

Understanding Test Management *And the Relationship with Quality Management*



RBCS
TIME TESTED.
TESTING IMPROVED.
www.RBCS-US.com



Test Management and Quality Management

- ❖ Are test management and quality management the same or different?
- ❖ How does testing relate to quality?
- ❖ Confusion abounds, and creates problems for test teams
- ❖ Let's look at five key lessons in clearing the confusion and resolving the problems



Key Lessons in Testing and Quality

- ❖ Testing is a skilled activity
- ❖ Testing has a quantifiable defect detection effectiveness
- ❖ Schedule, budget, and quality trade-offs come later in the lifecycle
- ❖ Understand bugs in the wild and bugs in captivity
- ❖ Integrate testing into quality management



Testing Requires Skills

- ❖ Sign of confusion: “ Anyone can test. How hard can it be? Just make sure it works before we ship!”
- ❖ Amateur testing (i.e., testing done by people other than skilled test professionals) is less effective and efficient
- ❖ Testing involves a mix of technical, application, and test-specific skills
- ❖ Testing should be managed by experienced test managers

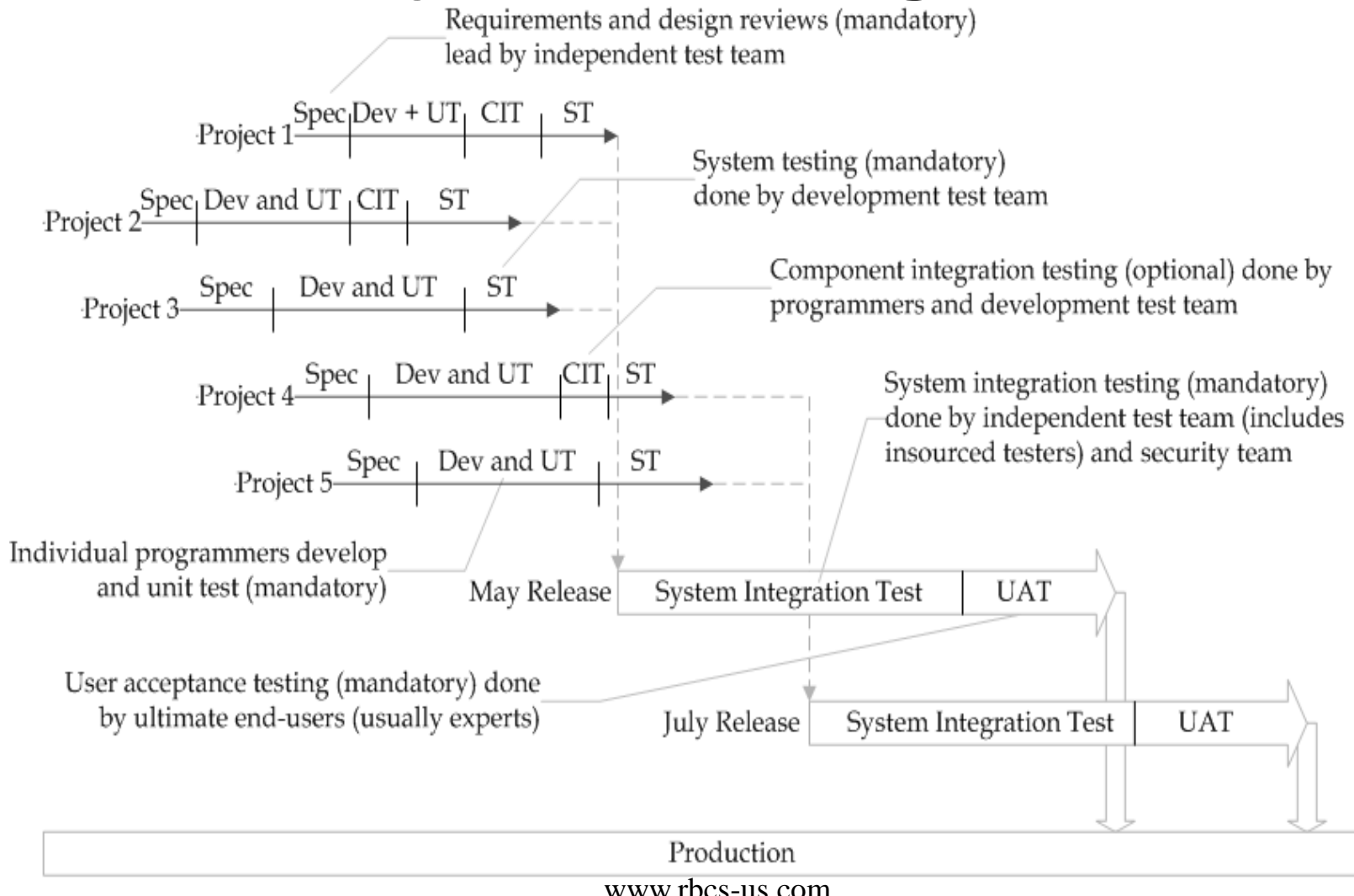


Case Study: Testing Skills Pay Off

- ❖ One client used former support staff for their testing
- ❖ These people had no testing skill or experience
- ❖ After their testers got ISTQB Foundation certified, one tester found a major efficiency improvement
- ❖ By using equivalence partitioning to eliminate redundant regression tests, over 25% of the test team's time was freed up



Case Study: Skills throughout SDLC



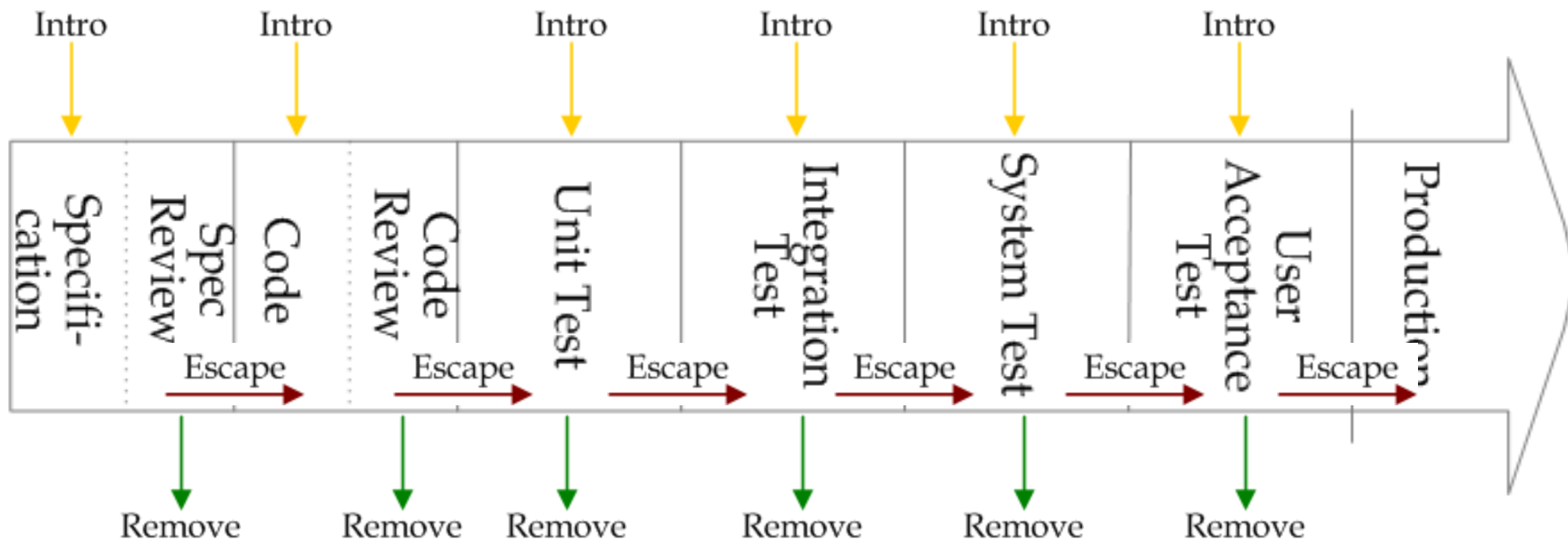


Measuring Defect Detection Effectiveness

- ❖ Sign of confusion: “We have bugs in Production! The testing failed!”
- ❖ Testing has a quantifiable defect detection effectiveness (DDE)
- ❖ DDE is the number of defects detected by any test or quality activity, as a percentage of the defects present
- ❖ $DDE = \text{defects detected} / \text{defects present}$



