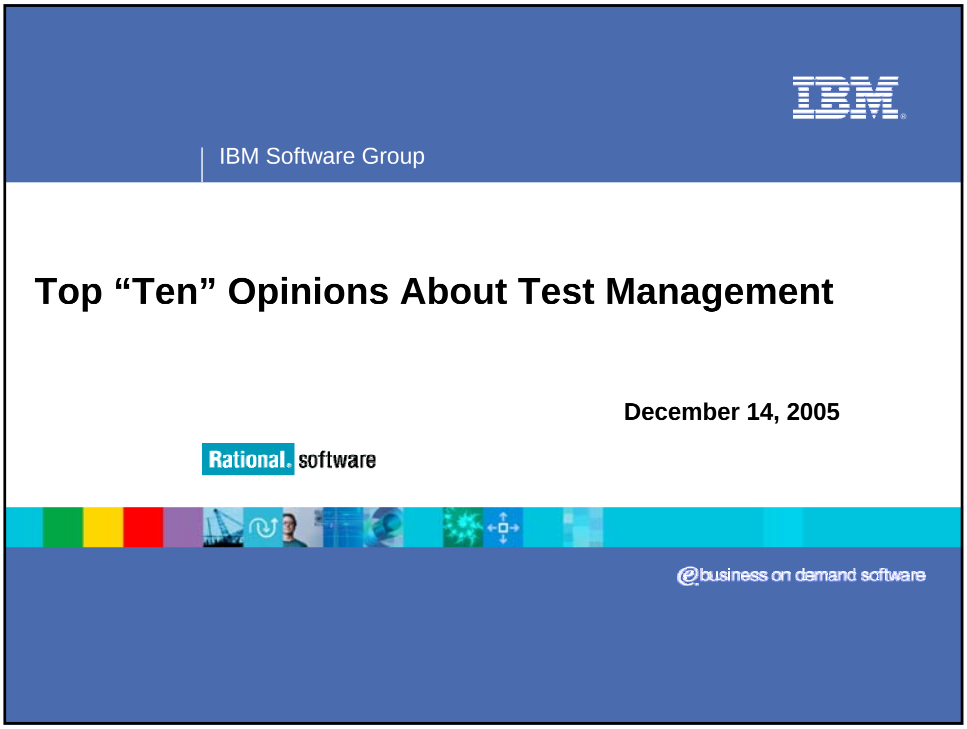


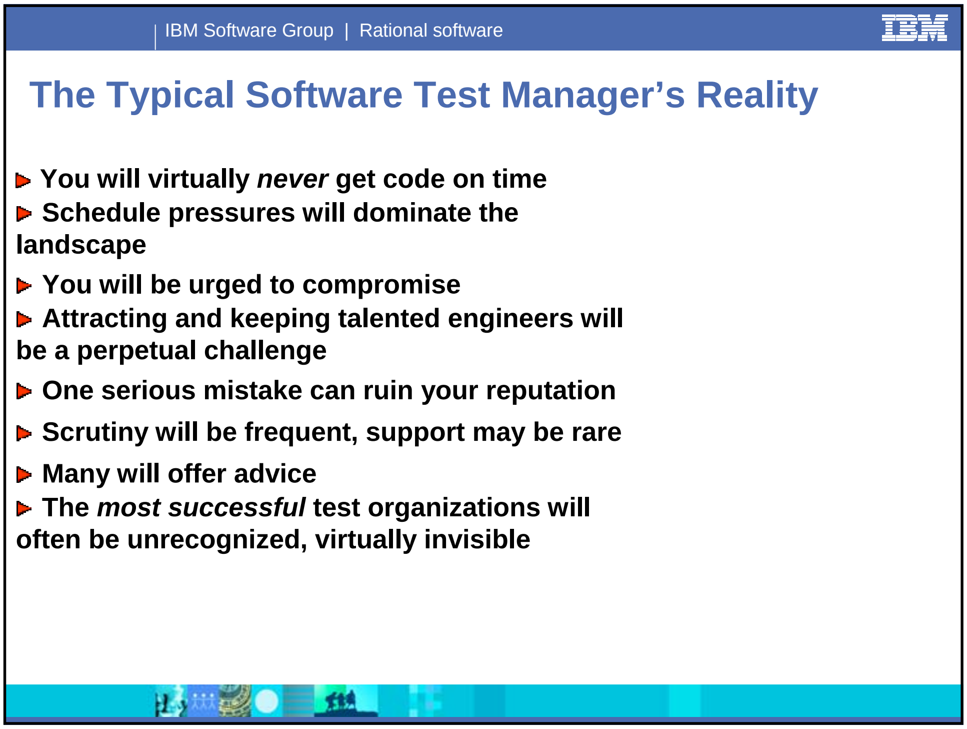
IBM Software Group

Top “Ten” Opinions About Test Management

December 14, 2005

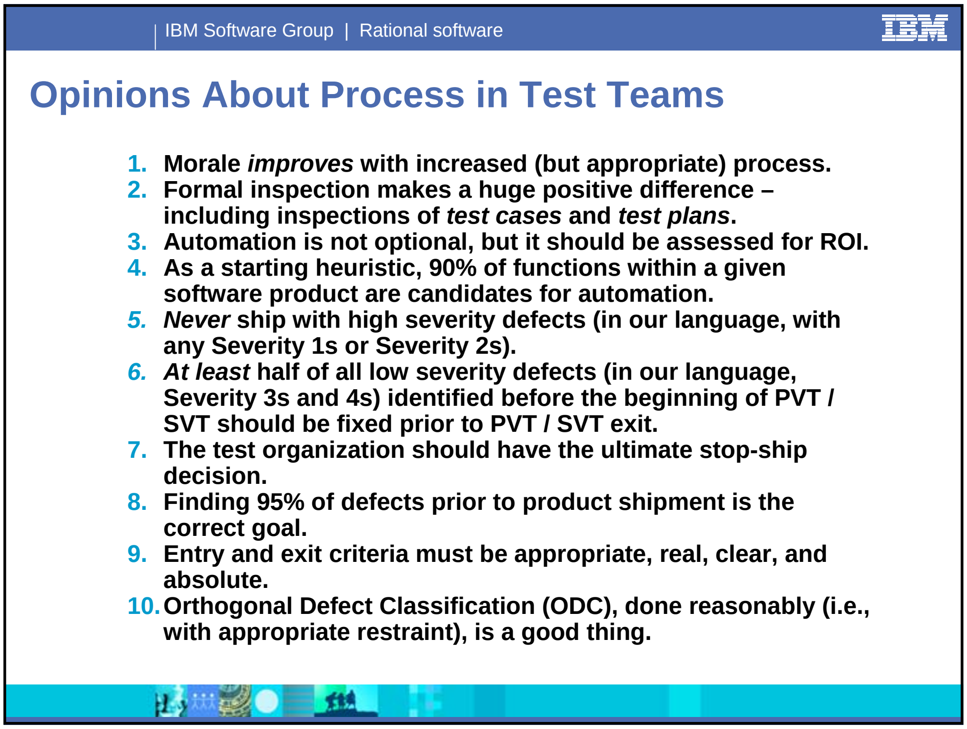
Rational. software

@business on demand software



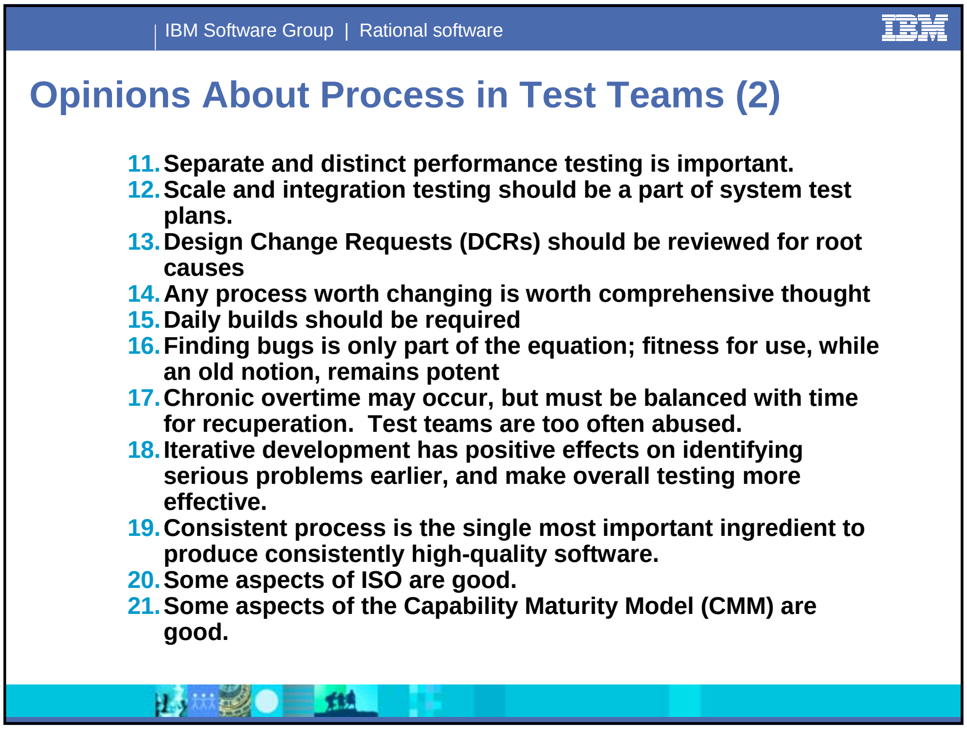
The Typical Software Test Manager’s Reality

- You will virtually *never* get code on time
- Schedule pressures will dominate the landscape
- You will be urged to compromise
- Attracting and keeping talented engineers will be a perpetual challenge
- One serious mistake can ruin your reputation
- Scrutiny will be frequent, support may be rare
- Many will offer advice
- The *most successful* test organizations will often be unrecognized, virtually invisible



