

# Core Requirements Overview

December 2007

David Flater

National Institute of Standards and Technology

dflater@nist.gov

# Strategy for this presentation

- Overall organization
  - Series of major topics
  - List of topics not addressed
  - Discussion
- Goals
  - Explain the most significant changes, with impact
  - Answer questions
- Non-goals (unless someone asks)
  - Less significant changes
  - Long tutorials
  - Technical details

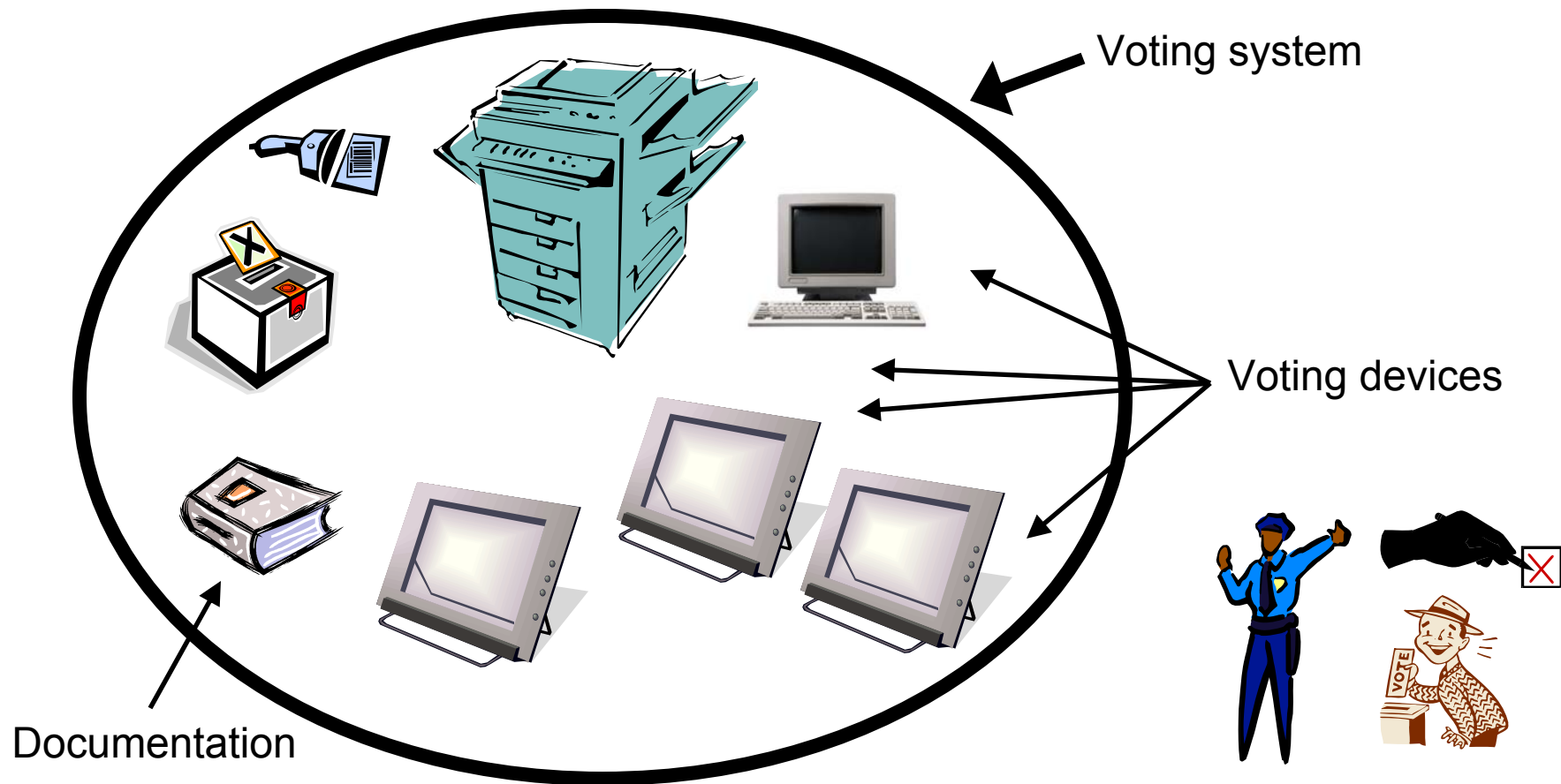
## List of major topics

- Systems versus devices
- Benchmarks and related test methods
- “COTS”
- Accuracy for optical scanners
- Post-election reporting requirements

## List of major topics

- **Systems versus devices**
- **Benchmarks and related test methods**
- **“COTS”**
- **Accuracy for optical scanners**
- **Post-election reporting requirements**

# System and device



## Meaning in requirements

- “Voting systems SHALL...”
  - Means: The system *as a whole* shall do this
  - The specific devices involved may vary
- “Voting devices SHALL...”
  - Means: *Each* and *every* voting device shall do this, individually
- “DREs SHALL...”
  - Means: Each and every *DRE* shall do this, individually

# Voting *process*

- People and processes are
  - Outside the scope of a product standard
  - Not included in the definition of voting *system*
  - Not assessed by test labs except indirectly, “as specified by the manufacturer”
- Where the requirement on the system is to “play nice” with a certain process, the VVSG refers to the voting process, but does not constrain the process



# Impact of changes

- Clarified the applicability of requirements
  - To systems that combine different technologies
  - To different classes of voting devices



## List of major topics

- Systems versus devices
- **Benchmarks and related test methods**
- “COTS”
- Accuracy for optical scanners
- Post-election reporting requirements

## Reliability, accuracy, misfeed rate

- Both the old benchmarks and the old test method had issues
- New benchmarks based on estimates provided by NASED representative to TGDC
  - All numbers old & new considered somewhat arbitrary
  - There is no “typical” case
- Corrected mistakes
  - Average case vs. worst case
  - Observed vs. demonstrated
- New test method fixes problems and is simpler

## Benchmark terms

- Reliability → failure rate
  - There is a precise (but complex) definition of failure designed more for arbitration than readability
  - In plain language, failures are equipment breakdowns, including software crashes, such that continued use without service or replacement is worrisome to impossible
  - Normal, routine occurrences like running out of paper are not considered failures
  - Misfeeds of ballots into optical scanners are handled by a separate benchmark, so these are not included as failures



# Reliability benchmarks

- Part 1 Sections 6.3.1.2 and 3 describe the scenario and the estimates from which the reliability benchmarks are derived
  - Different device classes have different reliability requirements depending on how they are used, how easily replaced, etc.
- Section 6.3.1.4: Derivation of reliability benchmarks based on 1 % risk of exceeding tolerances
  - Special case: Benchmark for failures resulting in disenfranchisement set to zero

## Benchmark terms

- Accuracy → error rate
  - Specifically, the report total error rate defined by Part 3 Req. 5.3.4-B
  - Plain language: Observable discrepancy between the reported number and the correct result
  - Not the human factors meaning of accuracy (usability testing)
  - Strictly a measure of mechanical performance
  - Bad inputs are thrown out

