BIO
PRESENTATION
PAPER

T5

November 7, 2002
10:30 AM

CREATING FIVE-STAR TEST METRICS ON A ONE-STAR BUDGET

Rick Tennant
SBC Communications, Inc.

International Conference On
Software Testing Analysis & Review
November 4-8, 2002
Anaheim, CA USA

Rick Tennant

Rick Tennant is a Senior Technical Director at SBC Communications Inc., overseeing the quality control activities for the DSL ordering applications. Rick joined SBC in 1989, after 12 years at McDonnell Douglas. He is experienced in software testing (most technologies), test automation, test management, test data and environments, test strategies, and testing processes. In 1997, Rick coordinated several enterprise wide “end-to-end” tests following the merger with Pacific Telesis as the corporation opened up their systems to external usage. In September 2000, he moved to his current position. Under his direction, the testing organization has been instrumental in the rollout of DSL ordering systems in the company’s 13-state territory.

Five Star Testing Reports on a One Star Budget

Rick Tennant

SBC Communications, Inc.

Presentation Outline

- Background information
- Strategy in creating useful, cost effective test reports
- Examples
- Questions?

SBC Communications

- **Second largest telecommunications provider in the US; local service provider in 13 states**
- **Over 58 million access lines**
- **Largest provider of DSL service in the US, 1.7 million lines**
- **190,000 employees**

SBC Information Technology

- One of the largest IT shops in the US
- Over 15,000 employees across 13 states
- 2,206 applications supported
- Over 35 million customer bills produced each month
- 130 million orders taken annually from our retail and wholesale customers
- SEI CMM process improvement model

DSL Ordering Applications

- DSL application suite that includes online, batch, XML interfaces - along with several tools
- Applications are used across 13 state territory
- 160 member project team
- Client/Server model, using Powerbuilder, JAVA, COBOL, C++, Websphere
- Quarterly release schedule

DSL Ordering Quality Control

- 23 member team - 15 in St, Louis, MO and 8 in San Ramon, CA
- 3 teams - Online, Batch/XML, and QC Coordination & Support
- Automated regression testing using Mercury Interactive's WinRunner

Our Issues

- Current reporting and metrics were inconsistent and did not benefit decision making on project
- Had to create without adding additional people
- Limited resources - could not invest in new processes or tools to generate source data for metrics
- Source data must be consistent
- Very diverse audience - business, technical, management, and geographically dispersed
- Needed a report that would give credibility to and support verbal readouts

Checklist for designing test reports

- Identify the target audience and how this information will be used
- Design reports that meet the informational and decision making needs of that audience
- Make reports concise and easy to understand
- Identify your data sources
- Make the reporting cost effective - reporting should be a natural by-product of your testing process!

Identify your target audience

- Senior Management
- Middle Management
- Clients
- End users
- Project Managers
- Interfacing projects
- Quality Control
- Development
- Change Managers
- Requirement Analysts
- Operations & Support
- SQA (process monitors)

How will this information be used?

- Daily release status
- Scheduling decisions
- Deployment decisions
- Product quality
- Product stability
- Business decisions
- Project decisions
- Turnover and status for UAT
- QC resource allocations
- Tester performance
- Key learnings
- Process improvements

Report Design

Answer the questions that drive the business decisions and informational needs for your target audience

- How complete is our testing (coverage)?
- Is our testing on schedule?
- How stable is the product?
- How severe are the defects?
- Do our resources or assignments need to be adjusted?
- What is the root cause of our defects?
- How are the testers performing?

Design reports that are concise and easy to understand

- Answer multiple questions in a single illustration, when possible
- Put project wide release status up front, testing specific information last - get the most important information on the first page
- Illustrate your data and title it so it will not be subject to interpretation
- Remember your audience

Identify your data sources

- **Defect data fields**
  - *Originator, date, time*
  - *Assigned to, date, time*
  - *Application name*
  - *Release number*
  - *Test phase*
  - *Issue/Defect status*
  - *Date/time problem fixed*
  - *Date/time fix validated*
  - *Cause code*
  - *Priority*
- **Test planning data**
  - *Testable items (requirements)*

Being cost effective

- Find ways to use your current information to produce meaningful metrics
- Automate your report process
- Don’t produce metrics that are not going to be used

Benefits

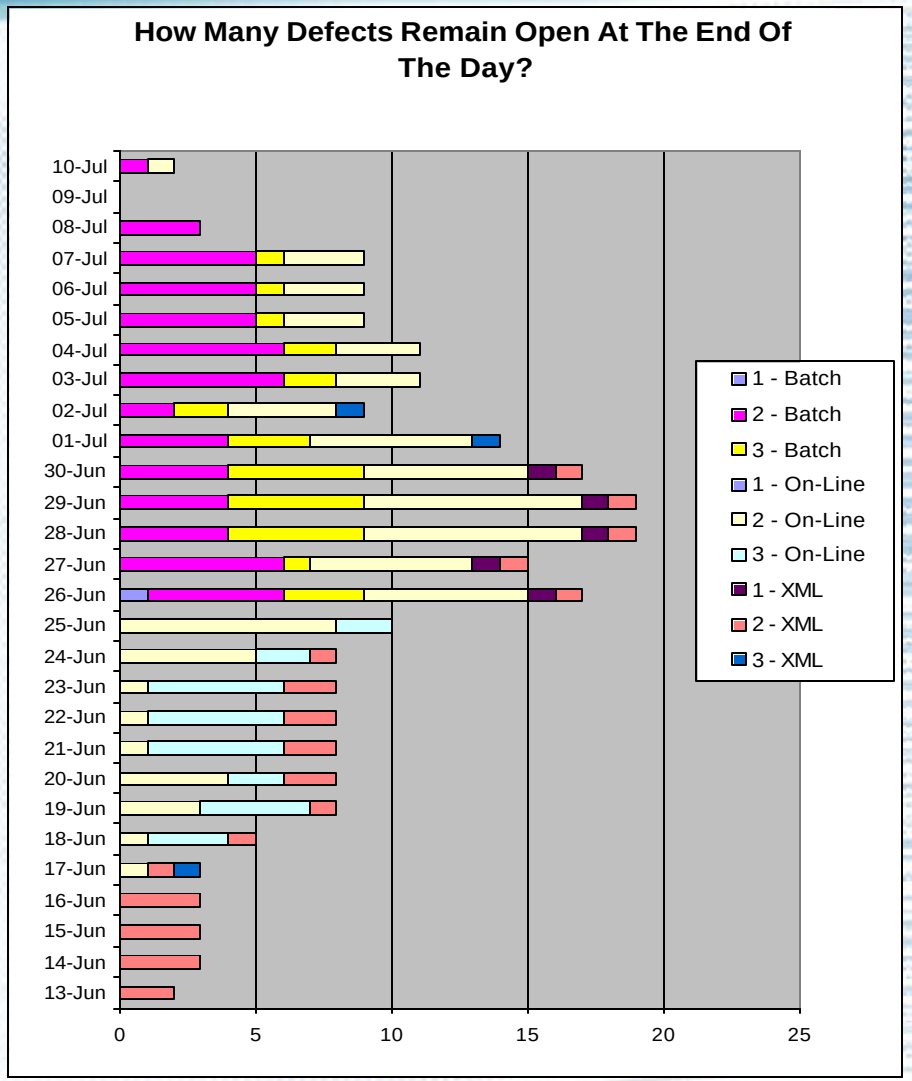
- Project reporting and readouts
- Decisions are based on a variety of quality factors
- Consistency
- Improved unit testing
- Quantified feedback to testers on their performance
- Confidence in the testing organization
- Minimal investment

Daily Status Report Examples

How stable is our application?