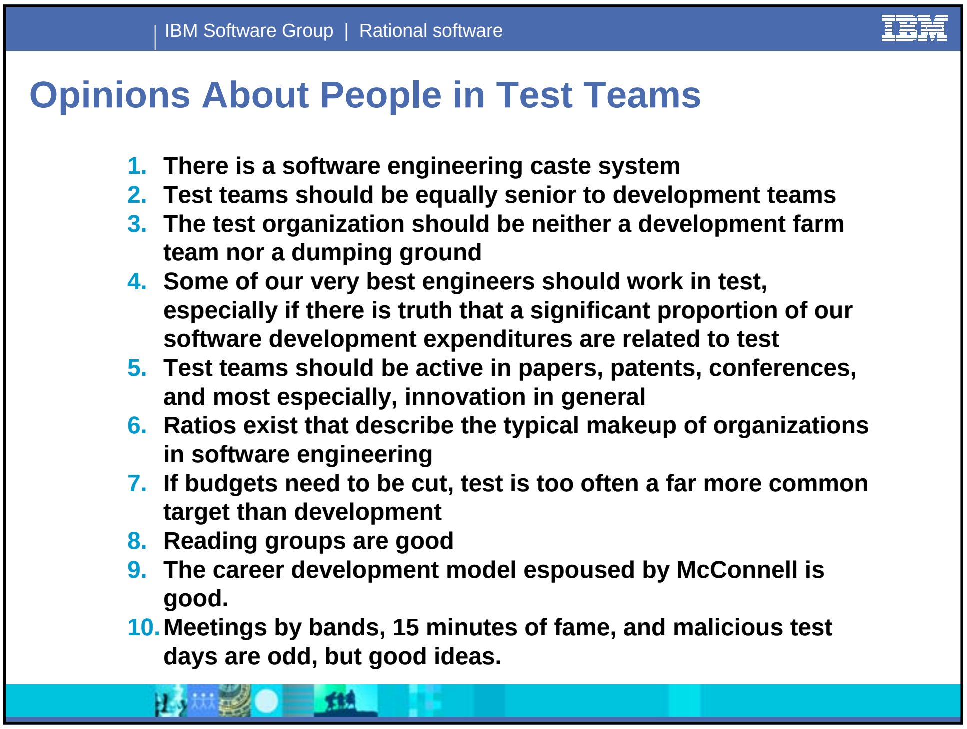
Opinions About Process in Test Teams

1. Morale *improves* with increased (but appropriate) process.
2. Formal inspection makes a huge positive difference – including inspections of *test cases* and *test plans*.
3. Automation is not optional, but it should be assessed for ROI.
4. As a starting heuristic, 90% of functions within a given software product are candidates for automation.
5. *Never* ship with high severity defects (in our language, with any Severity 1s or Severity 2s).
6. *At least* half of all low severity defects (in our language, Severity 3s and 4s) identified before the beginning of PVT / SVT should be fixed prior to PVT / SVT exit.
7. The test organization should have the ultimate stop-ship decision.
8. Finding 95% of defects prior to product shipment is the correct goal.
9. Entry and exit criteria must be appropriate, real, clear, and absolute.
10. Orthogonal Defect Classification (ODC), done reasonably (i.e., with appropriate restraint), is a good thing.



Opinions About Process in Test Teams (2)

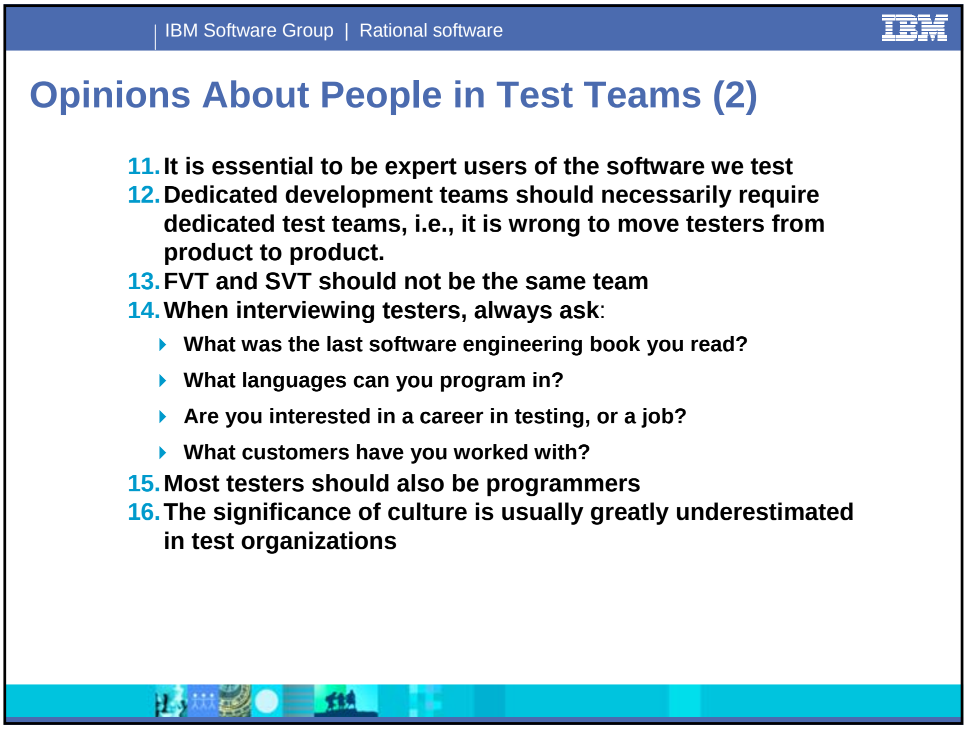
11. Separate and distinct performance testing is important.
12. Scale and integration testing should be a part of system test plans.
13. Design Change Requests (DCRs) should be reviewed for root causes
14. Any process worth changing is worth comprehensive thought
15. Daily builds should be required
16. Finding bugs is only part of the equation; fitness for use, while an old notion, remains potent
17. Chronic overtime may occur, but must be balanced with time for recuperation. Test teams are too often abused.
18. Iterative development has positive effects on identifying serious problems earlier, and make overall testing more effective.
19. Consistent process is the single most important ingredient to produce consistently high-quality software.
20. Some aspects of ISO are good.
21. Some aspects of the Capability Maturity Model (CMM) are good.