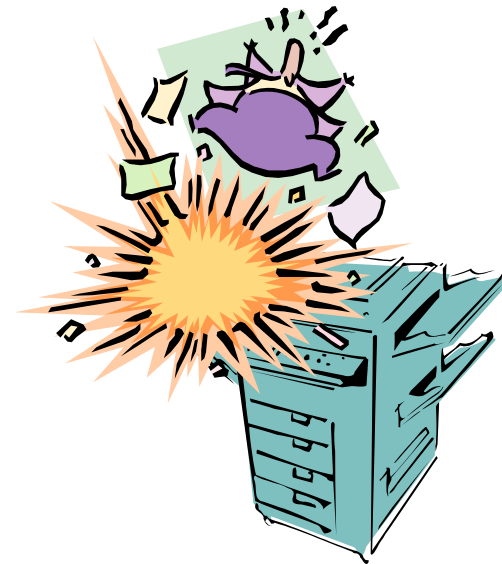


# Accuracy benchmark

- Part 1 Requirement 6.3.2-B
- Interpretation of benchmarks appearing in VVSG 2005 and 2002 VSS
  - Intermingling of requirements with test method resulted in multiple benchmarks
  - Derived from the "maximum acceptable error rate" used as the lower test benchmark in VVSG 2005 (ballot position error rate of 1 / 500 000)
- Metric is different
  - Old metric (ballot position error rate) was problematic
  - Conversion to new metric (report total error rate) explained in the discussion field

## Benchmark terms

- Misfeed rate = combined rate of bad things that scanners do to paper ballots
  - Multiple feed
  - Paper jam
  - Rejection of a ballot that has nothing wrong with it



## Misfeed rate benchmark

- Part 1 Section 6.3.3
- Has ranged between 2 % (i.e., 1 in 50 ballots) and  $10^{-4}$  (1 / 10 000)
- Per input from NASED representative, now set at .002 (1 / 500)



## Relevant procedural changes for testing

- Part 3 Section 5.3: Benchmarks are now evaluated over the course of the entire test campaign
- Part 3 Section 2.5.3: Not permissible to bypass portions of the voting system that would be exercised in an actual election
- Part 3 Req. 5.2.3-D: Volume test

## Part 3 Req. 5.2.3-D: Volume test

- “Large” test in conditions approximating normal use in an election
  - More equipment
  - More people
  - Simulated election day
- This is a general functional test, not designed just for evaluating the benchmarks
- However, data collected during this test contribute substantially to the evaluation of the benchmarks

# Impact of changes

- Benchmarks are more defensible
- Can no longer get a pass on reliability and accuracy just by “acing the exam” – must be consistently reliable and accurate throughout the entire test campaign
- More valid system test
  - Simulated ballots forbidden, replaced by volume test

## Impact of changes

- Test method improvements are free
  - Making better use of data collected throughout testing campaign
- Volume test is new (sort of)
  - If *only* EAC certification is considered, it is a new test with added cost
  - On the other hand, it reduces cost vs. having multiple states do their own volume tests
  - Size and scope made to agree with California Volume Reliability Testing Protocol for DREs

## List of major topics

- Systems versus devices
- Benchmarks and related test methods
- **“COTS”**
- Accuracy for optical scanners
- Post-election reporting requirements

## Terms

- COTS
  - Originally, acronym “Commercial Off-The-Shelf”
  - Now understood more broadly—any “standard” part, package or software that is widely used
- Logic
  - Complex functions are typically, but not necessarily, implemented in software
  - Neutral term “logic” covers all possible designs—software, firmware, hardwired, mechanical, ...

## Why COTS is important

- It has often been claimed that the VVSG grants COTS a blanket exemption from testing
  - An uncharitable interpretation of possibly confusing language
  - VVSG language had to be understood in context: hardware vs. software, new systems vs. retesting of slightly modified systems, ...
- Certain borderline cases were not anticipated

## Terms

- COTS: includes shrink-wrapped commercial products as well as analogous open-source packages
  - General-purpose
  - Widely used
  - Unmodified
- Application logic: logic from any source that is specific to the voting system, with the exception of border logic
- Border logic: “glue code”
- Third-party logic: neither application logic nor COTS
  - So-called “modified COTS”
  - Source code generated by a COTS package



## “COTS exemption” busted

Categories	Level of scrutiny	Tested?	Source code/data required?	Coding standards enforced?	Shown to be correct?
COTS	Black box	Yes	No	No	No
Third-party logic, border logic, configuration data	Clear box	Yes	Yes	No	No
Application logic	Coding standards	Yes	Yes	Yes	No
Core logic	Logic verification	Yes	Yes	Yes	Yes

## “COTS exemption” busted

- In some cases, previous certifications of COTS products to equivalent standards may be found to render portions of the test campaign redundant
- All such findings must be documented in the test plan, which is subject to EAC review and approval
- C.f. EAC Decision on Request for Interpretation 2007-05, testing focus and applicability








## Impact of changes

- More precise terms and carefully scoped requirements clarify that nothing is exempt from testing
- Should be little or no change to current test practices
  - Borderline cases have been clarified
- R.I.P. “COTS exemption”

## List of major topics

- Systems versus devices
- Benchmarks and related test methods
- “COTS”
- **Accuracy for optical scanners**
- Post-election reporting requirements

# Categories of optical scan marks

-  Ideal mark (not marginal)
-  Reliably detectable mark (not marginal)
-  Standard, reliably detectable mark (not marginal)
-  **Marginal mark**
-  Hesitation mark (not marginal)
-  Extraneous mark (not marginal)
-  No mark (not marginal)

## 7.7 Counting

- 7.7.5 Accuracy
  - SHALL count reliably detectable marks
  - SHALL count standard mark (MCOS)
  - SHOULD reject marginal marks (PCOS) (→ Part 1 Req. 3.2.2.2-E)
  - SHOULD ignore hesitation marks and imperfections
  - SHALL ignore extraneous marks and non-marks
  
- C.f. Part 2 Req. 4.1.2-A.2 User documentation, reliably detectable marks

## Impact of changes

- Optical scanner requirements caught up to the state of the practice
- Don't expect any changes to products
- A greater range of marks should be tested by test labs
- Manufacturers must disclose what constitutes a reliably detectable mark vs. a marginal mark

## List of major topics

- Systems versus devices
- Benchmarks and related test methods
- “COTS”
- Accuracy for optical scanners
- **Post-election reporting requirements**



## 7.8 Reporting

- 7.8.3 Vote data reports
  - 7.8.3.1 General functionality
    - Accurate, human-readable report of all votes cast
    - Account for all cast ballots and all valid votes
    - No discrepancies / detect, flag, and diagnose discrepancies
    - No tallies before close of polls
  - 7.8.3.2 Ballot counts
  - 7.8.3.3 Vote totals (including overvotes and undervotes)

## Changes

- Clarified and disambiguated reporting requirements
  - Guiding principle: account for all cast ballots and all valid votes
  - Cast vs. read vs. counted
  - Vote counts vs. ballot counts

## Impact of changes

- Expected impact: localized software changes to report generator in some systems
- Worst case impact: if system does not currently keep track of all of the needed data, may need to add some new counters
- C.f. EAC Decision on Request for Interpretation 2007-06, recording and reporting undervotes

## Topics not addressed

- Details of class structure
- Voting variations
- Support for early voting and Electronically-assisted Ballot Markers (EBMs)
- Software workmanship
- Quality assurance and configuration management
- Other revised test methods
  - Logic verification
  - Operating test for humidity
- Requirements carried over unchanged from VVSG 2005

End of presentation



**Bonus Slides**

Deleted Segments

Extras

## Zingers

- Where is the ballot counter requirement?
  - Part 1 Section 4.3.5, "Ballot counter"
  - Design requirement changed to functional requirement
- Where is the 2 hour battery backup requirement?
  - Part 1 Requirement 6.3.4.3-A.4, "Outages, sags and swells"
  - Entire section was rewritten

## Zingers

- How is Ballot On Demand / Ballot Now / etc. covered?
  - Tabulation of the resulting ballots would be tested in the normal course of the test campaign
  - Missing requirement: If a voting device prints ballots, then it shall print ballots that conform to manufacturer specifications

Certain commercial products are identified to foster understanding. Such identification does not imply recommendation or endorsement by the National Institute of Standards and Technology, nor does it imply that the products identified are necessarily the best available for the purpose.