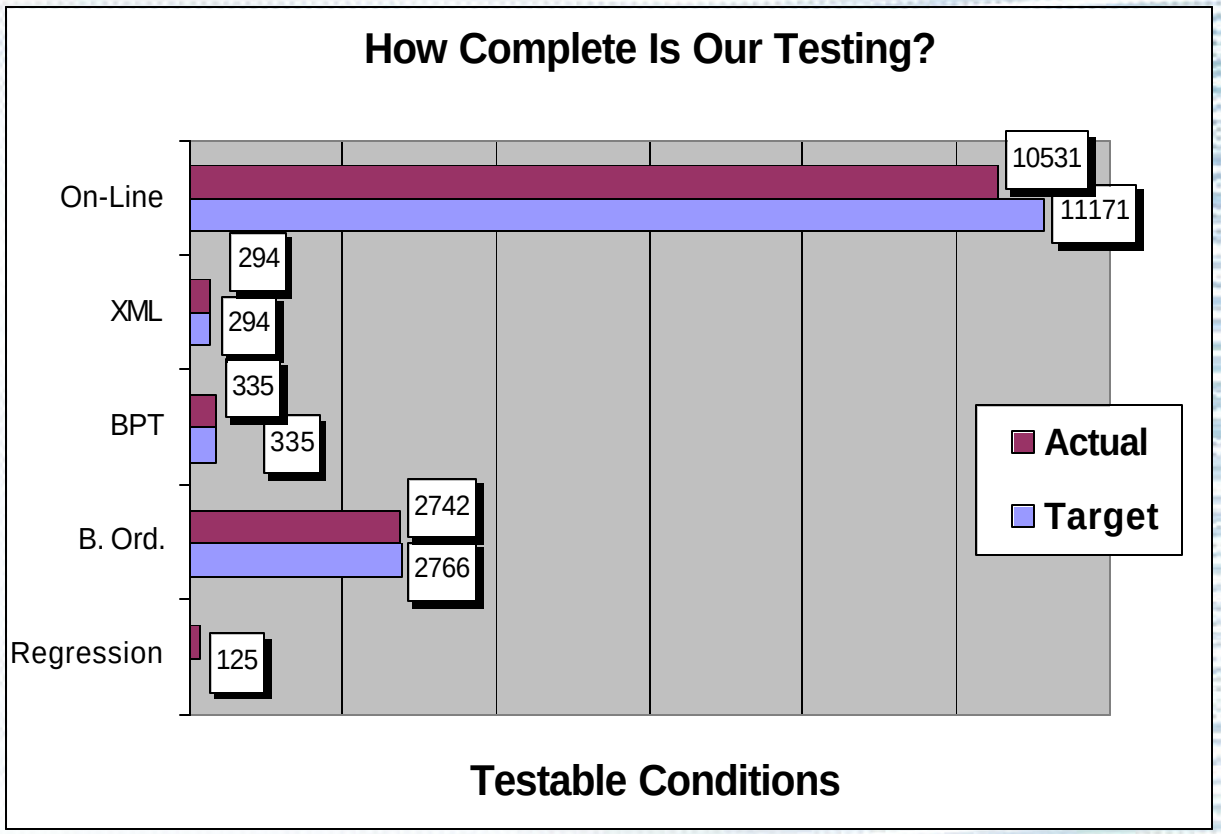
Curve illustrates code stability

- **Discovery phase**
- **Code adjustment phase**
- **Stabilization phase**



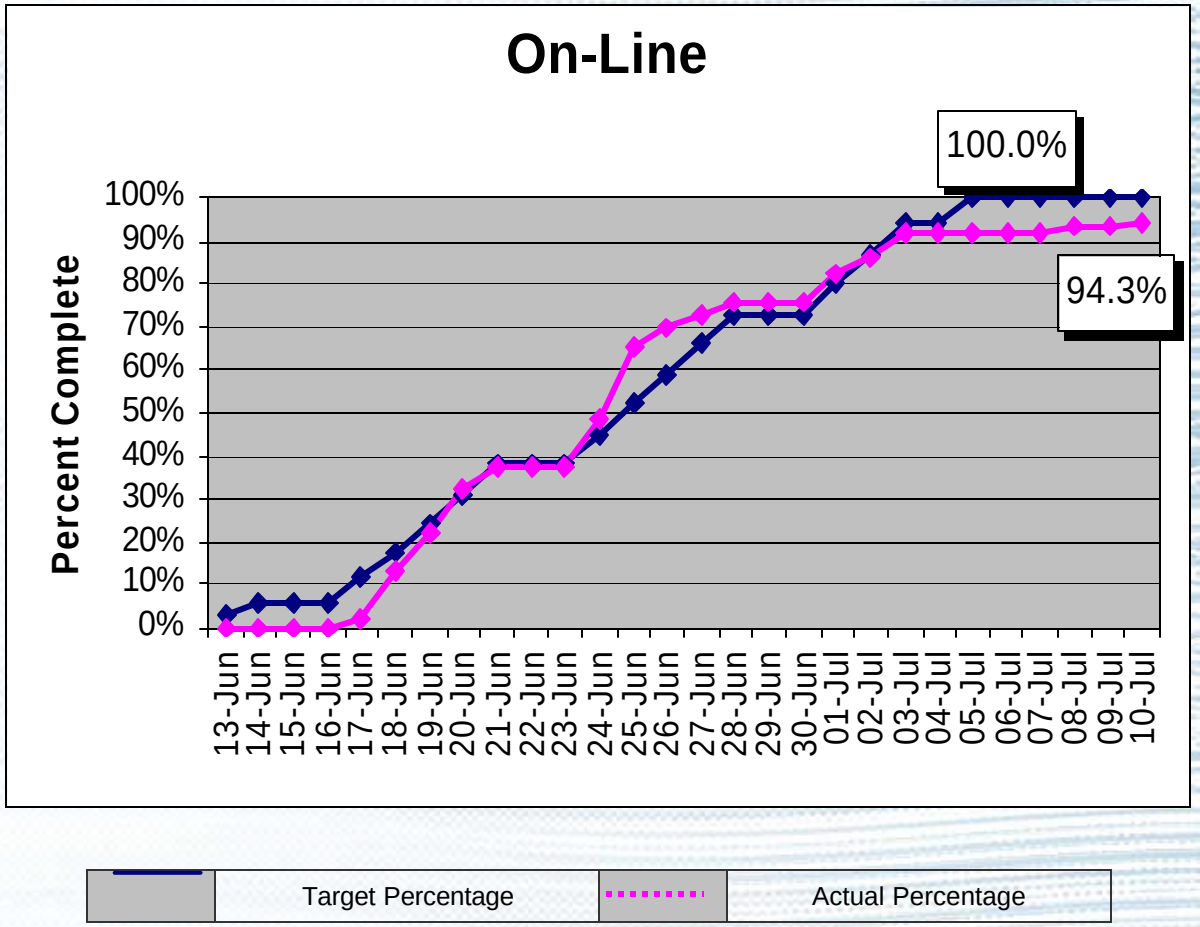
How complete is our testing?