IBM Software Group | Rational software

Opinions About People in Test Teams

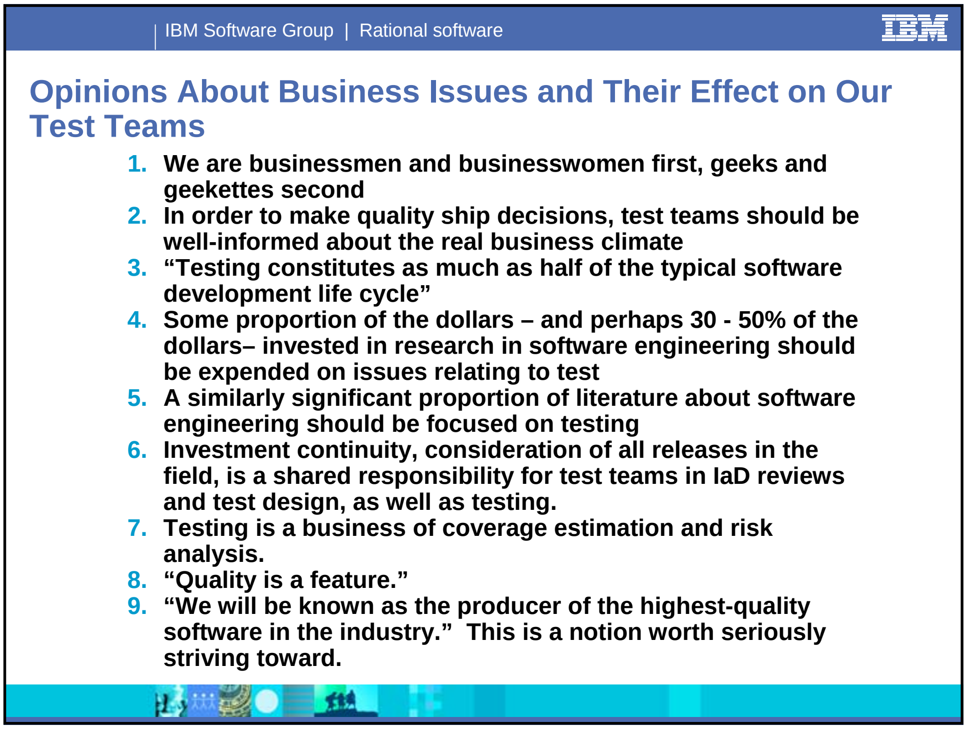
1. There is a software engineering caste system
2. Test teams should be equally senior to development teams
3. The test organization should be neither a development farm team nor a dumping ground
4. Some of our very best engineers should work in test, especially if there is truth that a significant proportion of our software development expenditures are related to test
5. Test teams should be active in papers, patents, conferences, and most especially, innovation in general
6. Ratios exist that describe the typical makeup of organizations in software engineering
7. If budgets need to be cut, test is too often a far more common target than development
8. Reading groups are good
9. The career development model espoused by McConnell is good.
10. Meetings by bands, 15 minutes of fame, and malicious test days are odd, but good ideas.



IBM Software Group | Rational software

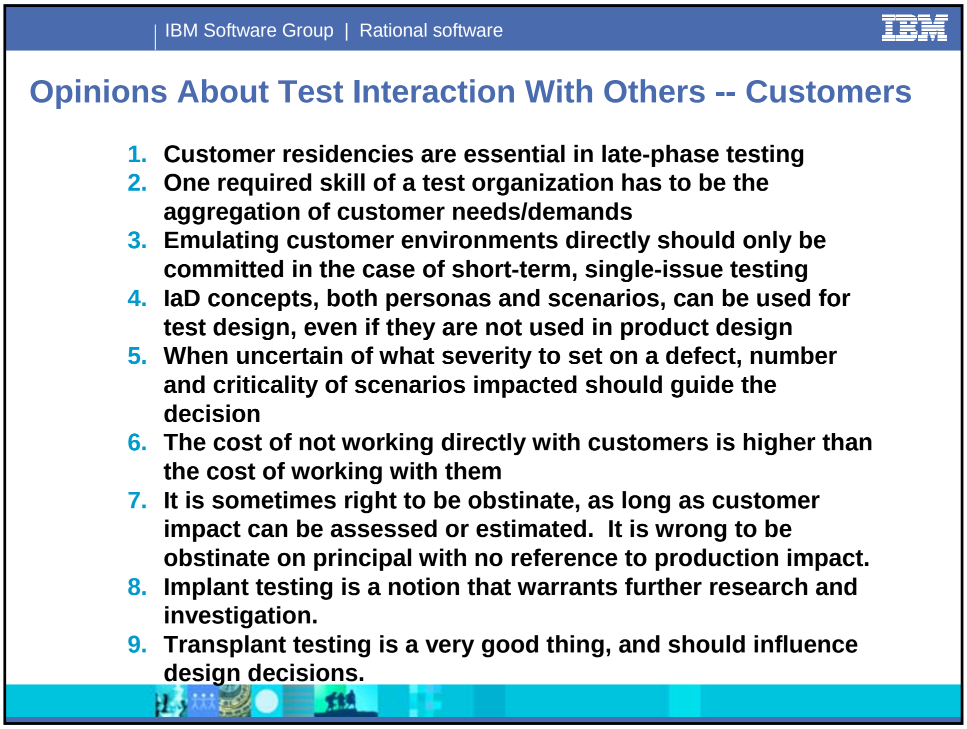
Opinions About People in Test Teams (2)

