SRE at Agilent Technologies

Colorado Springs

Audience Participation Version

http://www.netbox.com/tbc/hacks/issre2k.htm

By Tim_Chambers@Agilent.com, 11 October 2000

Agilent Technologies
Innovating the HP Way

- Characteristics of our organization
- History
- Processes & techniques
- Measurable improvements

Agilent Technologies

Products For Digital Designers

Characteristics of Our Organization

- Products for digital design engineers
  - Microprocessor development systems (emulators)
  - Logic analyzers
  - Oscilloscopes
- Software ranges from UNIX applications to embedded firmware
- SRE principles practiced for over ten years

History

- Bugbusters project was our wake-up call (mid-1980's)
- SRE evolved along with the software development process
- Basics have remained unchanged
  - "Abuse" the product instead of developing operational profiles
  - Measure time spent testing
  - Measure defect rate (weighted by severity)
  - Measure turmoil (NEW)

Processes & Techniques

- Applying the 'Black Team' principle
- Measuring test time
- Weighting defects
- Adjusting the moving average to smooth the defect rate
- Counting source statements to measure defect density
- Measuring how a software system changes over time (turmoil)

Applying the 'Black Team' principle

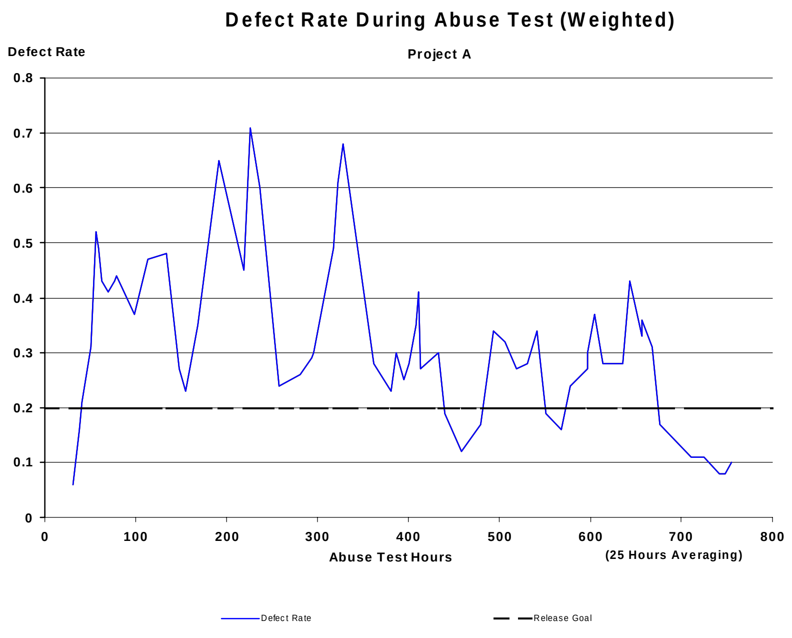
- "Given enough eyeballs, all bugs are shallow."
- Recruit testers
- Schedule time commitments
- Heuristics to plan tests:
  - Code changed since last test cycle
  - New features
- Abuse testers in QA area
- Other benefits

Measuring Test Time

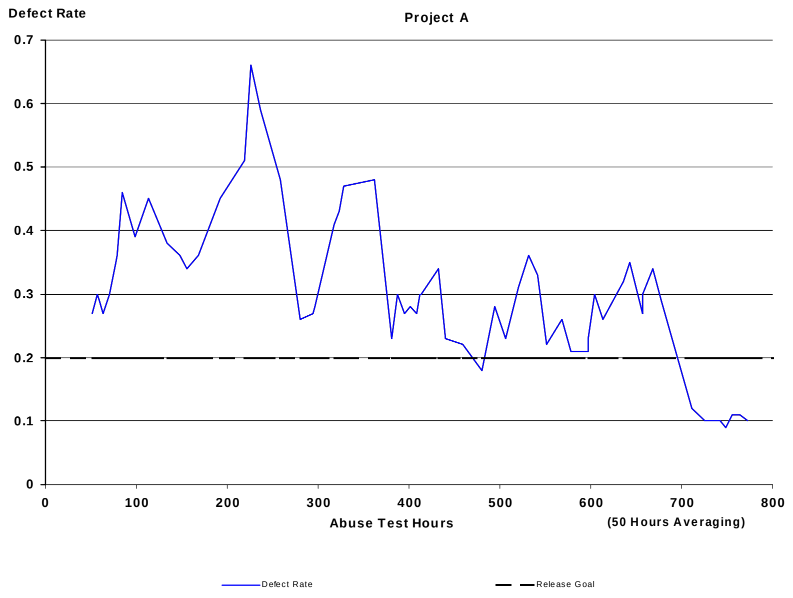
- Track time to nearest half-hour
- Count all test activities
  - Running the test
  - Configuring the system
  - Studying test plans
  - Reading documentation