Graph indicates level of coverage achieved during test

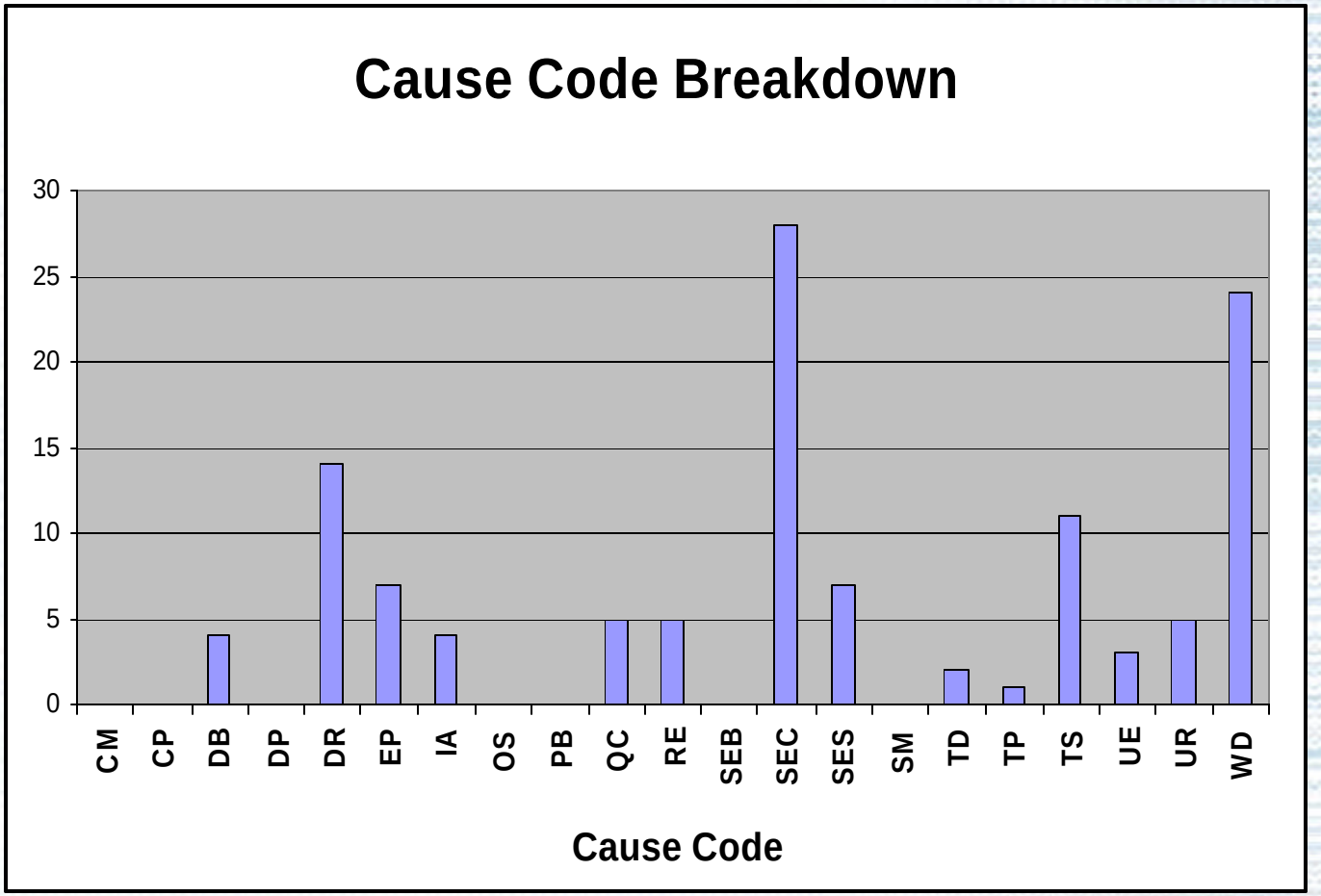


Are we on schedule?

Graph indicates where our test effort is in relation to schedule



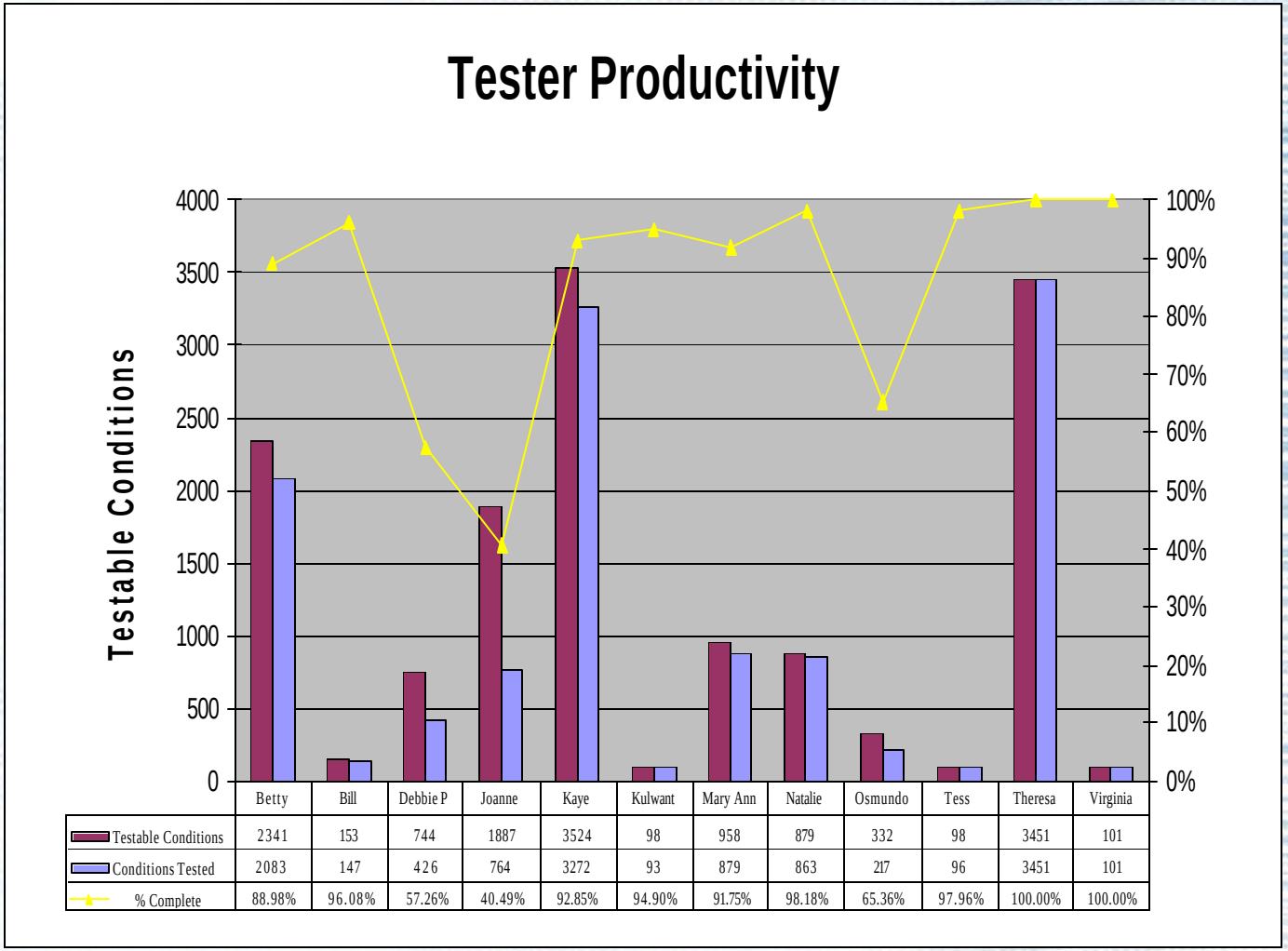
What is the root cause of our defects?



Code	Error/Problem
CM	Chg. Mgmt.
CP	Check-In
CV	Conversion
DB	Datbase
DP	Design
DR	Duplicate Record
EP	Existing Production
IA	Interfacing Application
OM	Order Manager Error
OS	Operating System
PB	PB Limitation
QC	QC Error
RE	Requirements
SE	Software
SEB	Error - Both
SEC	Error - Client
SES	Error - Server
SM	Software Movement
TD	Test Data
TP	Table Problem
TS	Test System
UE	User Error
UR	Unable to Reproduce
WD	Working as Designed

What is the breakdown for each tester?