11. It is essential to be expert users of the software we test
12. Dedicated development teams should necessarily require dedicated test teams, i.e., it is wrong to move testers from product to product.
13. FVT and SVT should not be the same team
14. When interviewing testers, always ask:
    - What was the last software engineering book you read?
    - What languages can you program in?
    - Are you interested in a career in testing, or a job?
    - What customers have you worked with?
15. Most testers should also be programmers
16. The significance of culture is usually greatly underestimated in test organizations



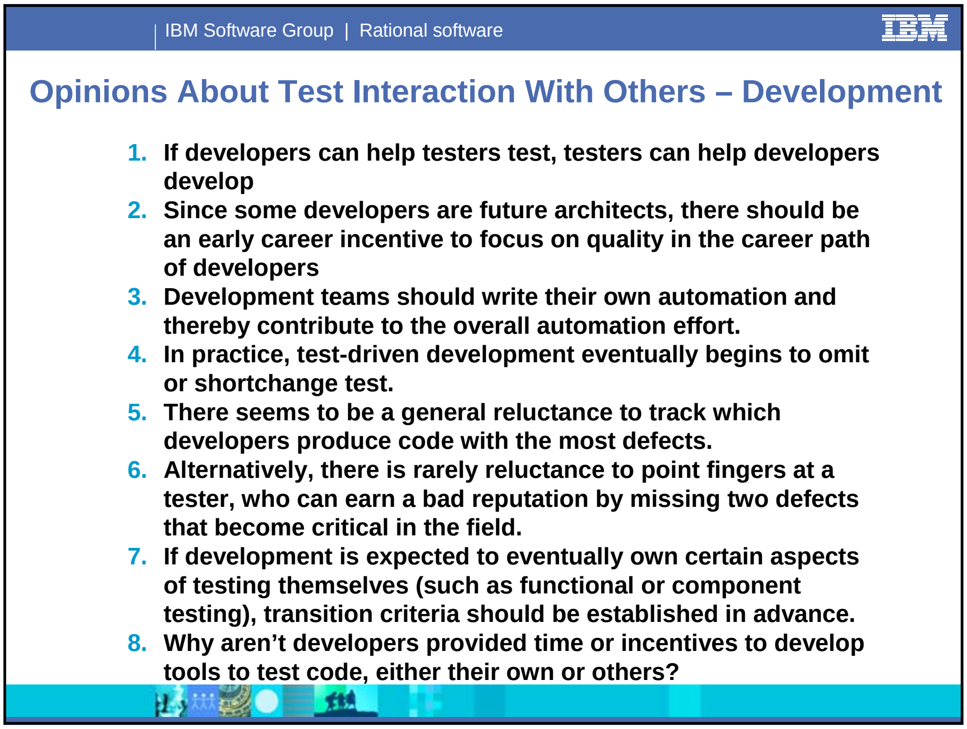
Opinions About Business Issues and Their Effect on Our Test Teams

1. **We are businessmen and businesswomen first, geeks and geekettes second**
2. **In order to make quality ship decisions, test teams should be well-informed about the real business climate**
3. **“Testing constitutes as much as half of the typical software development life cycle”**
4. **Some proportion of the dollars – and perhaps 30 - 50% of the dollars– invested in research in software engineering should be expended on issues relating to test**
5. **A similarly significant proportion of literature about software engineering should be focused on testing**
6. **Investment continuity, consideration of all releases in the field, is a shared responsibility for test teams in IaD reviews and test design, as well as testing.**
7. **Testing is a business of coverage estimation and risk analysis.**
8. **“Quality is a feature.”**
9. **“We will be known as the producer of the highest-quality software in the industry.” This is a notion worth seriously striving toward.**