Defect Severity Classification

Critical ***(weight = 1.8)*** • Product is unusable due to: • Crashing (terminates in unexpected or abnormal manner) • Hanging (stops responding to normal user interactions, must kill or restart) • Data loss (stored data or user input is lost, changed, or added to in a manner not consistent with normal operation) • Product is usable, but produces incorrect data and does not warn of error	**Serious** ***(weight = 1.4)*** • Product does not perform as advertised in datasheet, manual, application literature • Product is usable, but severely restricted • Customer cannot use the product to get their job done without risking the loss of data from the current work session or loss of previously acquired data • Product is usable but produces incorrect data and warns that an error may have occurred
Medium ***(weight = 0.8)*** • Product is usable, but there are some limitations in the feature set • Problems are obvious and repairable without extensive effort • If a defect impairs the ability for the user to get their job done, then a workaround solution must exist • The customer is annoyed, but the defect does not cause them to stop using the product • The product does not perform the way it was intended to, but the user can still get the job done	**Low** ***(weight = 0.2)*** • Product is functionally correct • Product is usable with a small amount of inconvenience • Product performs in an inconsistent manner, or may require some effort by the user to proceed • Workarounds are obvious and easy to use • Cosmetic defects, on-line help defects, packaging defects • Customer experiences a small amount of frustration because the product contains minor problems, but they are able to ignore them • Defects are easy to fix and have a low impact on the product without a high priority to be fixed • The customer probably will not call Agilent to complain about this defect



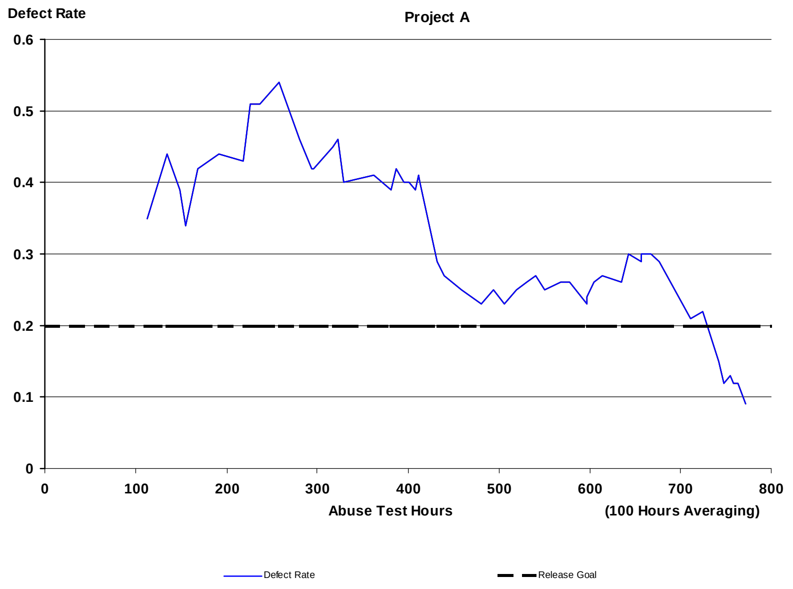
Defect Rate During Abuse Test (Weighted)



