effective to have a mix of automation and traditional testing.
- Four Automation myths
 - a. Automate anytime - need to plan and budget when you plan project, so the right resources and tools can be utilized.
 - b. Automate everything – reality is that you can't hit 100%. Reality is more like 60-70%, then you start losing efficiency and increasing cost.
 - c. Immediate ROI – takes 3 cycles, sometimes more. Another reason why proper planning is critical, to maximize longer term ROI.
 - d. Automation tools – one size fits all. 4-5 key tools cater to unique needs and platforms. Win-Runner good in Java environment, QTP good in .NET. Neither are optimal for mainframe environments. Let requirements guide you choice, don't let internal or vendor pressure force a decision.

4. Engage testing early and Graduate from “Corrective” to “Preventive” Validation.

- Don't hold your development team accountable for unit testing.
 - o Developer writes the code, tests to see that it performs to his/her perception of requirements.
 - o Then the tester finds more defects in 2nd stage.
 - o Need to have neutral testing function partner with developer. Developer has more time to develop.
- Don't expect your development team to share defect information.
 - o Usually assumed to just be part of dev process, rather than potential lessons for others to learn from. Don't want to lift the proverbial skirt.
 - o Entering defects into org's test management suite will help other groups, other projects, learn from mistakes. This is a good thing.
- Don't enter next phase, system testing, until unit testing is complete. May take more time in short run, but starts to build a focus on prevention in early stages rather than detection in later stages.

5. Expose applications to derive Quality at NO Additional Cost.

- Divide your testing budget throughout the development lifecycle
 - o You already know that cost of defects escalates the later they are detected
 - o If \$100 spent after coding is complete, shift \$50 to requirement and development testing. Find that overall budget savings to be 20% or more due to fewer defects, more preventions.
 - o Be patient. Shift slowly over time, using phased approach.

GENERAL INFORMATION FOR IT EDUCATIONAL PAPER:

Also reality that the short-term expense may be higher to shift from detection to prevention focus. Longer-term savings in direct budget costs, but also indirect customer satisfaction, team efficiency and motivation. Be the visionary to see the long-term value.

- Preventing a defect saves money. Finding one only identifies money that has already been wasted.
- Empower QA at every stage of the development process, from requirements to design to development to testing.
- While a typical success measure, the number of identified defects is actually a measure of failure. Every detected defect is something that should have been prevented upstream.
- **Old way of thinking** – are we going into production with defects we can't see?
- **New way of thinking** – how many defects did we prevent?
- **Need to train development team to prioritize** – it's much easier in the long run to prevent defects than detect and remediate, facilitates closer working relationships between development teams, and makes everyone more successful. Enthusiasm is contagious
- Use QA to help everyone throughout SDLC to understand the impact of quality, rather than as the auditors over the development process.
- IT leaders should be quality evangelists rather than defect detectives. Get complete organization to buy in and rally around.
- **People, process, and passion** – empower the team
 - o Old – not my responsibility
 - o New – offering suggestions, actively seeking input from peers, team effort.

ABOUT SCINTEL

Run by CIOs, Scintel partners with clients to understand their business environments, develop innovative applications and customized business processes, manage and enhance existing technology infrastructure and convert raw data into knowledge. Through a deep expertise in IT operations and a sustained focus on best practices, Scintel helps clients around the globe engineer a strategic IT direction, and then applies the knowledge and the expertise to execute with precision. We solve real business problems by developing real business relationships.



QA@scintel.com



1-866-scintel



www.scintel.com