# Plain language vs. precision for intended use: an analogy

TITLE 47—TELECOMMUNICATION

CHAPTER I—FEDERAL COMMUNICATIONS COMMISSION

PART 2—FREQUENCY ALLOCATIONS AND RADIO TREATY MATTERS; GENERAL  
RULES AND REGULATIONS—Table of Contents

Subpart B—Allocation, Assignment, and Use of Radio Frequencies

Sec. 2.106 Table of Frequency Allocations.

NG66 The band 470–512 MHz (TV channels 14–20) is allocated to the broadcasting service on an exclusive basis throughout the United States and its insular areas, except as described below:

Blah blah blah, diddy blah de blah blah, screech yowl arrrr...

# What is a “core” requirement?

- The distinction is an artifact of the subcommittee structure of the TGDC (STS, HFP, CRT)
- All requirements that are not within the scope of work of security or human factors specialists
- Functional requirements of the form “All voting systems shall be able to count votes”
- Reliability and accuracy
- Workmanship

## Strategy for revising core reqs.

- Global changes apply to everything
  - Reorganized the document
  - Identified the requirements
  - Clarified language and used terms consistently
- Requirement changes satisfy three criteria:
  - There is a problem
  - There is a solution
  - The solution is an improvement

## Strategy for revising core reqs. (con't)

- Clarifications
  - Reworded requirements that confused reviewers
  - Where significant dialogue was needed to establish that certain requirements made sense, that dialogue is included as informative text
- Miscellaneous mandates
  - Removed guidance for punchcard technology
  - Process model
  - Public Information Package

## Changes by STS

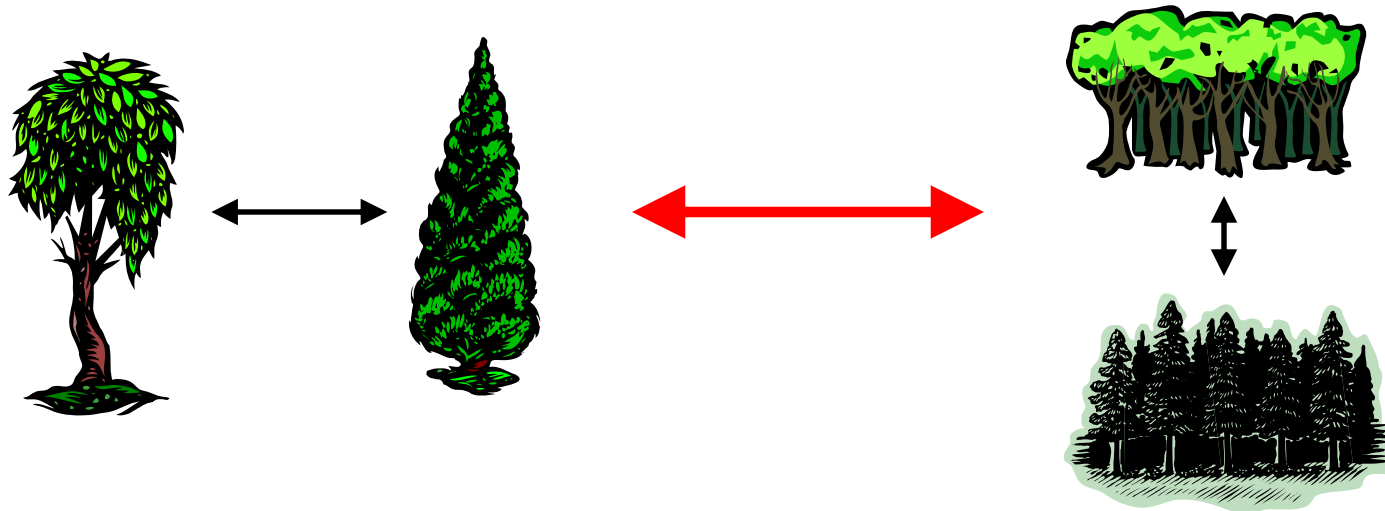
- POC: John Wack
  - Integratability and data export/interchange (a.k.a. common data formats, interoperability)
  - Issuance of voting credentials and ballot activation (a.k.a. e-pollbooks)

# Classes

- **Class:** Identified set of voting systems or voting devices sharing a specified characteristic
- Diagrams in Part 1, Section 2.5.2
- Classes are used to narrow the scope of requirements (Applies-to:)

# Classes

- Classes form a generalization/specialization hierarchy (more formally, a *lattice*)
- The part/whole relationship between systems and devices is separate and different



## Voting system class breakdown

- *Voting system*
- Supported voting variations
  - E.g., *Straight party voting*: voting systems that support straight party voting
- *IVVR*



# Voting device class breakdown

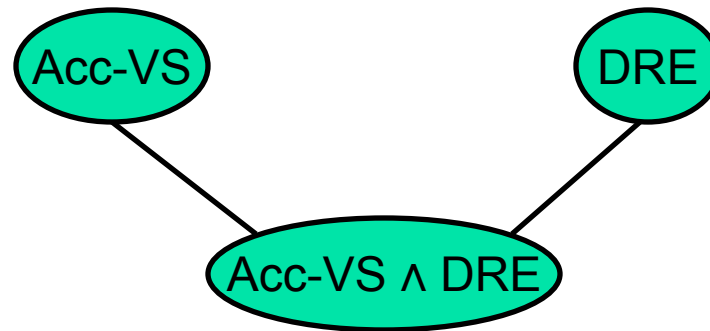
- *Voting device*
- Supported voting variations
  - E.g., *Straight party voting device*: voting devices that support straight party voting
- Commonly understood device categories
  - *DRE, Optical scanner, etc.*
- Generalizations
  - *Vote-capture device, Tabulator, Paper-based device, etc.*
- Other important concepts
  - *IVVR vote-capture device, Acc-VS, VVPAT*

Do not assume that classes are  
mutually exclusive

Acc-VS

DRE

# Do not assume that classes are mutually exclusive



Accessible voting stations that happen to be DREs, or DREs that are also accessible voting stations

## Why it makes sense

- Certain requirements apply to DREs
- Certain requirements apply to accessible voting stations
- A device that is both must satisfy *both* sets of requirements
- Inheritance minimizes repetition