Graph indicates where to shift resources

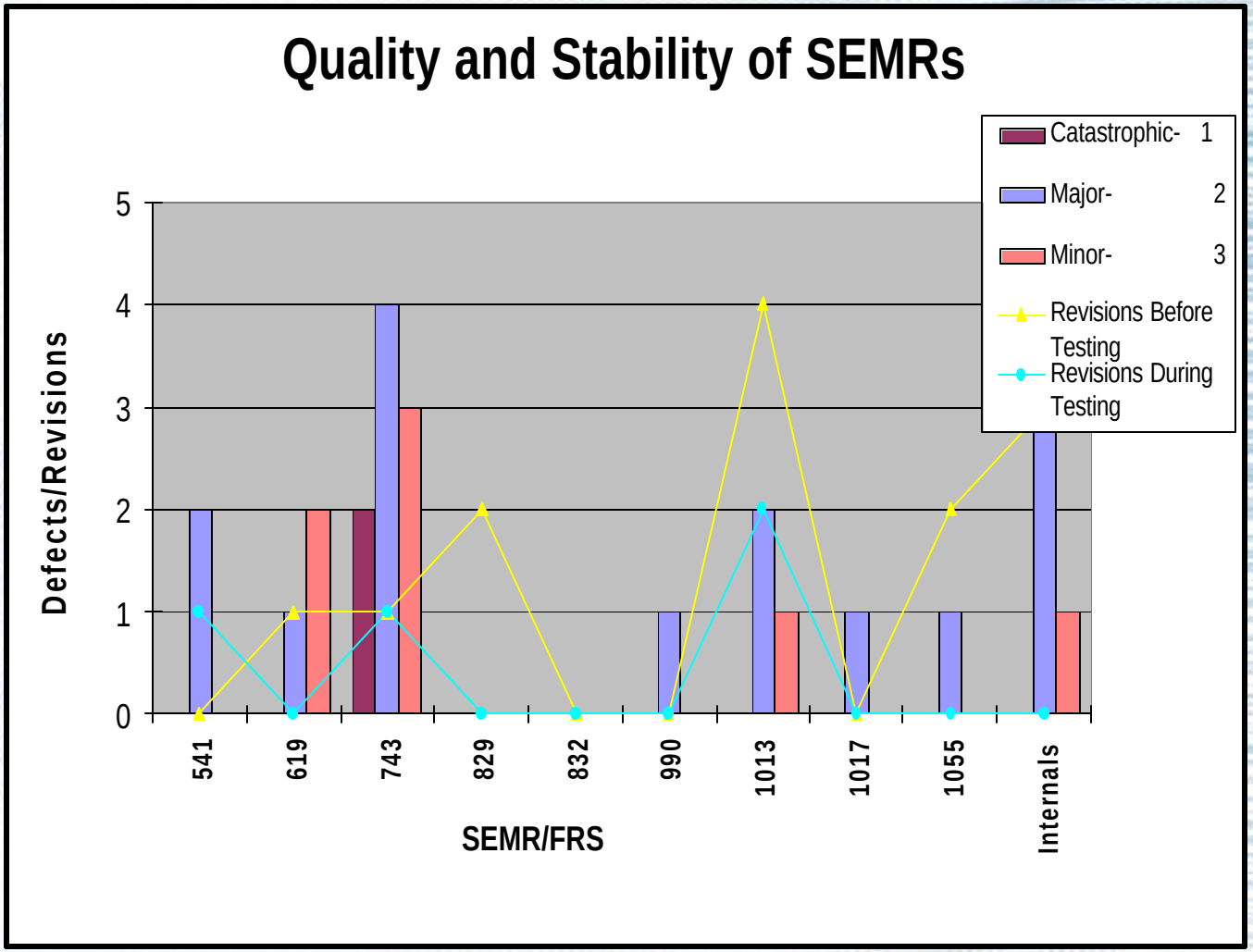


	Betty	Bill	Debbie P	Joanne	Kaye	Kulwant	Mary Ann	Natalie	Osmundo	Tess	Theresa	Virginia
Testable Conditions	2341	153	744	1887	3524	98	958	879	332	98	3451	101
Conditions Tested	2083	147	426	764	3272	93	879	863	217	96	3451	101
% Complete	88.98%	96.08%	57.26%	40.49%	92.85%	94.90%	91.75%	98.18%	65.36%	97.96%	100.00%	100.00%

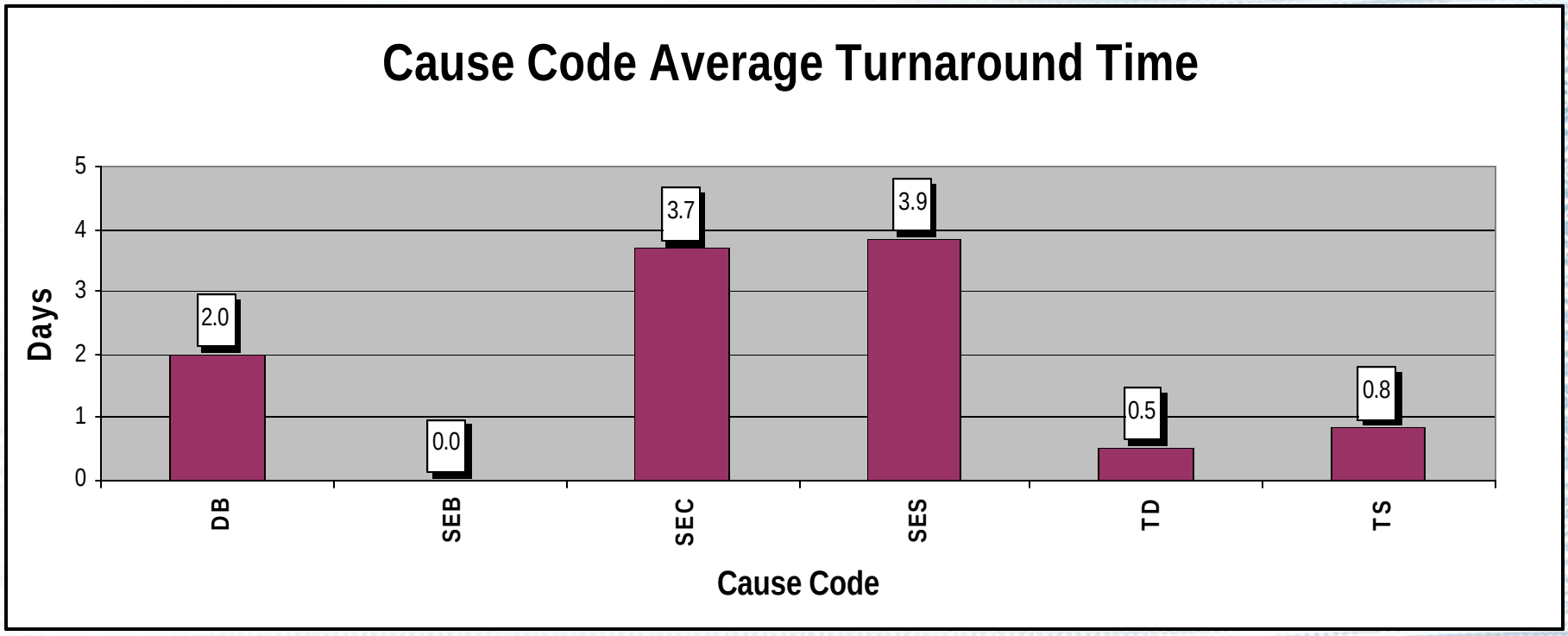
Release Analysis Report Examples

Correlation of requirements churn to software quality

Indicates impact to quality when late changes to requirements are worked in a release



