

Critical Interaction of Leadership, Tools, Process and Collaboration for Driving a High Performance Culture of Quality

Helen McKinstry
Director of Development Transformation
and Total Quality, IBM Corporation

PSQT North
12 September 2006

Picture the following....

- A large, diverse software development organization
 - Spread across multiple locations worldwide
 - Differing time zones (from 1 to 12 hours different)
 - Differing cultures
 - Technical and communication challenges in working with remote colleagues and projects
- Change is continuous
 - New projects are started; new teams formed
 - Acquisitions, mergers and reorganizations
 - New technologies



The Challenge

- To create a high performance culture of quality
 - One that consistently adopts best practices
 - One that demonstrates a passion for client success, and delivers products and solutions that delight our customers
 - One that fosters innovation
 - One that continuously strives for improved processes and development efficiencies
- And one that adapts to change

Classic Challenges to a Culture of Quality

- Schedule pressures
 - Unwillingness to invest in change or innovation in the face of tactical pressures
 - Diverse teams with differing practices
 - Lack of process or varying processes
 - Geographically dispersed organization with unique tools or infrastructure
 - Acquisitions, reorganizations
- Result: a loose federation of practices yielding widely varying results

How Do You Build a World Class Software Engineering Culture?

- It requires complementary ingredients
 1. Clear leadership priorities
 2. Disciplined execution of process or approach
 3. Broad adoption of tooling and best practices
 4. A community of collaboration with the desire to innovate and continuously improve
- Its all of the above, in cooperation

1. Clear Leadership Priorities

- Individuals and teams respond to what they perceive their management priorities are
- People need to feel their work is valued and appreciated
- Mutual accountability for success can be achieved with a common vision articulated at all levels of management and leadership

2. Disciplined Execution of Process or Approach

- Which approach or process?
 - It doesn't matter! (Say again?)
- Well, yes, it does...
 - There is a great deal of evidence that disciplined software engineering methods or approaches lead to improved quality and productivity.
- But, success can be achieved through any of several various approaches, *if*
 - Consistent discipline is used in the application of the approach
 - There is a shared objective of quality and client success
 - Innovation is encouraged, to facilitate learning and improvement

A Note on Discipline and Agility

- Methods of development
 - Agile development refers to a style of iterative development
 - Disciplined development often refers to plan-driven or waterfall approaches
- Misnomers
 - False: Using **Agile** methods implies a less-disciplined approach
 - A project using agile or iterative development methods, focusing on simplicity and lightweight approaches, does not imply a lack of self-discipline, structure or skill.
 - False: **Discipline** in development implies heavy-weight processes which do not permit agility or flexibility
 - Processes can be used with discipline while still permitting flexibility to make the appropriate decisions regarding implementation on a specific project.

3. Broad Adoption of Tooling and Best Practices

- Today's environment drives the need for increasing efficiencies, to drive higher value at lower costs
 - Innovative adoption of tooling and best practices can yield
 - Automation opportunities
 - Improved productivity and efficiencies
 - Repeatability of results
 - Rapid application of sophisticated solutions to common problems
- Improved quality

Tooling and Best Practices: Examples

- Common tools include
 - Static code analysis
 - Automated test execution
 - Automated test system provisioning
 - Defect classification
 - Memory leak and memory corruption detection
 - Code coverage
- Common best practices include
 - Iterative validation and refinement of goals with clients throughout development cycle
 - Peer reviews of software engineering deliverables
 - Change control

Broad Adoption of Best Practices: Challenges

Scenario 1:

- When asking a team to consider making an investment of time to adopt a best practice, the response is “no time”.
 - “You want to teach me how to fish. Right now, I’m starving, and I simply need to be fed. “
- Response: step-wise enablement of teams may yield the best results
 - “Give a man a fish; then teach the man to fish”
 - A collaborative technique

Example of step-wise enablement

- Step 1: Demonstrate value of static code analysis
 - Run the tool for a specific team’s code
 - Find/fix some really useful problems immediately
 - Build the team’s confidence in static analysis tooling
- Step 2: Enable team to execute themselves
 - Assist the teams in instrumenting their build
 - Run reports periodically from production build
 - Track progress (complaint density vs. code volume over time)
- Step 3: Extra credit: enable developers
 - Put “plug-in” capability of tools on developer desktops
 - Enables developers to find/fix problems within their own code
- Teams graduate from “prove it to me” to “help me integrate this”

Broad Adoption of Best Practices: Challenges

Scenario 2:

- A particular practice or tool is used once during each development life cycle, yet requires a certain level of expertise to use successfully
 - Lack of subject matter expertise results in not using the tooling, or using with poor results
- Response: Subject Matter Experts are made available as consultants to the teams at the time the expertise is required.
 - Another example of collaboration

Broad Adoption of Best Practices: Prerequisites

- In order for the broadest adoption to occur across an organization,
 - It must have been proven to work for at least one other team, with demonstrated results
 - The adopting teams need to understand the value to be gained, and the investment required
- Alternatively:
 - A given team may choose to accept some risk by participating in a “pilot” of a new practice
 - Again, the team must understand the value, and the risks
 - The team should be willing to become a “reference”

4. A Community of Collaboration

- Maximize “cross-pollination” of skills and knowledge
 - Peer-to-peer
 - Team to team
 - Expert-to-novice
- Promote innovation
- Share best practices
 - Tools, techniques, processes, procedures
- Share experiences of successes and failures
 - Where the value of sharing failures is in the lessons learned