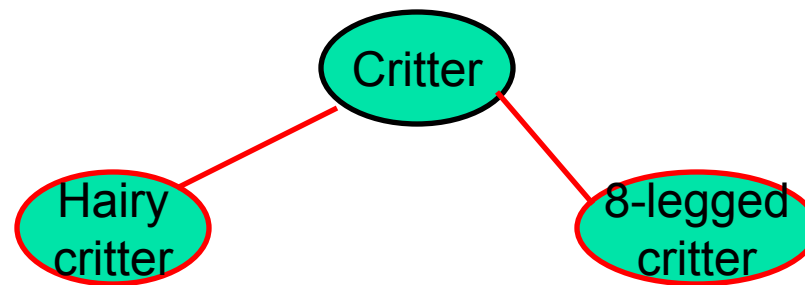
# Concept analysis by example



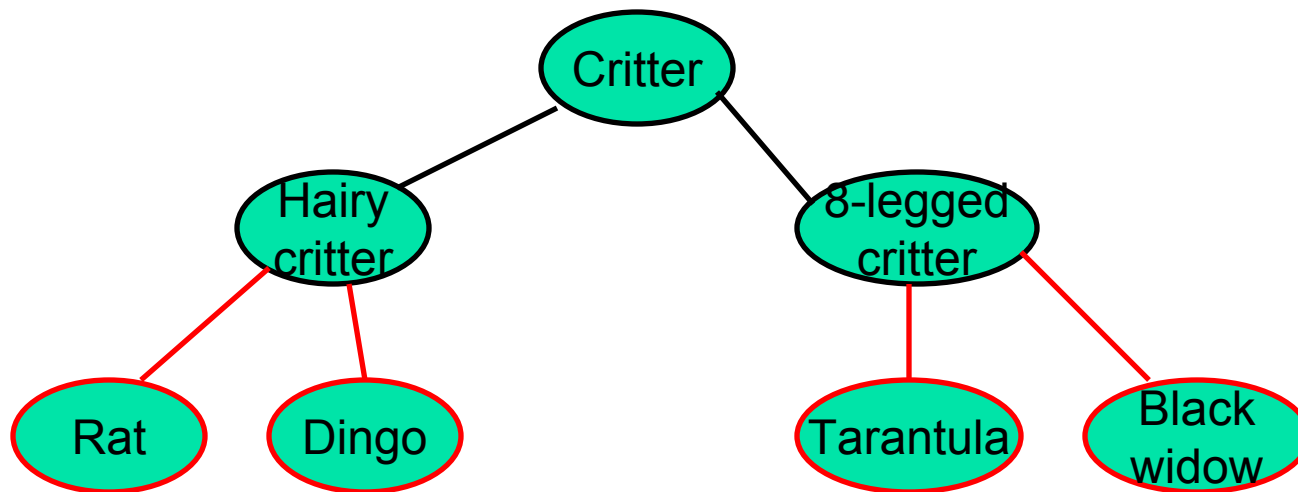
# Concept analysis by example

Critter

# Concept analysis by example



# Concept analysis by example

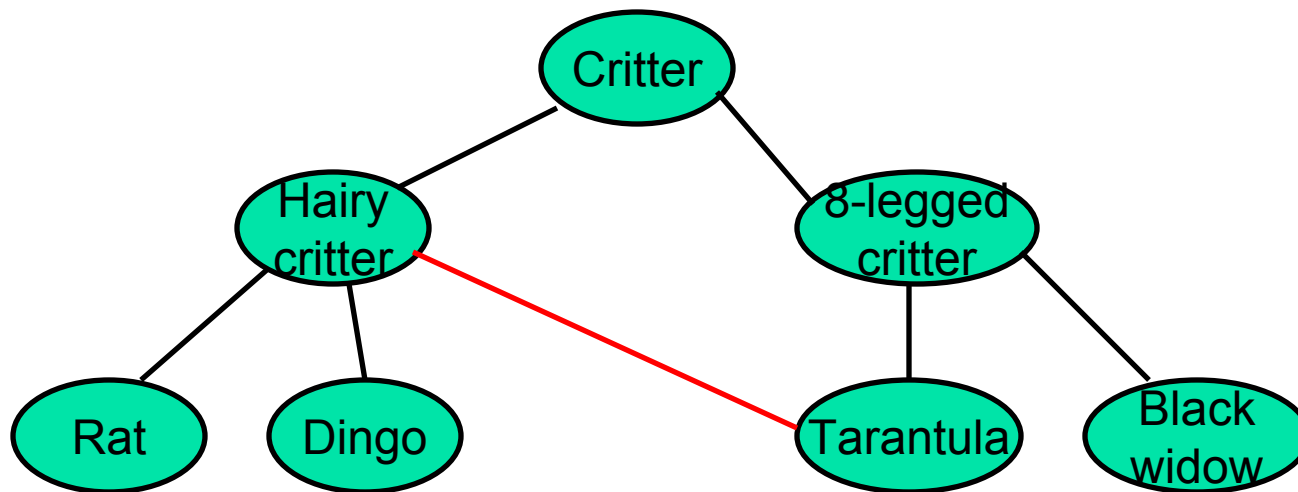


All rats are hairy critters.  
All dingos are hairy critters.

All tarantulas are 8-legged critters.  
All black widows are 8-legged critters.