International Symposium on Software Reliability Engineering
11Oct00

Agilent Technologies

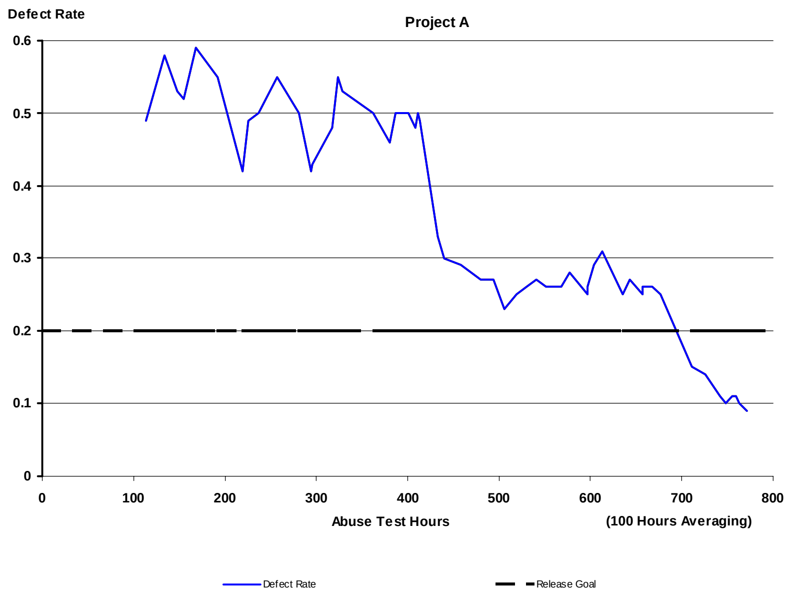
Defect Rate During Abuse Test (Weighted)



International Symposium on Software Reliability Engineering
11Oct00

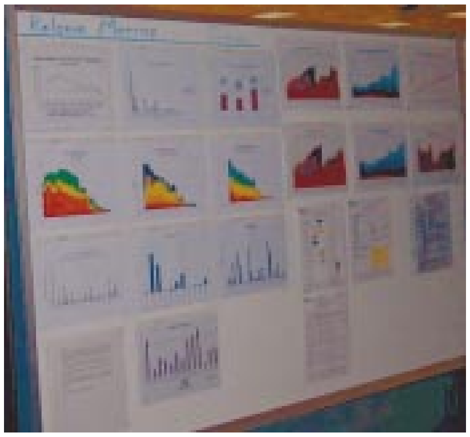
Agilent Technologies

Defect Rate During Abuse Test (NON-Weighted)



Project	KNCSS	Pre-release Defect Density	Post-release Defect Density
A	1,637	0.252	0.015
C	60	2.267	0.000
D	98	1.143	0.010
E	96	8.562	0.333
F	446	3.303	0.157
G	123	10.859	0.016

Department Software Metrics Wall



International Symposium on Software Reliability Engineering
11Oct00

Agilent Technologies

Presentation materials are accessible on the Web:

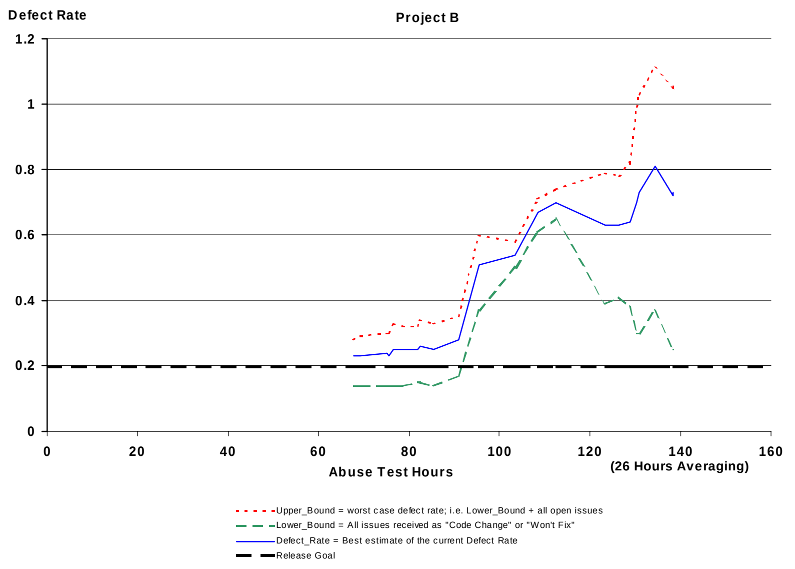
http://www.netbox.com/tbc/hacks/issre2k.htm