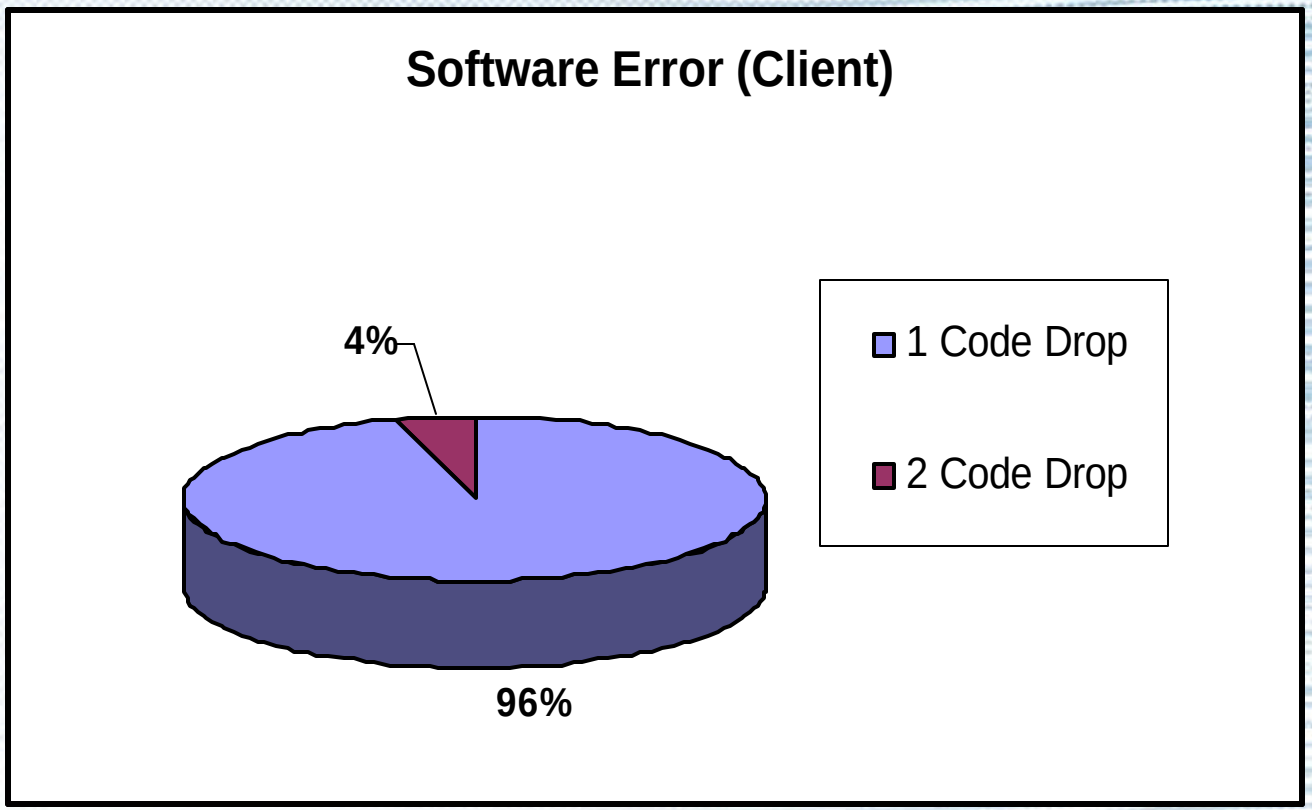
Turnaround time for each type of defect



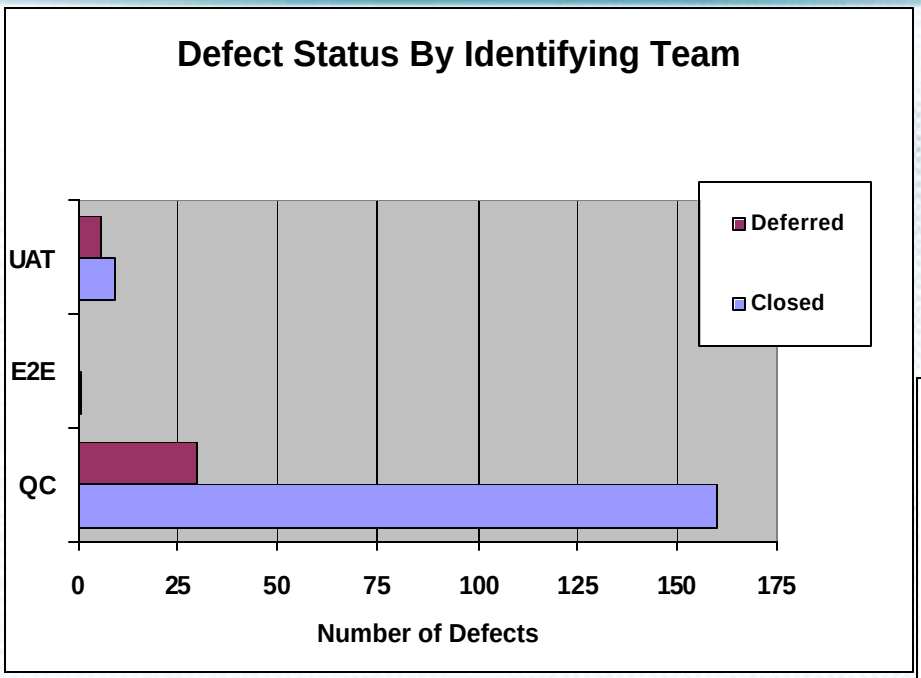
Graph indicates the average amount of time (in days) required to fix defects, by category (cause code)

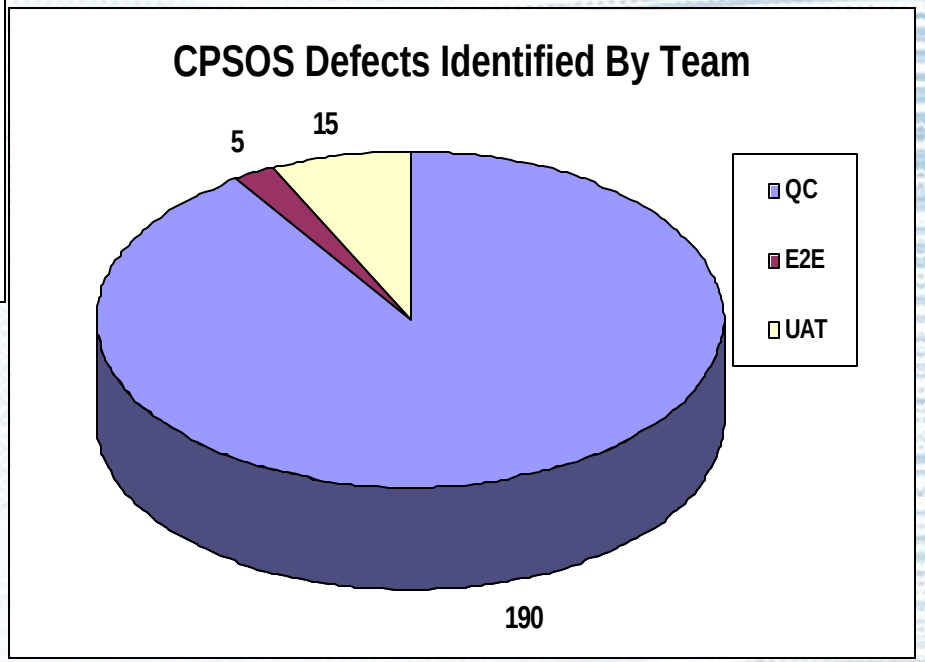
Defect correction success rate

Graph indicates the success rate on defect corrections, by team. How many times did it take?



Who discovered the defects?