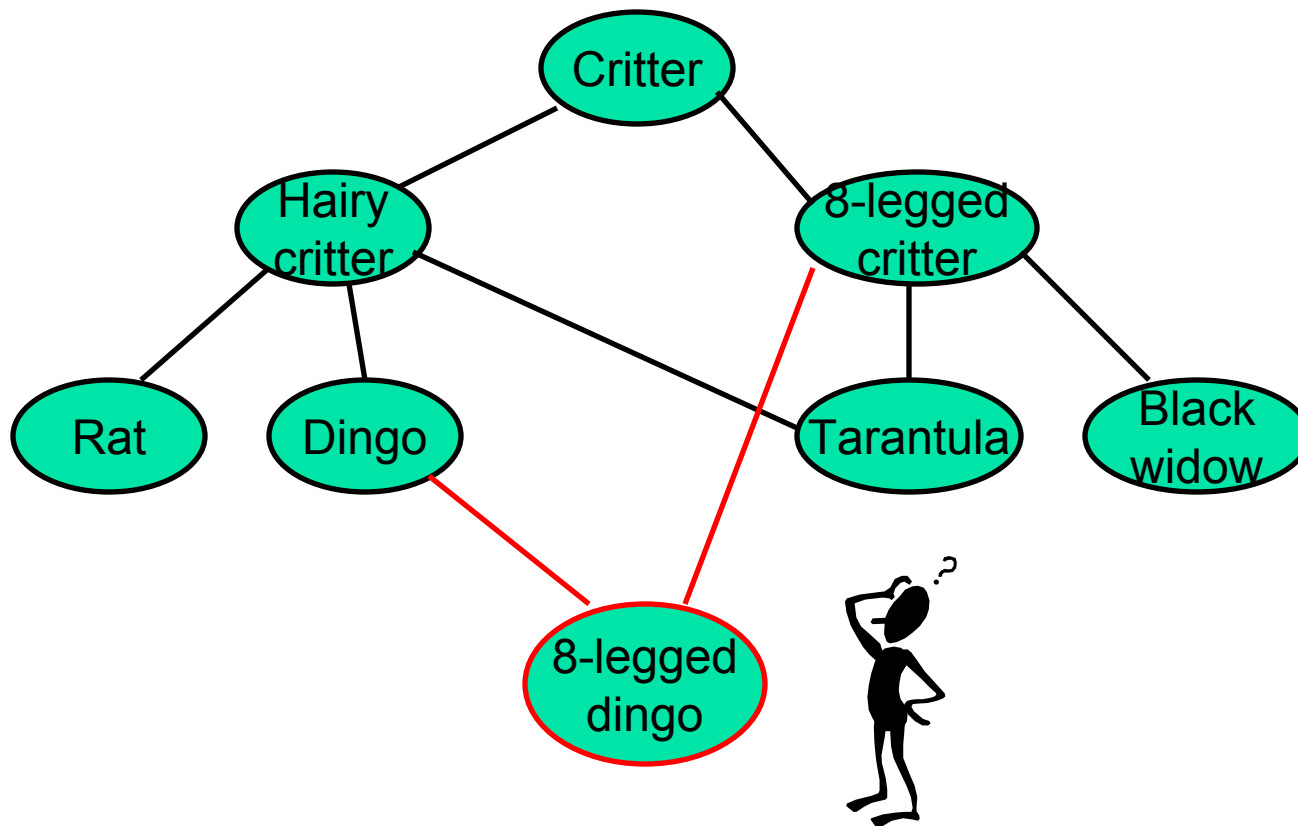
# Concept analysis by example



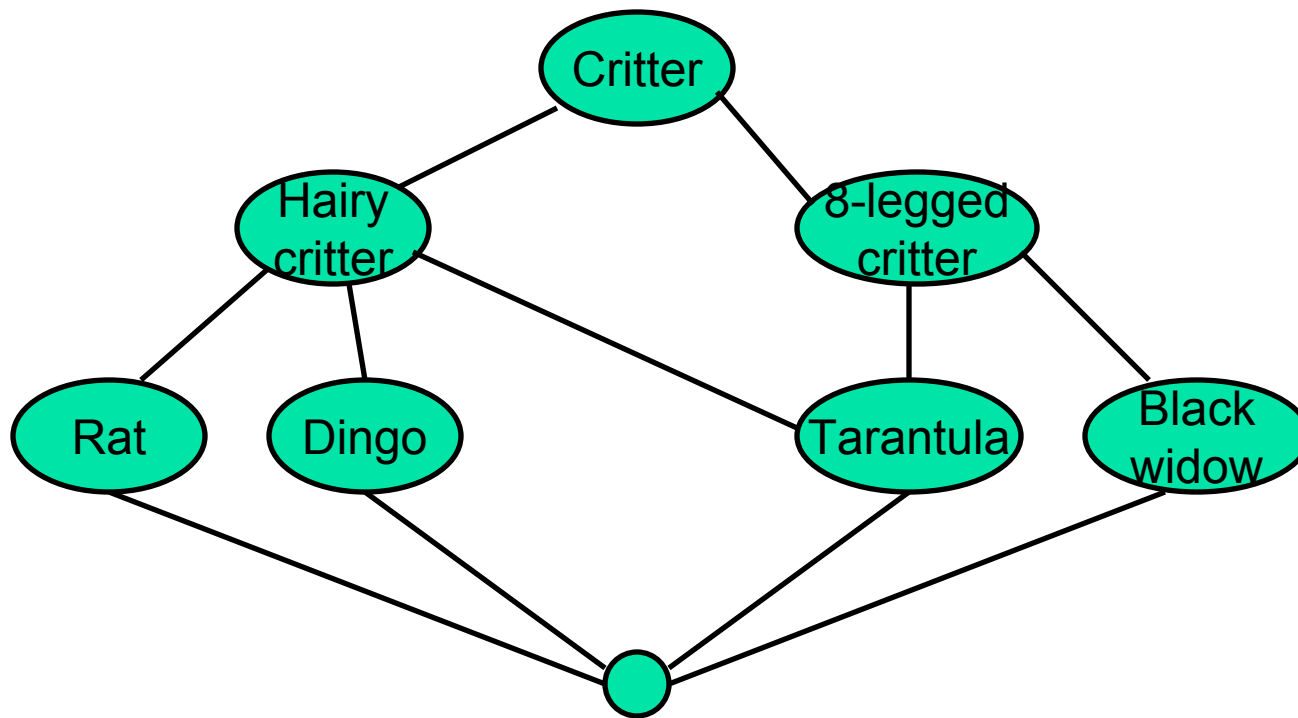
All tarantulas are hairy, 8-legged critters.

**Do not assume that classes are mutually exclusive**

# Concept analysis by example



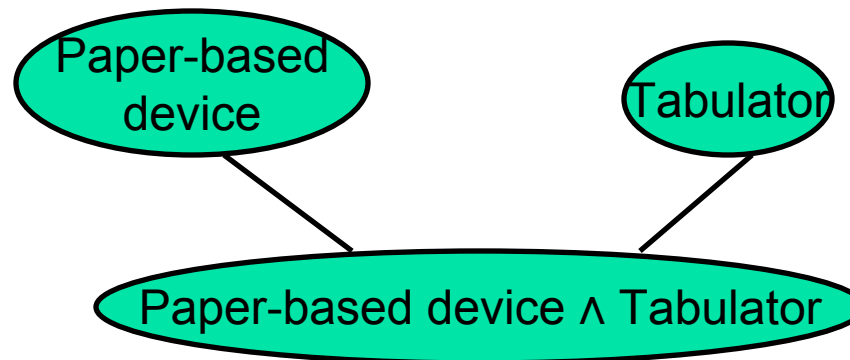
# Concept analysis by example



## Semantics of classes Part 1 Section 2.5.4

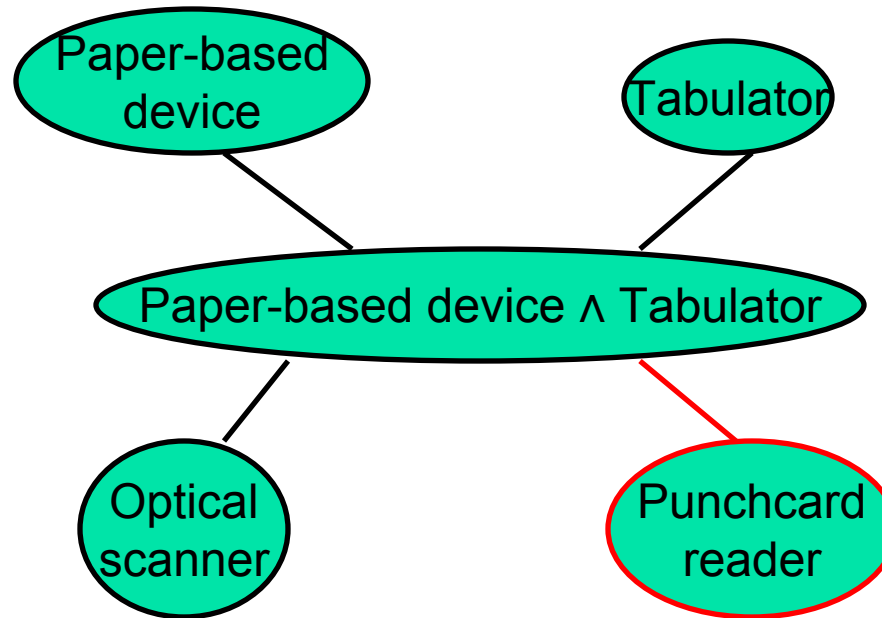
- Under attack due to mathematical language
- Section 2.5.4 just defines in mathematical language what is described in plain language in Section 2.5.2
- Technically oriented readers (“language lawyers”) want formality to resolve the ambiguities of plain language
- Removing the mathematical language would reduce the precision and completeness of the specification
- Not to be confused with Section 8.3 (Logic Model)

# Quiz time



Is the subclass Paper-based device  $\wedge$  Tabulator  
any different from Optical scanner?