A Community of Collaboration: Cross-organizational teams

- Build a living community for collaboration on topics regarding a specific discipline
 - Active cross-team participation facilitates learning and adoption across large organizations
 - A wide variety of vehicles can be used
 - Meetings, conference calls or web meetings
 - On-line forums, web sites
 - Brings together individuals with common skills or interests who might otherwise not interact on a regular basis

A Community of Collaboration: Assessments

- Software Assessments
 - The purpose is diagnostic in nature
 - Not an end, but a means to an end
 - Identifies best practices as well as flaws or serious problems
 - Facilitates identification of remedies
- Target specific areas for improvement
 - Tooling, processes, practices
- Continue a program of follow-ups
 - To review progress on targeted improvements
 - To facilitate adoption of particular recommendations
- Collect additional best practices for use in other parts of the organization
- Study of aggregate assessment data can facilitate broader organizational actions

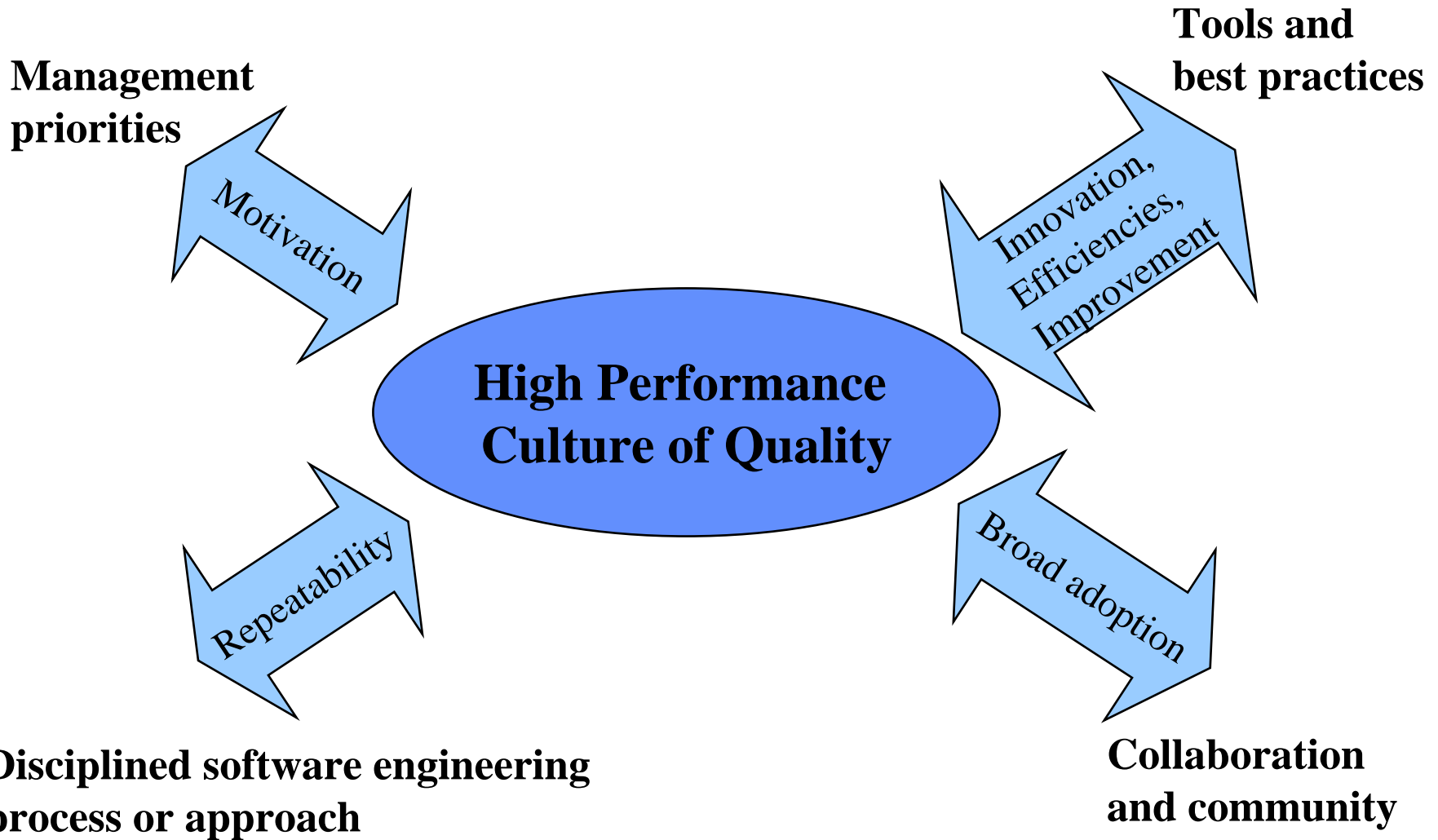
A Community of Collaboration: Enablement

- Focused enablement teams collaborating with project teams
 - Subject matter experts acting as hands-on consultants and enablers
 - Demonstrate the technique / tool / approach; highlight the value
 - Enable simple first steps to show benefit
 - First objective: Make them believers
 - Next objective: Help them integrate into ongoing practice
 - Final objective: Recruit new practitioners to become references, additional enablers for other teams
 - End result is teams motivated and enabled to execute best practices as part of on-going business

Bringing It All Together

1. Clear leadership priorities
 - Common vision: passion for client success
 - Promotes individual and team response
2. Disciplined execution of process or approach
 - Consistent, repeatable approach
3. Broad adoption of tooling and best practices
 - Enables improved productivity, quality.
4. A community of collaboration
 - To share successes, failures across teams
 - To provide focused enablement of individual teams with subject matter expertise

Bringing It All Together



Thank You!

Helen McKinstry
helenmck@us.ibm.com