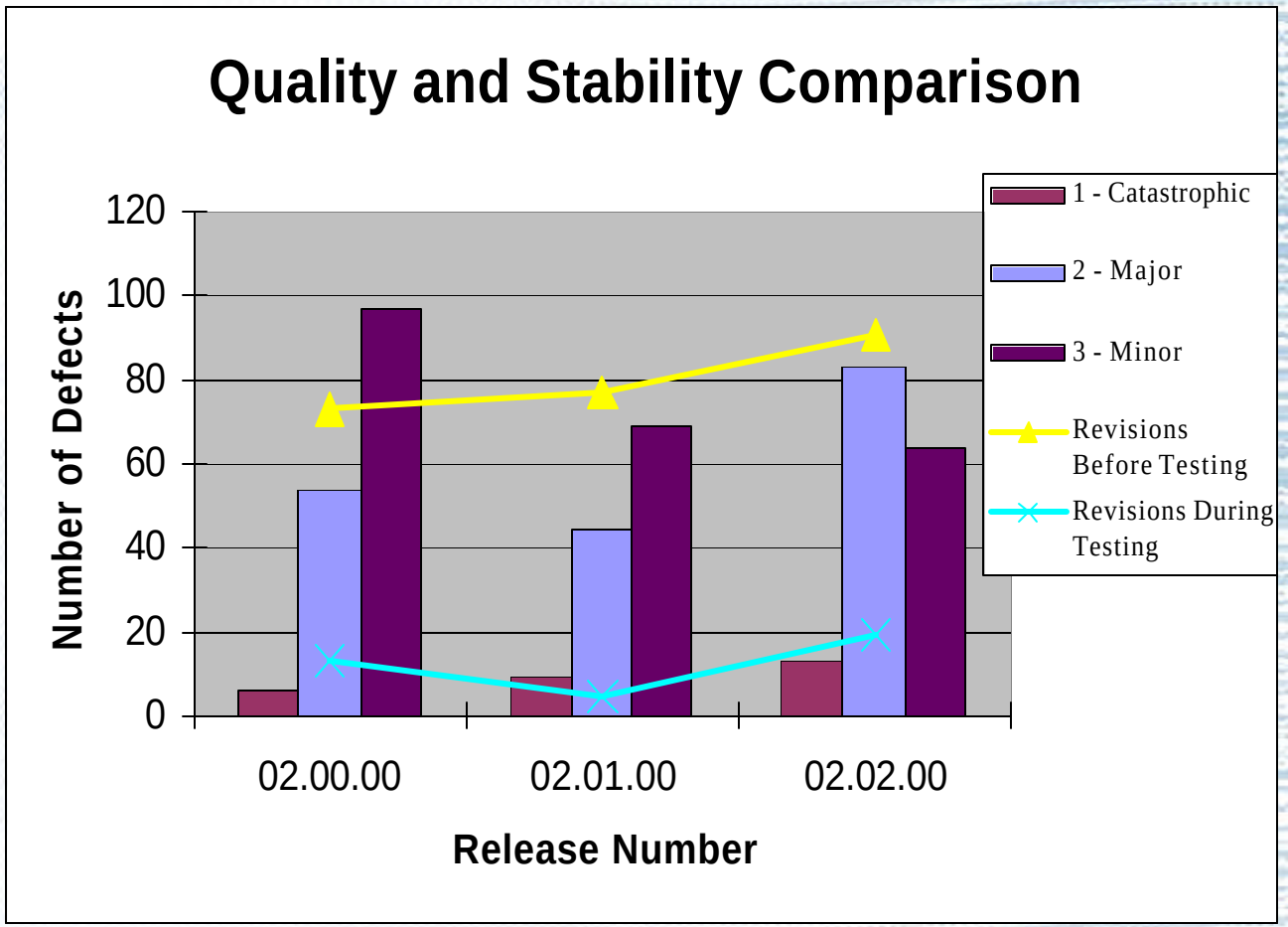




Graph indicates how many defects were reported by the various teams involved in testing (E2E=End to End)

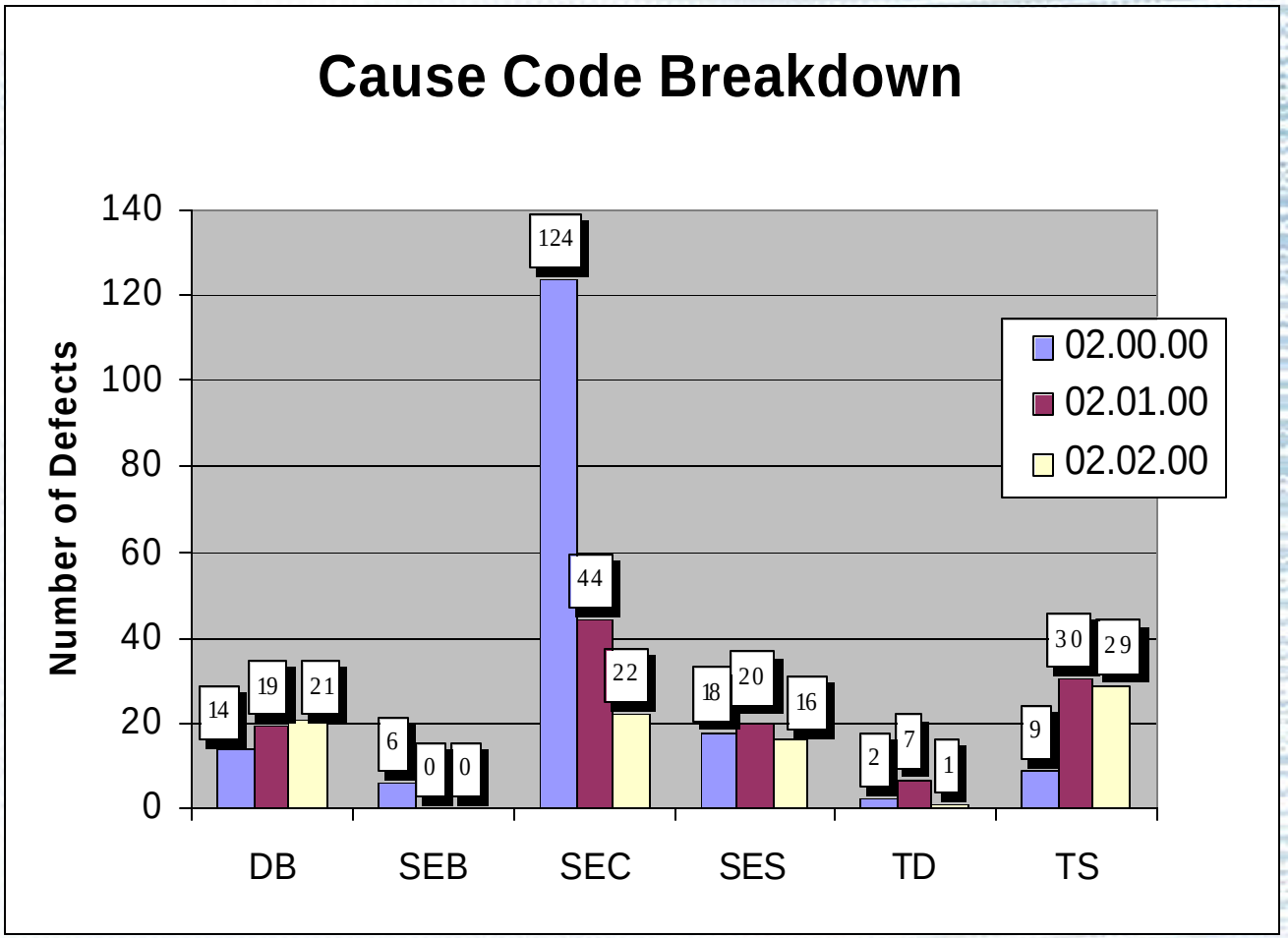
Historical quality and stability analysis

Graph looks at the late changes to requirements and their impact to overall quality for the last three releases



Historical cause code analysis

Graph compares the root cause of defects over the last three releases.