Visualizing Defect Flow



Defect detection effectiveness (DDE) = defects detected/defects present
Defect removal effectiveness (DRE) = defects removed/defects present
Defects present = defects introduced (this phase) + defects escaped (previous phase)
Defects removed \leq defects detected
For system test (ST), DDE = ST defects/(ST defects + UAT defects + prod defects)



Case Study: Observed Industry Capability

Activity	DDE	Activity	DDE
Requirements Review	65%	Developer Testing (Best)	50%
Design Review	65%	Professional Testing (Avg)	85%
Code Review	60%	Professional Testing (Good)	90%
Developer Testing (Poor)	10%	Professional Testing (Best)	99%
Developer Testing (Avg)	25%		

Industry figures taken from Capers Jones, *Estimating Software Costs, 3e*, except for the professional testing figures, which derive from RBCS assessments of clients. Professional testing can only achieve high DDE when it's part of an integrated testing and quality strategy.



Trade-offs in the Lifecycle

- ❖ Sign of confusion: “We don’t have the time/money to deliver high quality, so we have to settle for good enough.”
- ❖ In fact, the schedule, budget, and quality trade-offs arise in the later phases of the lifecycle
- ❖ Because the cost of removing bugs escalates with each phase, early defect removal reduces overall schedule, budget, and quality risk



Case Study: Overwhelmed at the End

- ✦ We assessed one client which relied heavily on bug removal in the final phases of testing
- ✦ We found that they had \$100M to \$250M in avoidable costs of failure
- ✦ Defects that would cost \$37 to remove in a requirements review were being removed for \$3,700 each in QA testing
- ✦ Since that extra \$3,600 represents effort, the trade-off between quality, schedule and budget was **created** by high escape rates earlier in the lifecycle