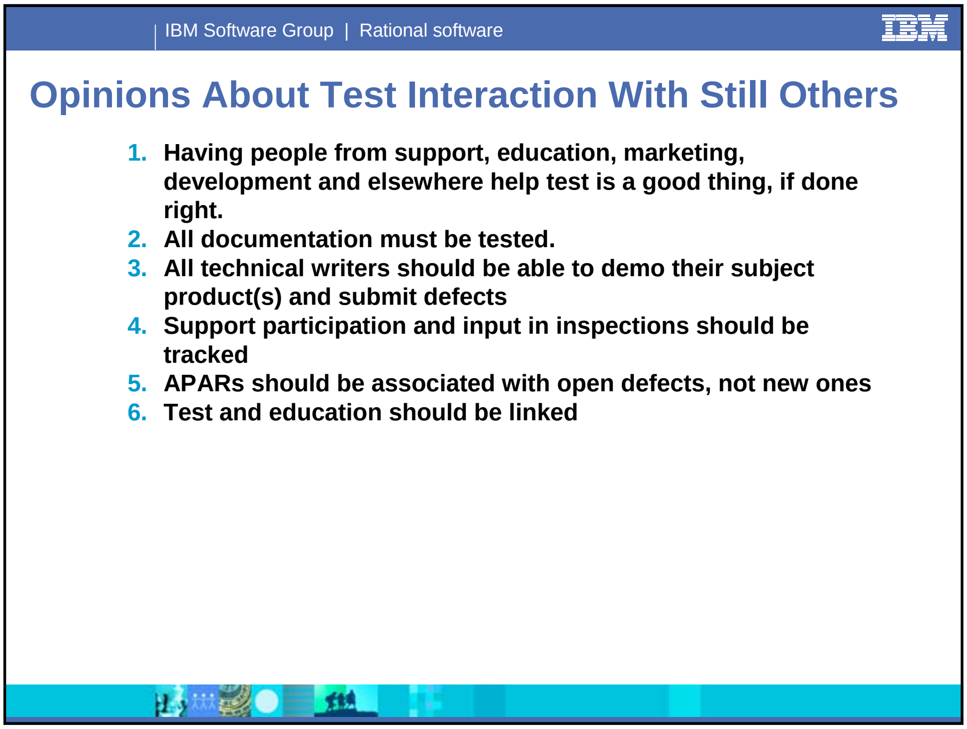
Opinions About Test Interaction With Others -- Customers

1. **Customer residencies are essential in late-phase testing**
2. **One required skill of a test organization has to be the aggregation of customer needs/demands**
3. **Emulating customer environments directly should only be committed in the case of short-term, single-issue testing**
4. **IaD concepts, both personas and scenarios, can be used for test design, even if they are not used in product design**
5. **When uncertain of what severity to set on a defect, number and criticality of scenarios impacted should guide the decision**
6. **The cost of not working directly with customers is higher than the cost of working with them**
7. **It is sometimes right to be obstinate, as long as customer impact can be assessed or estimated. It is wrong to be obstinate on principal with no reference to production impact.**
8. **Implant testing is a notion that warrants further research and investigation.**
9. **Transplant testing is a very good thing, and should influence design decisions.**



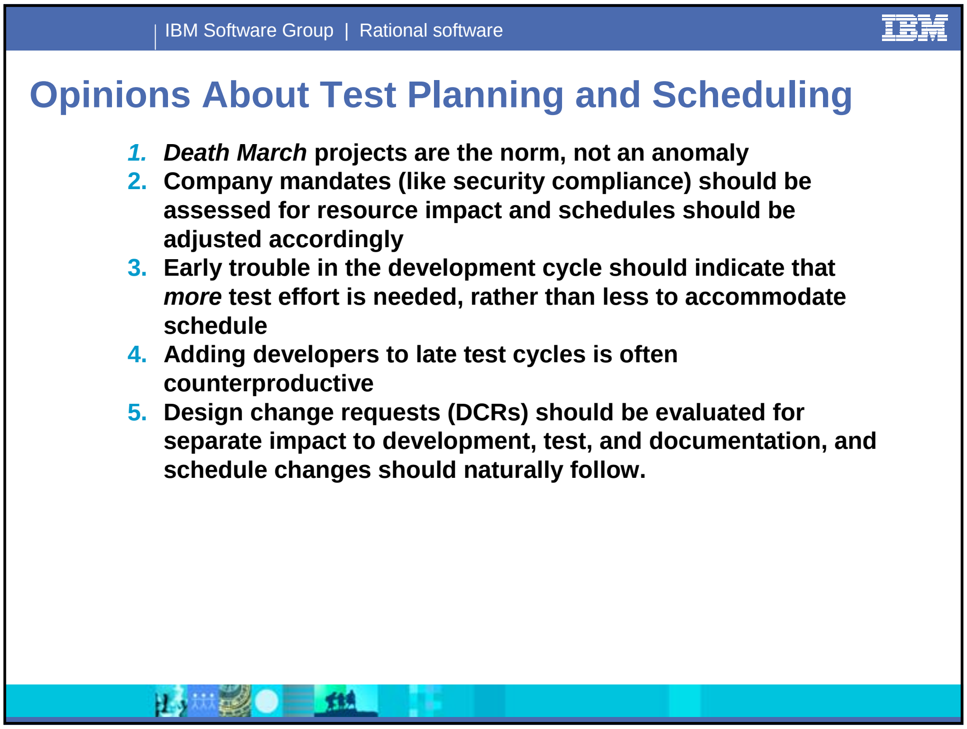
Opinions About Test Interaction With Others – Development

1. If developers can help testers test, testers can help developers develop
2. Since some developers are future architects, there should be an early career incentive to focus on quality in the career path of developers
3. Development teams should write their own automation and thereby contribute to the overall automation effort.
4. In practice, test-driven development eventually begins to omit or shortchange test.
5. There seems to be a general reluctance to track which developers produce code with the most defects.
6. Alternatively, there is rarely reluctance to point fingers at a tester, who can earn a bad reputation by missing two defects that become critical in the field.
7. If development is expected to eventually own certain aspects of testing themselves (such as functional or component testing), transition criteria should be established in advance.
8. Why aren't developers provided time or incentives to develop tools to test code, either their own or others?