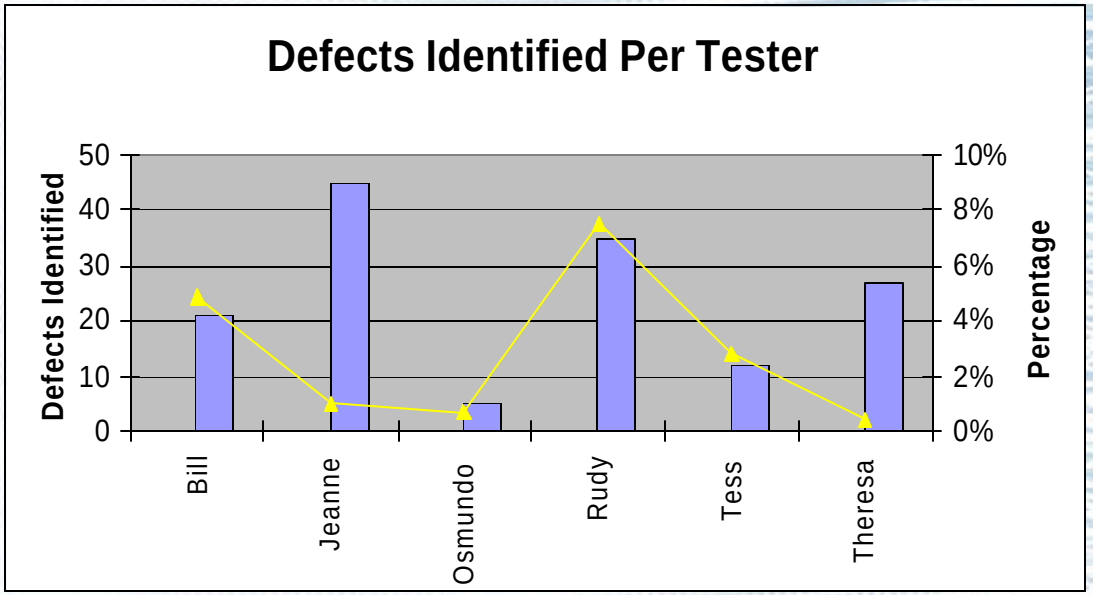
QC personnel performance analysis

Tester Defect Cause Codes

	DEF	CM	CP	DB	DP	DR	EP	OS	PB	QC	RE	SEB	SEC	SES	SM	TD	TP	TS	UR	WD	TOTAL
Bill	23	0	0	1	1	2	0	0	0	0	1	0	8	6	0	0	0	1	1	2	**23**
Jeanne	52	0	0	7	0	2	0	0	1	5	8	0	12	2	0	0	0	4	1	10	**52**
Osmundo	5	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	4	0	0	**5**
Rudy	42	0	0	2	0	5	0	0	0	2	11	0	1	2	0	2	0	6	0	11	**42**
Tess	15	0	0	1	0	3	0	0	0	0	0	0	0	1	0	2	0	5	1	2	**15**
Theresa	30	0	0	3	0	1	0	0	0	2	2	0	3	1	0	2	0	0	5	11	**30**
TOTAL	**167**	**0**	**0**	**14**	**1**	**13**	**0**	**0**	**1**	**9**	**22**	**0**	**24**	**13**	**0**	**6**	**0**	**20**	**8**	**36**	**167**

Tester Defects Reported

		Defects		DR, QC		Testable Conditions		Defects Identified	%
Bill		23		2		431		21	4.9%
Jeanne		52		7		4554		45	1.0%
Osmundo		5		0		722		5	0.7%
Rudy		42		7		466		35	7.5%
Tess		15		3		429		12	2.8%
Theresa		30		3		5932		27	0.5%
Total		167		22		12534		145	1.2%



Questions?

Rick Tennant
SBC Communications, Inc.
One SBC Center, 38-L-4
St. Louis, MO 63101

Five Star Testing Reports on a One Star Budget

Cost effective reporting that returns value to your project team

Abstract

This paper provides the strategy and approach in order to design and produce useful project metrics from basic test planning and defect data. Timely and accurate data is essential on a fast paced project team – especially one that operates in an intensely competitive marketplace. A project team at SBC Communications has implemented the approach described within this paper.

Why are good status reports important?

In a fast paced project environment, when time to market is critical, sound project decisions are crucial to your success. Project managers require information on product quality, stability, and schedule to make the right decisions.

Some of the problems we were having before we developed our reporting:

1. Project readouts and status information were inconsistent and unorganized. Our project team could not rely on information being correct and critical information was buried in detail. Decisions were made on "gut feelings" or were based on sporadic readouts.

2. Project managers and senior management never had a complete story on the release. Poor decisions were being made due to a lack of metrics.

3. Budgets and resources were tight. We did not have the luxury of adding a person to create reports or build a web site. Money was not available for tools or consultants. Due to extremely tight schedules and business demand, process changes were difficult to implement.

4. Potential information available for testing metrics was not always captured in a consistent manner. Defect information was not always maintained. Analysis required for root cause analysis was inconsistent and incomplete.

5. Many different groups were requesting testing status and reporting. Our project team, clients, and management wanted the same type of information – our communications did not meet everyone's needs.

6. Due to the above mentioned problems, the Quality Control team suffered from a lack of credibility. The readouts were not organized, information was inconsistent, and the data was not professionally presented. Groups wondered how well we were really testing when we couldn't provide accurate and consistent detail on our progress and results.

We had many areas to address in our testing team to improve the quality and efficiency of the testing services provided to the project. Creating test metrics to drive process improvement in the test group and project team was a logical place to start.

What do you need to consider when designing your test reports?

Trying to communicate with a diverse audience can be extremely challenging. Each project stakeholder has unique drivers. Analysis to determine what is information is important to your audience is critical in the acceptance and use of these reports. Organizing and presentation is critical to ensure that the message is not subject to interpretation. Here are some of the important elements to consider when you are deciding on what type of reports are needed:

1. Determine who will be reading and using this data. Who needs this information to make release decisions? Who needs this information to drive action during the testing windows? Who needs this information to make adjustments during the release cycle? Who needs to be aware when you are behind schedule or quality is at questionable levels? Consider the project team members in your Information Technology department, your clients, and senior management.

2. Report layouts and the information being presented should benefit your audience in their decisions. What types of decisions to each of these stakeholders make? What information is critical? What types of information do these people need in order to do their jobs? What information benefit would benefit *all* of the project stakeholders? Here are some questions that our project team addressed with our reporting:

   - *What is the status of our defects? How severe are they?*
   - *How complete is our testing? What level of coverage have we achieved?*
   - *Is our testing on schedule?*
   - *How stable is the product?*
   - *Do our resources need to be reallocated in order to put us back on schedule?*
   - *What is the root cause of our defects?*

3. Data used to create test reports must be identified. Data must be recorded in a consistent manner. Process should be in place to ensure that the information being captured is accurate. In order to ensure that your reports are accepted and used, data integrity should be without question. Here are the data elements (from our defect reports and test planning) used by our project team:

   Defect data fields

   - *Originator, date, time*
   - *Assigned to, date, time*
   - *Application name*
   - *Release number*
   - *Test phase*
   - *Issue/Defect status*
   - *Date/time problem fixed*
   - *Date/time fix validated*
   - *Cause code*
   - *Priority*

Test planning data

– *Testable items (requirements)*

1. Combine multiple needs into a single chart or graph. The first page of a status report is critical. Address the most critical needs and satisfy as many needs as you possibly can with that page. Put multiple graphs and charts on the same page. Subsequent pages should be used as supplemental input into project decisions or to address individual stakeholder needs (including the test team's needs!)