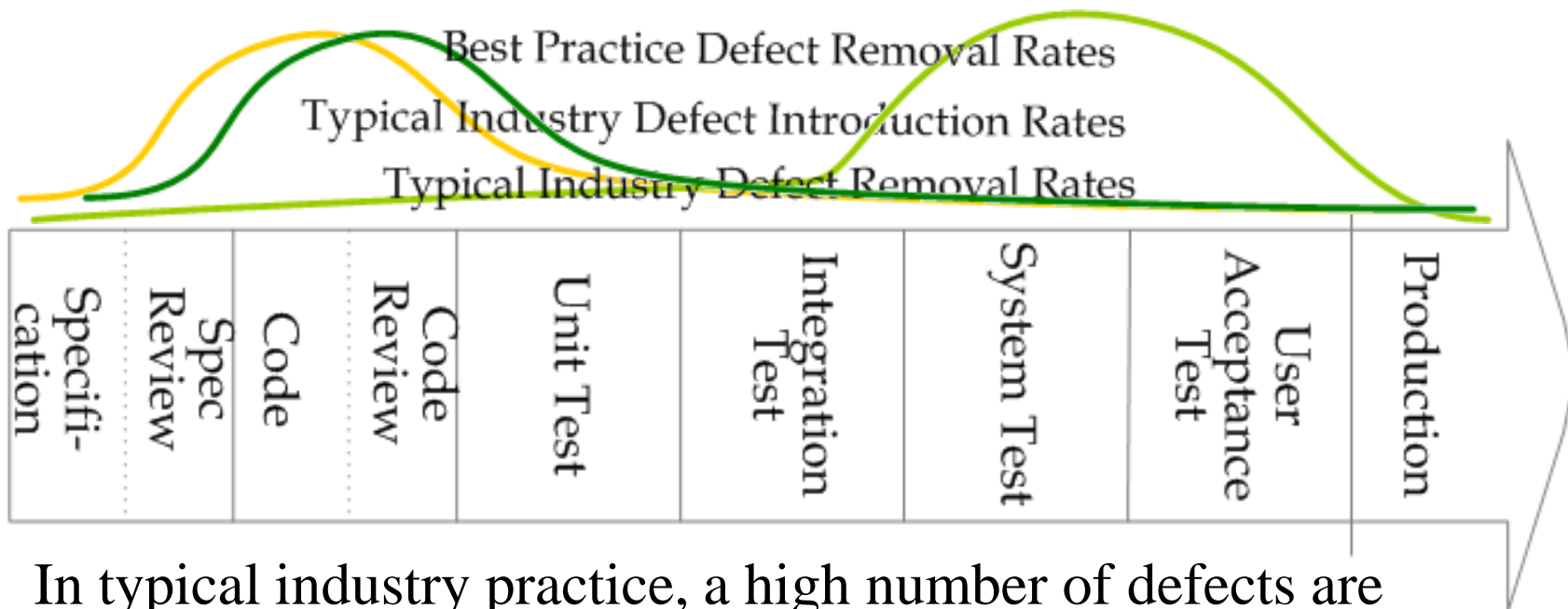


Bugs, Wild and Captive

- ❖ Sign of confusion: “Testing is about breaking software.”
- ❖ Actually, the software comes to testing with most of the bugs it will have
- ❖ The illusion of testing breaking software arises because so many bugs are captured in testing
- ❖ Bugs are introduced throughout lifecycle, but primarily in the early activities, while bugs are often removed primarily late in the lifecycle
- ❖ A defect removal model, such as the one shown earlier, helps people understand the true nature of bugs, both in the wild and in captivity



Case Study: Typical vs. Best Practices



In typical industry practice, a high number of defects are introduced early in the lifecycle, then removed (at a much higher cost) later in the lifecycle. Best practices focus on early defect removal, with later test phases focusing on building confidence and reducing risk.



Integrate Testing into Quality Management

- ❖ Sign of confusion: “We miss important areas in our testing, and we don’t have a clear view of the level of quality before we release.”
- ❖ All testing activities in the lifecycle should have well-defined coverage goals
- ❖ All test results should consolidate into a single view on the status of the product



Case Study: Holistic Approach

Test Level	Owner	Objectives
Unit	Development	<ul style="list-style-type: none">•Early detection of bugs in code units•Reduce risk of unit failures in production•Unit testing run before CIT•Unit tests results documented in the feature turnover
CIT (Comp Int Test)	Development	<ul style="list-style-type: none">•Early detection of bugs in unit interfaces•Reduce risk of dataflow and workflow failures in production•Ensure development ownership for delivering quality features
CIT	QA	<ul style="list-style-type: none">•Early QA validation of completed features
System/ SIT	QA	<ul style="list-style-type: none">•Detect bugs, reduce risk, build confidence in use cases and end-to-end scenarios•Detect bugs, reduce risk, build confidence in user workflows
Beta	Customer	<ul style="list-style-type: none">•Detect bugs related to customer deployment•Reduce risk of failing customer business requirements•Demonstrate readiness for deployment

www.rbc-us.com



Conclusion

- ❖ People confuse test management and quality management, which creates problems
- ❖ Key lessons to resolve the confusion and problems:
 - ❖ Testing is a skilled activity
 - ❖ Testing has a quantifiable defect detection effectiveness
 - ❖ Trade-offs come later in the lifecycle
 - ❖ Bugs in the wild and bugs in captivity
 - ❖ Integrate testing into quality management
- ❖ By applying these key lessons to the software engineering process, test and quality professionals can deliver quality software, thoroughly tested



...*Contact RBCS*

For over a dozen years, RBCS has delivered services in consulting, outsourcing and training for software and hardware testing. Employing the industry's most experienced and recognized consultants, RBCS conducts product testing, builds and improves testing groups and hires testing staff for hundreds of clients worldwide. Ranging from Fortune 20 companies to start-ups, RBCS clients save time and money through improved product development, decreased tech support calls, improved corporate reputation and more. To learn more about RBCS, visit www.rbc-us.com.

Address: RBCS, Inc.
31520 Beck Road
Bulverde, TX 78163-3911
USA

Phone: +1 (830) 438-4830

Fax: +1 (830) 438-4831

E-mail: info@rbc-us.com

Web: www.rbc-us.com