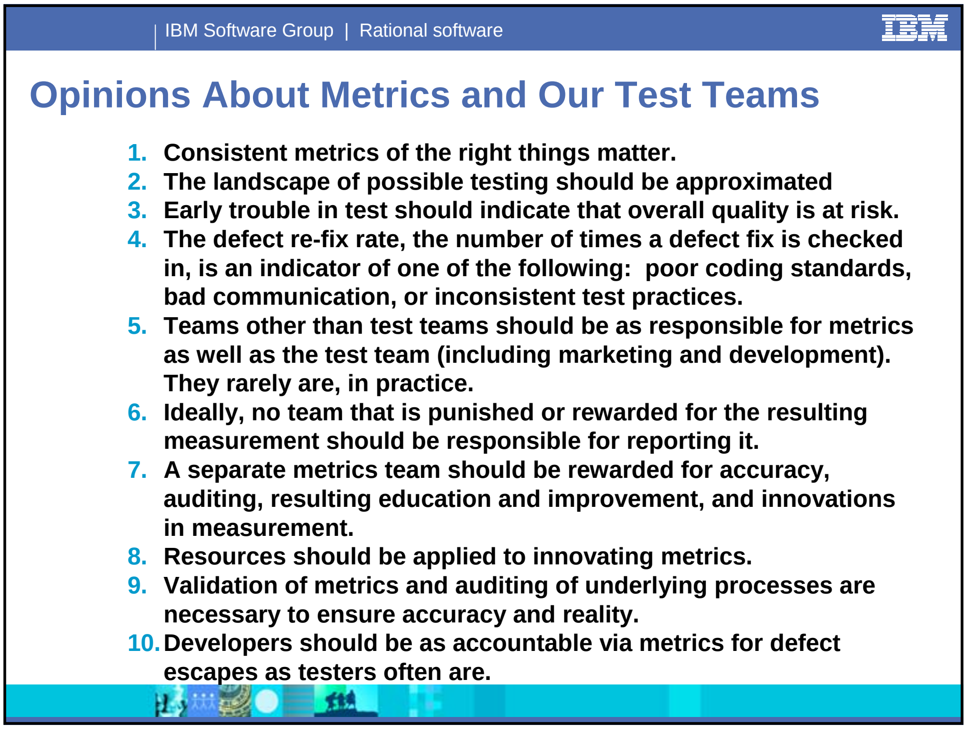
Opinions About Test Interaction With Still Others

1. Having people from support, education, marketing, development and elsewhere help test is a good thing, if done right.
2. All documentation must be tested.
3. All technical writers should be able to demo their subject product(s) and submit defects
4. Support participation and input in inspections should be tracked
5. APARs should be associated with open defects, not new ones
6. Test and education should be linked



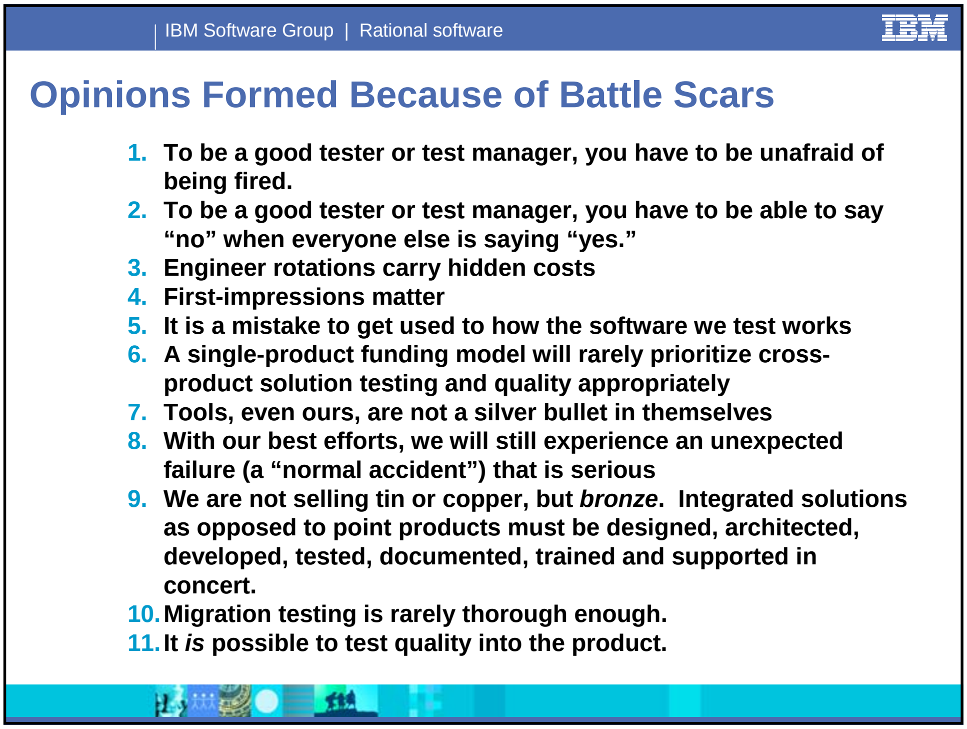
Opinions About Test Planning and Scheduling