2. Spend extra time on designing your charts and graphs so that all of your stakeholders get something out of it. Do not use technical jargon, complex plotting of data, and ambiguous titles – ***KEEP IT VERY SIMPLE!***

3. When considering your data sources for reporting, look at what you are capturing today. Can you use this data to address some of your project stakeholder needs? Basic defect and test planning information can be used to answer many questions.

Tips on cost effectiveness

In today's environment, financial impact is a consideration in everything we do. Taking what you have and making the best of it is often our only choice. Here are some of the lessons we learned during this effort:

1. Find ways to drive your information for your current sources. Clean up data and processes to ensure data integrity and consistency.

2. Automate whenever possible, including data collection, report creation, and distribution.

3. Investigate MS Excel or your in-house spreadsheet application and other low cost tools for report creation. Try to use what you already have!

4. Only produce reports, charts, and graphs that are being used by your stakeholders.

Overall benefits

Effective and accurate test reporting will yield a significant return on investment, as it provides a dependable barometer to provide a basis business decisions. Good testing metrics can also be used as a tool to evaluate and develop the talent in your testing organizations. Find out who on your team is the most efficient tester, who finds the quality defects, and who can handle the extra workloads.

Project wide, the benefits are significant. Historical trends can justify change to your clients and trigger process improvement within your IT department.

Sample Reports

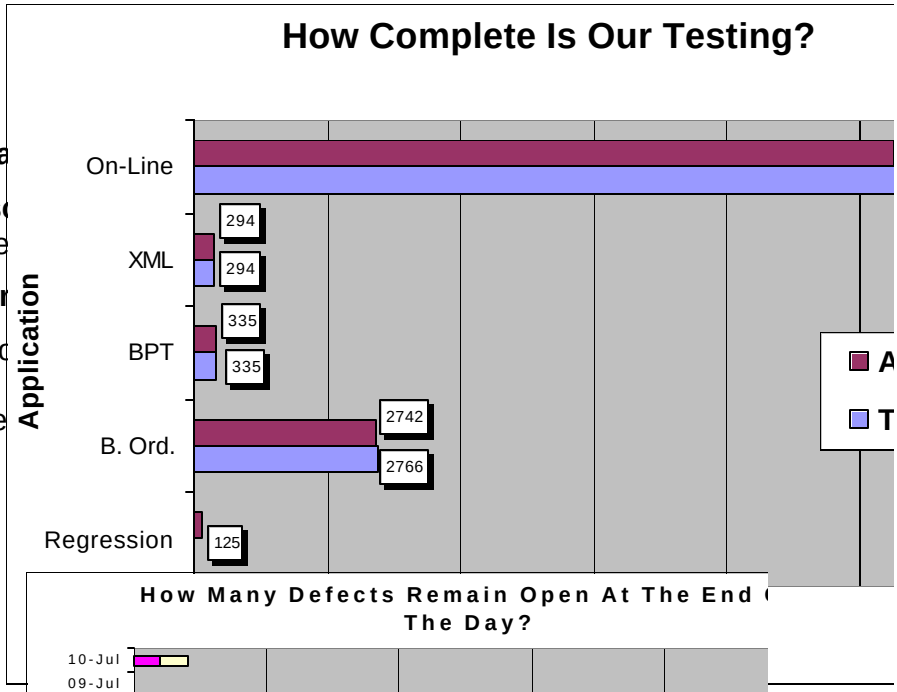
Report Name: How complete is our testing?

Brief Description: Provides summary information on the number of test conditions that have been validated.

Data Source(s): Testable items (requirements) for the release and daily input from the testers on how many requirements were validated (Note: Test case forms are in MS Excel and this information is imported into the test report).

Use: Determine how complete the test is; how many conditions have been validated. Detailed reports are available for each change request to further define problem areas.

Sample:



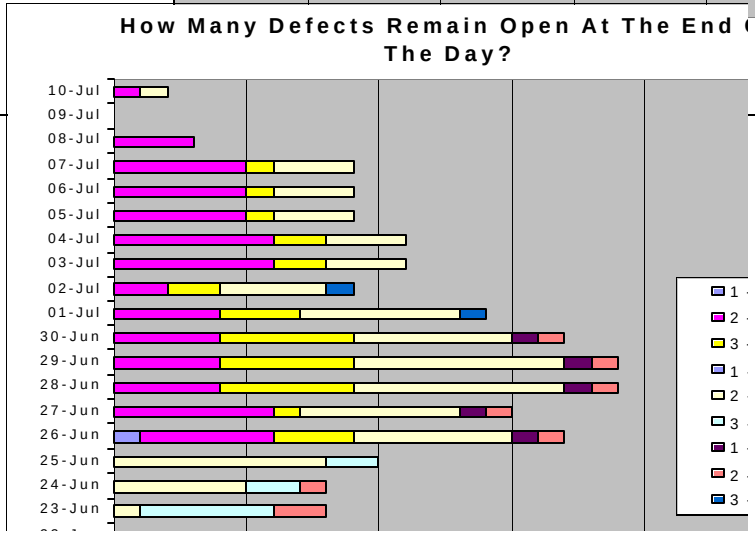
Report Na

Brief Des
Defect line

Data Sour

Use: Repo
to release
serious de

Sample:



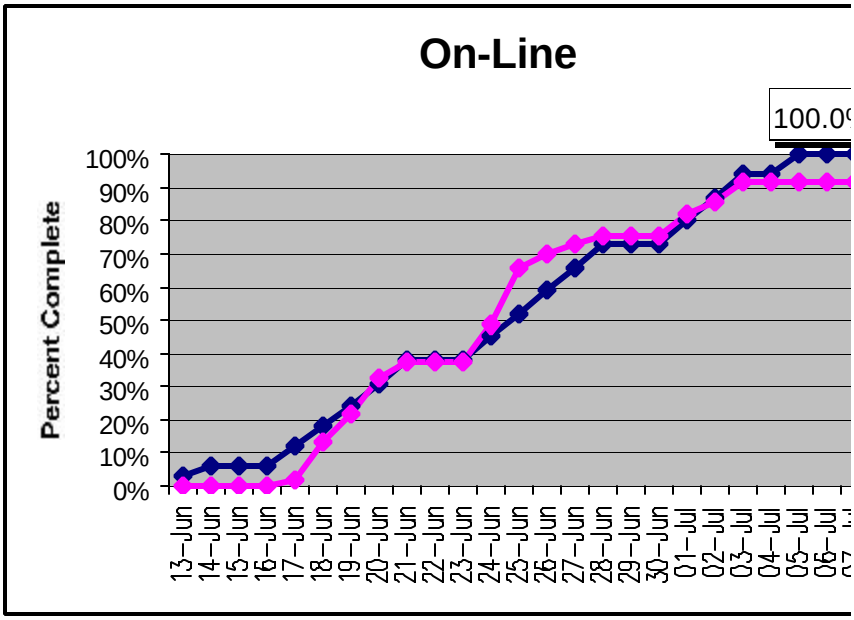
Report Name: Are we on schedule?

Brief Description: Provides project and actual data for conformance to schedule using testable items as the basic unit of measure. (Note: Flat lines are unscheduled weekends).

Data Source(s): Testable items, daily log of items actually tested (derived from our test cases)

Use: Report indicates whether we are on schedule or not. Used to trigger resource adjustments, determining whether overtime is required, also helps if confronted with a decision on late requirement change (used in conjunction with the stability chart shown on the previous page.

Sample:



Report Name: Tester Productivity (Daily)

Brief Description: Provides analysis on how complete each tester is with their release assignment (including percentage)

Data Source(s): Testable items by tester, daily log of items actually tested (derived from our test cases)

Use: Used to shift resources to help out people that may be behind, on vacation, or a testing a change that has significant issues.

Sample:

Tester Productivity