URL guaranteed good through March, 2002

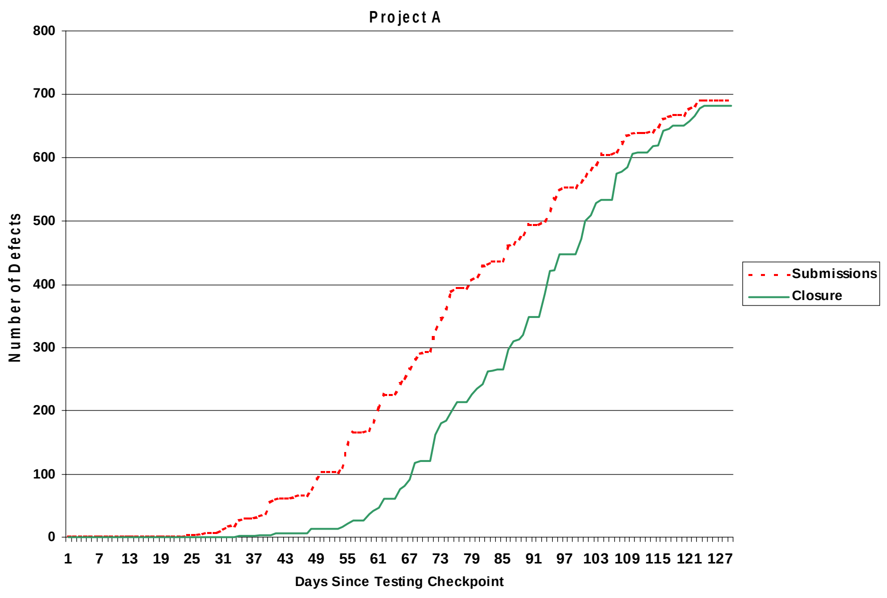
International Symposium on Software Reliability Engineering
11Oct00

Agilent Technologies

Defect Rate During Abuse Test (Weighted)



Find and Fix Rate



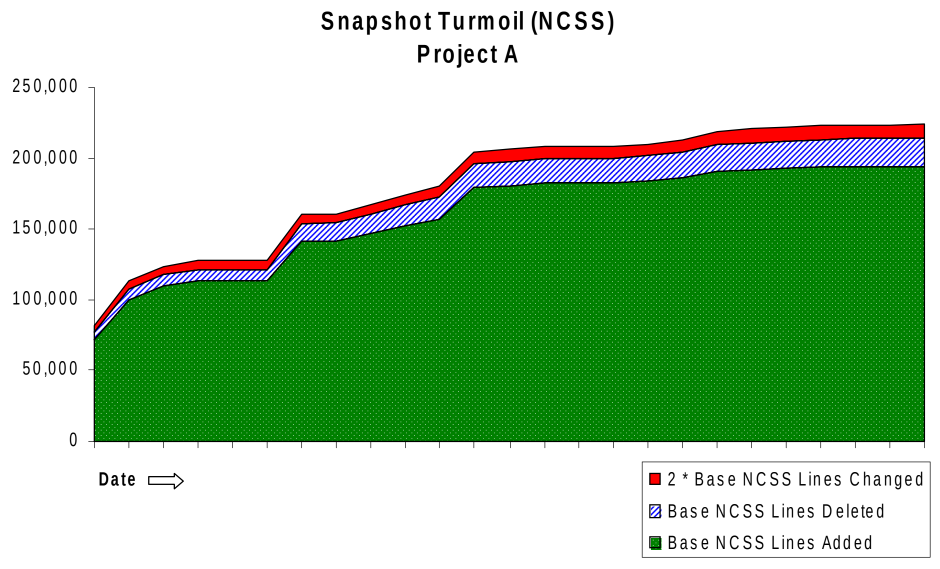
Turmoil

Turmoil is a measure of code stability. The turmoil of a selected **release** is calculated for the first time by identifying all source files at a point in time early in the development cycle. For subsequent data points, each file is compared against the previous version, and the number of non-commented source statements (NCSS) added, deleted, and changed is counted. The number of NCSS changed is then doubled. The rationale is that a changed line can be represented by the transformation: delete the old line, add the new one.