# Quiz time



Not at the moment... But the equivalence could be temporary.



## Be careful

- “Generally understood terms” have more conflicting interpretations than generally believed
- Terms appearing in requirements are defined in Appendix A
- Quite often, issues go away when the intended meanings of terms are understood

## Terms

- General: *Vote-capture device, Tabulator, Programmed device, ...*
- Specific: *DRE, EBM, PCOS, EMS, ...*
- These all have definitions in Appendix A



# Terms

- **Tabulator:** Programmed device that counts votes
  - There is a whole set of requirements that applies to any device that counts votes
  - No hair-splitting between counting vs. aggregating
  - DREs are tabulators
  - EMSs are tabulators

# Terms

- VEBD = voter-editable ballot device
  - All vote-capture devices that let you back up and change your votes without starting over
  - An abstraction that has relevant requirements to be inherited by many subclasses
  - Comes in audio (VEBD-A) and video (VEBD-V) flavors
  - An Acc-VS (accessible voting station) must satisfy both VEBD-A and VEBD-V

# Terms

- EBM = electronically-assisted ballot marker
  - VEBD
  - When finished, print filled-in paper ballot
  - Radically different implementations: DRE-style vs. vote-by-phone
- EBP = electronic ballot printer
  - Special case of EBM
  - Does not require you to feed in a blank ballot
  - Activates whichever ballot style is needed
  - When finished, print that on blank ballot stock

## Flavors of optical scanner

- PCOS: precinct-count optical scanner
  - Low volume
  - Interacts directly with voters
  - Opportunity to reject ambiguous ballots
- CCOS: central-count optical scanner
  - High volume
  - Operated by central election officials
  - Ambiguous ballots must be arbitrated somehow
- A given optical scanner might be configurable to support both sets of requirements

## Flavors of optical scanner

- MCOS: MMPB-capable optical scanner
  - MMPB = manually-marked paper ballots
- ECOS: EMPB-capable optical scanner
  - EMPB = EBM-marked paper ballot
- Differences
  - MCOS required to handle the range of manual marks
  - ECOS must be able to raise alarm if marginal marks or overvotes are detected (equipment malfunction)
- Likely every optical scanner will support both

# Next VVSG Training

## Chapter 6: General Core Reqs.

December 2007

David Flater

National Institute of Standards and Technology

dflater@nist.gov

## Chapter 6: General Core Reqs.

- 6.1 General Design Requirements
- 6.2 Voting Variations
- 6.3 Hardware and Software Performance
- 6.4 Workmanship
- 6.5 Archival[ness] Requirements
- 6.6 Integratability and Data Export/Interchange
- 6.7 Procedures required for correct system functioning

# Let's get this out of the way

## 6.7 Procedures required for correct system functioning

The requirements for voting systems are written assuming that these procedures will be followed.

Follow instructions: The voting system must be deployed, calibrated, and tested in accordance with the voting equipment user documentation provided by the manufacturer.



## 6.1 General Design Requirements

- General principles
  - No obvious fraud
  - Verifiable vote recording and tabulation
  - Minimum devices included
- Misc. design requirements carried over
  - Paper ballots
  - Card holders
  - Ballot boxes
  - Activity indicator
  - Operable in a polling place

## 6.2 Voting Variations

- If the manufacturer claims that the voting *system* supports feature  $X$ , then it must contain *devices* that support feature  $X$
- Requirements in later sections specify what  $X$  means at the device level

# Voting variations



## Counting logic

- All voting systems must support
  - 1-of-M voting (vote for [not more than] one)
  - Yes/no questions
- Classes for optional features
  - N-of-M voting
  - Cumulative voting
  - Ranked order voting
  - Straight party voting
    - Cross-party endorsement

## Ballot handling

- In-person voting
- Absentee voting
- Provisional/challenged ballots
- Review-required ballots

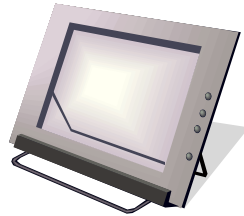
## Other features

- Write-ins
- Ballot rotation
- Primary elections
  - Closed primaries
  - Open primaries
- Split precincts



## A careful distinction

### *Write-ins*



MICKEY MOUSE (write-in) ... 275  
... other write-ins ...

---

### *Review-required ballots*



- ← Counted ballots
- ← Outstacked ballots containing write-ins

WRITE INS ... 300

## Impact of changes (voting variations)

- This was purely a matter of defining things that were left unspecified in the old Guidelines
- Manufacturer claims of support for feature  $X$  now invoke specific requirements
- All of these features are optional in the Guidelines
  - States may opt to require certain ones in an RFP
- Manufacturers have a choice
  - Satisfy the requirements
  - Modify the claim



## 6.3 Hardware and Software Performance

- “Benchmarks”
  - 6.3.1 Reliability
  - 6.3.2 Accuracy
  - 6.3.3 Misfeed rate
- Electromagnetic compatibility
  - 6.3.4 Electromagnetic immunity
  - 6.3.5 Electromagnetic emission limits
  - 6.3.6 Dielectric withstand



## Another careful distinction

- Failure as used for reliability benchmarking versus “failing” a conformance test or generic “failure” to meet expectations
- A system can “fail” a test even if there is no operational failure
  - Operation completed without failure but did the wrong thing
  - Part 3 Req. 5.2.1.2-D: A demonstrable violation of any applicable requirement of the VVSG during the execution of any test SHALL result in a test verdict of Fail.
- The reliability benchmark deals specifically with operational failures

## Old benchmarks

- Failure rate (reliability)
  - 163 hours MTBF widely condemned
  - Rationale supported neither the benchmark nor the test—confusion of non-comparable values
  - Tested: 90 % > 45 hours
  - Time-based benchmark considered less appropriate than a volume-based benchmark
- Misfeed rate
  - Two different benchmarks, no specific test

## Old benchmarks

- Error rate (accuracy)
  - Untestable benchmarks on low-level operations
  - Conflation of ballot positions with votes
  - Model assumes maximum one error per ballot position
  - Unclear how inaccuracies in ballot counts and totals of undervotes and overvotes factor in
  - Upper benchmark (1 in 10 000 000) considered arbitrary, possibly unattainable by paper-based systems
  - Tested: 95 % > 1 / 500 000