1. ***Death March*** **projects are the norm, not an anomaly**
2. **Company mandates (like security compliance) should be assessed for resource impact and schedules should be adjusted accordingly**
3. **Early trouble in the development cycle should indicate that *more* test effort is needed, rather than less to accommodate schedule**
4. **Adding developers to late test cycles is often counterproductive**
5. **Design change requests (DCRs) should be evaluated for separate impact to development, test, and documentation, and schedule changes should naturally follow.**



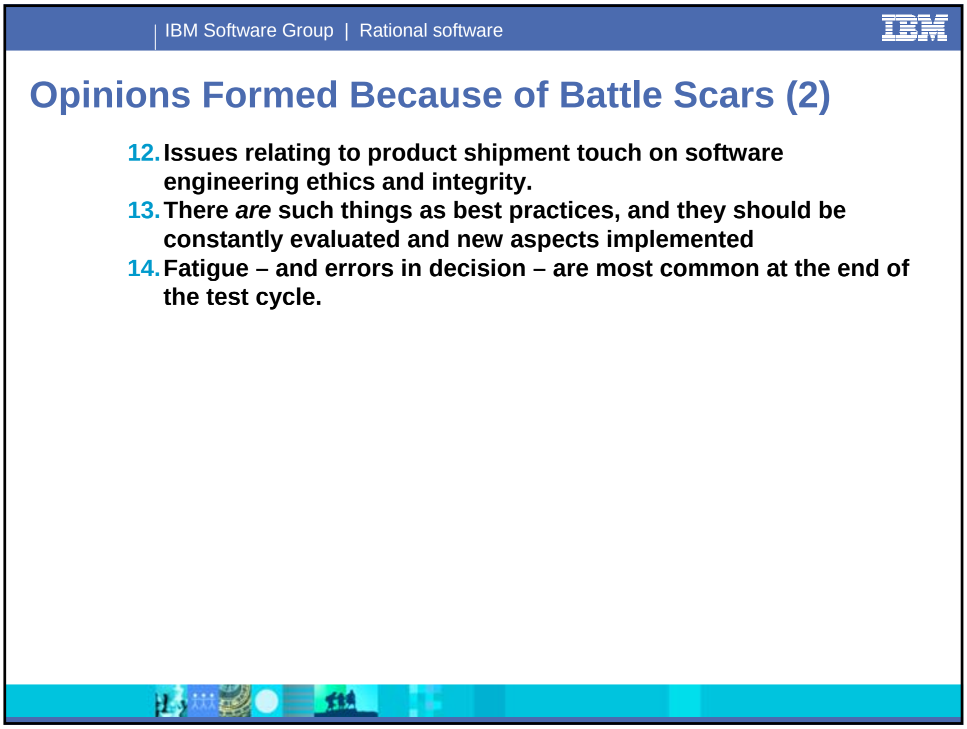
Opinions About Metrics and Our Test Teams

1. **Consistent metrics of the right things matter.**
2. **The landscape of possible testing should be approximated**
3. **Early trouble in test should indicate that overall quality is at risk.**
4. **The defect re-fix rate, the number of times a defect fix is checked in, is an indicator of one of the following: poor coding standards, bad communication, or inconsistent test practices.**
5. **Teams other than test teams should be as responsible for metrics as well as the test team (including marketing and development). They rarely are, in practice.**
6. **Ideally, no team that is punished or rewarded for the resulting measurement should be responsible for reporting it.**
7. **A separate metrics team should be rewarded for accuracy, auditing, resulting education and improvement, and innovations in measurement.**
8. **Resources should be applied to innovating metrics.**
9. **Validation of metrics and auditing of underlying processes are necessary to ensure accuracy and reality.**
10. **Developers should be as accountable via metrics for defect escapes as testers often are.**



Opinions Formed Because of Battle Scars

1. **To be a good tester or test manager, you have to be unafraid of being fired.**
2. **To be a good tester or test manager, you have to be able to say “no” when everyone else is saying “yes.”**
3. **Engineer rotations carry hidden costs**
4. **First-impressions matter**
5. **It is a mistake to get used to how the software we test works**
6. **A single-product funding model will rarely prioritize cross-product solution testing and quality appropriately**
7. **Tools, even ours, are not a silver bullet in themselves**
8. **With our best efforts, we will still experience an unexpected failure (a “normal accident”) that is serious**
9. **We are not selling tin or copper, but *bronze*. Integrated solutions as opposed to point products must be designed, architected, developed, tested, documented, trained and supported in concert.**
10. **Migration testing is rarely thorough enough.**
11. **It *is* possible to test quality into the product.**



Opinions Formed Because of Battle Scars (2)

12. **Issues relating to product shipment touch on software engineering ethics and integrity.**
13. **There *are* such things as best practices, and they should be constantly evaluated and new aspects implemented**
14. **Fatigue – and errors in decision – are most common at the end of the test cycle.**