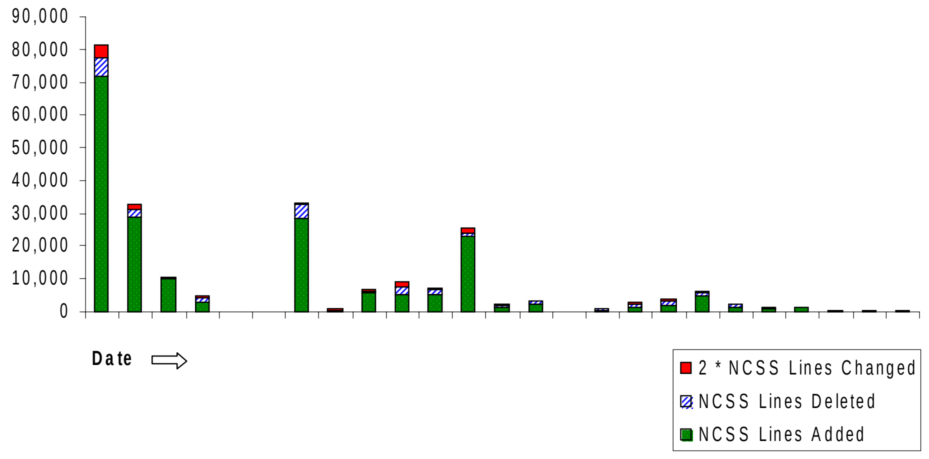
Two types of turmoil are measured over time. For *snapshot* turmoil, the differences are always measured against the baseline version. *Incremental* turmoil is measured by comparing the newer version against the version used for the previous turmoil measure.

Snapshot turmoil measures the accumulated turmoil since the **release** diverged from the baseline and plots the trend over time. It is useful for determining the amount of testing effort needed.

Incremental turmoil is a differential between the times plotted. It is useful for measuring the stability of the code and for tracking progress toward closure. If incremental turmoil falls to zero for an extended period, then it is likely that the code is stable enough to release.

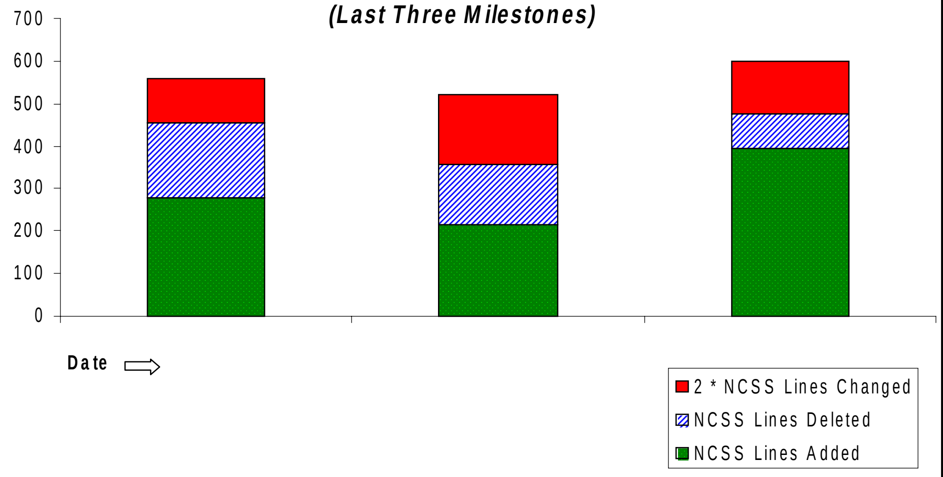


Incremental Turmoil (NCSS)
Project A



Agilent Technologies

Incremental Turmoil (NCSS)
Project A
(Last Three Milestones)



Agilent Technologies