## Old benchmarks

Benchmark	P(failure) in 15 hours	Min. test time (if benchmark used as lower test MTBF)	w/ 7.1 days
45 hours (VVSG'05 II.C.4)	28 %	169 hours (7.04 days)	1 device
135 hours (VVSG'05 II.C.4)	11 %	21.1 days	3 devices
163 hours (VVSG'05 I.4.3.3)	8.8 %	25.5 days	4 devices

$$p = 1 - e^{-15/\theta}$$

$$t \geq 3.75 \times \theta_1$$

## What happens if we just raise the number

Benchmark	From	P(failure) in 15 hours	Min. test	w/ 7 days	w/ 100 devices
1500 hours	[1]	1.0 %	234 days	34 devices	2.34 days
5000 hours	[2]	0.30 %	2.14 years	112 devices	7.82 days
15000 hours	[3]	0.10 %	6.42 years	335 devices	23.4 days

[1] IEEE Draft 5.3.1 ballot comment (later increased to 15000)

[2] VVSG'05 public comment #2056, lower bound

[3] VVSG'05 public comment #2056, upper bound

## Report total error rate

- Need a definition of error that allows them to be counted
  - Errors must be observable
  - Given test report, tell me how many errors were made
- It's not as simple as previously believed

# Report total error rate

Part 3 Req. 5.3.4-B (generalized from 1990 VSS F.6)

- **Report item:** Any one of the numeric values (totals or counts) that must appear in any of the vote data reports. Each ballot count, each vote, overvote and undervote total for each contest, and each vote total for each contest choice in each contest is a separate report item.
- **Report item error:** Absolute value of the difference between the correct value and the reported value.
- **Report total error:** Sum of all of the report item errors.
- **Report total volume:** Sum of all of the *correct* values.
- **Report total error rate** = report total error / report total volume

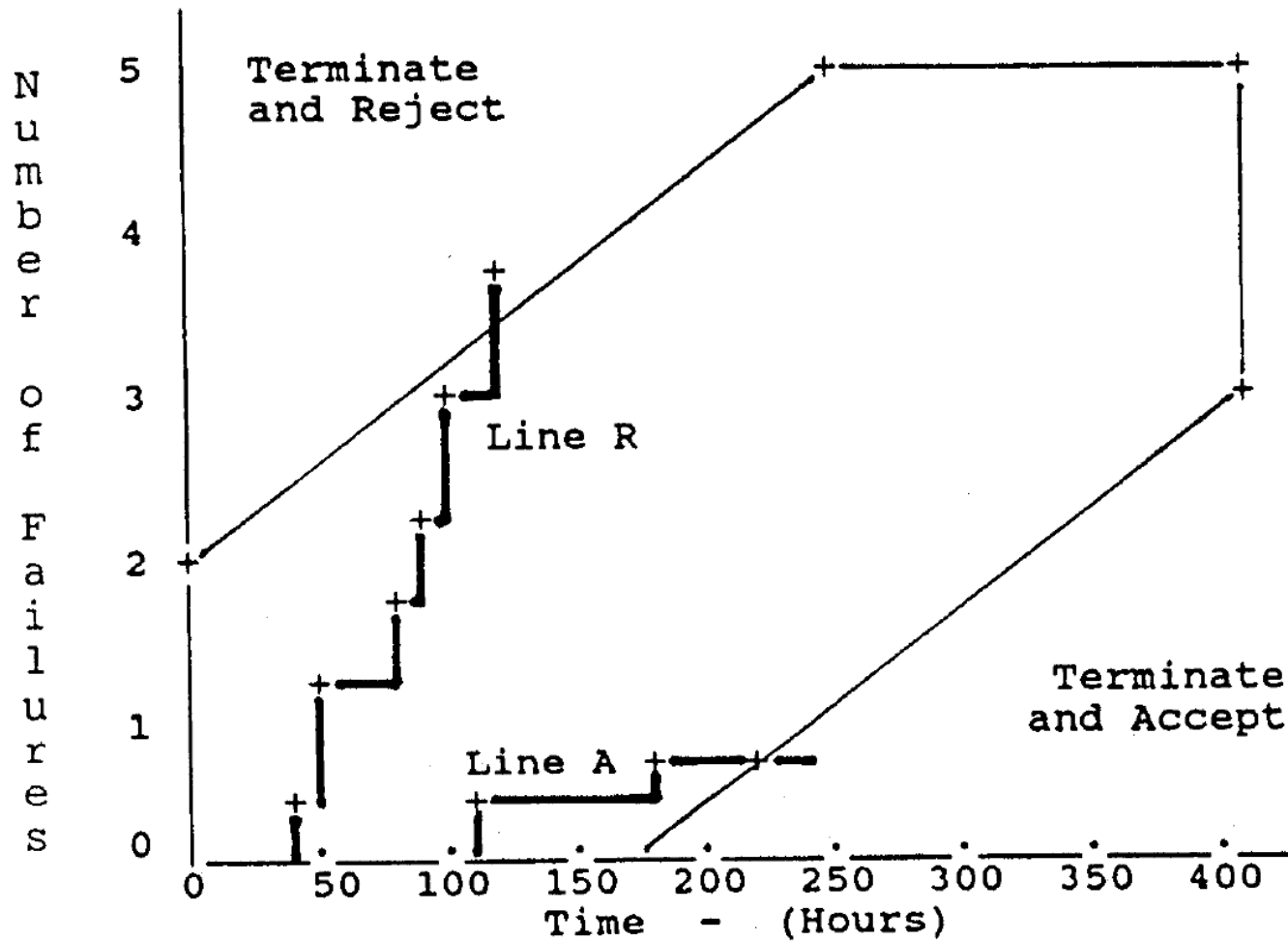


# The old test method



# Probability Ratio Sequential Test (PRST)

- A.k.a. Sequential Probability Ratio Test (SPRT)
- VVSG follows recipes from MIL-HDBK-781A
- Originally designed for munitions testing
- Simultaneously
  - Obtain statistically significant result
  - Minimize the length of the test (number of bombs used)
  - Avoid bias

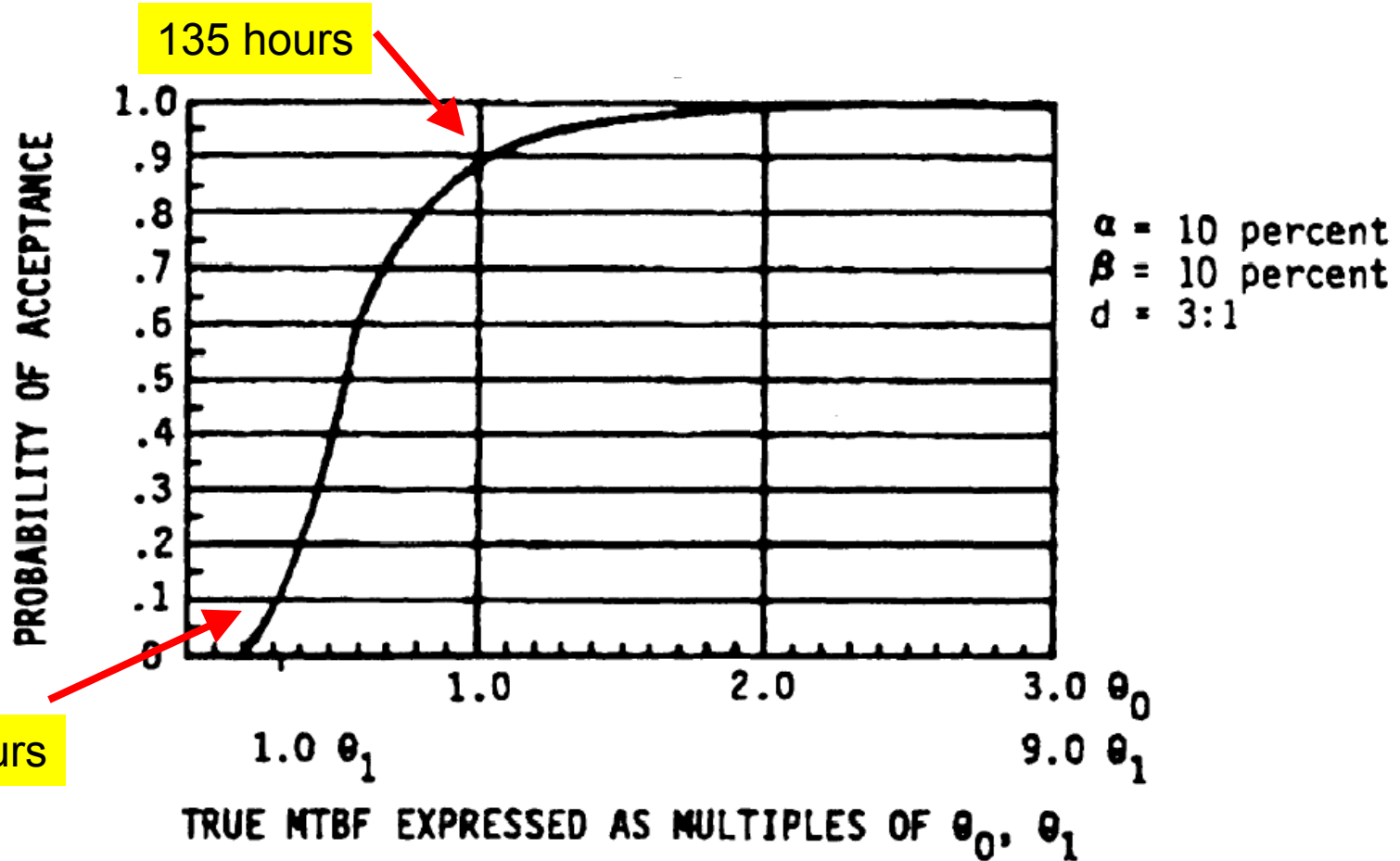


# Improving U.S. Voting Systems

- NIST activities supporting the Help America Vote Act

NIST

National Institute of  
Standards and Technology



## Whither 163 hours?

- “The MTBF demonstrated during certification testing shall be at least 163 hours.” (VVSG’05 I.4.3.3)
- In 1990 and 2002 VSS, minimum duration of Probability Ratio Sequential Test to demonstrate  $MTBF \geq 45$  hours with the given parameters was calculated to be 163 hours
  - In VVSG’05, this was revised to 169 hours
- MTBF actually demonstrated is a different number, and it varies
  - Ranges from 44 hours to 73 hours (at 90 % confidence)
- Confusion of non-comparable values

## Whence 45 hours?

- “A typical system operations scenario consists of approximately 45 hours of equipment operation, consisting of 30 hours of equipment set-up and readiness testing and 15 hours of elections operations.” (VVSG’05 I.4.3.3)
- With MTBF = 45 hours, 63 % of our machines die by the end of the election (after 45 hours)
- More confusion of non-comparable values

# Issues applying PRST

- PRST assumes that we can adjust the length of the testing as it specifies
- Ignore failures occurring in other portions of the test campaign?
- Somehow make the length of the test campaign coincide with what PRST specifies?

## Old test not end-to-end

- VVSG'05 permits test labs to bypass portions of the system that would be exercised during an actual election (VVSG'05 II.1.8.2.3)
  - "May use a simulation device... provided that the simulation covers all voting data detection and control paths that are used in casting an actual ballot."
  - But... "In the event that only partial simulation is achieved, then an independent method and test procedure shall be used to validate the proper operation of those portions of the system not tested by the simulator."
  - Also: "For systems that use a light source as a means of detecting voter selections, the generation of a suitable optical signal by an external device is acceptable."
- If the test is compromised, the number of ballot positions tested is of little relevance
  - "One flight test is worth a thousand simulations" – Henry Spencer



# The new test method

## Part 3 Section 5.3



# Classical hypothesis testing

- Collect data throughout the entire test campaign
- At the end, analyze the data and determine what they demonstrate
- Length of testing is fixed in advance (the approved test plan)
  - Specific length is not critical, but discretionary stopping introduces bias

# Terms

- “Event” = failure, error, or misfeed, as applicable
- “Volume” = e.g., # ballots (defined by the benchmark)
- Hypothesis  $H_0$ : The system is conforming
- Critical value  $v_c$ : “Cutoff volume”
  - Calculated based on benchmark event rate  $r_b$  and number of observed events  $n_o$
  - $v_c$  is the minimum volume at which it would not be “unusual” for  $n_o$  or more events to occur in a marginally conforming system
    - “Unusual” = less than 10 % chance
  - If tested volume  $v_o$  is at least  $v_c$  you pass
- Critical value  $r_d$ : “Demonstrated rate”
  - $r_d$  may be greater or less than the benchmark event rate  $r_b$

## Possible outcomes

- Fail
  - Demonstrated nonconformity with 90 % confidence ( $v_o < v_c$ )
- Pass
  - Strong pass: Demonstrated conformity with 90 % confidence ( $r_d \leq r_b$ )
  - Weak pass: Inconclusive, pass by default ( $r_d > r_b$  and  $v_o \geq v_c$ )
- Longer testing makes an inconclusive result less likely, but they can never be prevented entirely
  - Impossible to demonstrate conformity to benchmark of zero

## Example

- Benchmark event rate  $r_b = 10^{-2}$
- Observations
  - Tested volume  $v_o = 600$
  - Observed events  $n_o = 3$

## How-to: Part 3 Section 5.3.2

- Cutoff volume  $v_c$ 
  - $P(2, r_b v_c) = 0.9$
  - $v_c = 1.102065 / 10^{-2}$   
 $= 110.2065$
  - $600 > 110.2065$
  - Pass
- Demonstrated rate  $r_d$ 
  - $P(3, r_d v_o) = 0.1$
  - $r_d = 6.680783 / 600$   
 $= 1.113464 \times 10^{-2}$
  - Not a strong pass ...

$n$	$rv$ satisfying $P(n,rv)=0.1$	$rv$ satisfying $P(n,rv)=0.9$
2	5.322320	1.102065
3	6.680783	1.744770

## 6.4 Workmanship

- 6.4.1 Software engineering practices
- 6.4.2 Quality assurance and configuration management
- 6.4.3 General build quality
- 6.4.4 Durability
- 6.4.5 Maintainability
- 6.4.6 Temperature and humidity
- 6.4.7 Equipment transportation and storage

## 6.4.1 Software engineering practices

- 6.4.1.1 Scope
- 6.4.1.2 Selection of programming languages
- 6.4.1.3 Selection of general coding conventions
- 6.4.1.4 Software modularity and programming
- 6.4.1.5 Structured programming
- 6.4.1.6 Comments
- 6.4.1.7 Executable code and data integrity
- 6.4.1.8 Error checking
- 6.4.1.9 Recovery



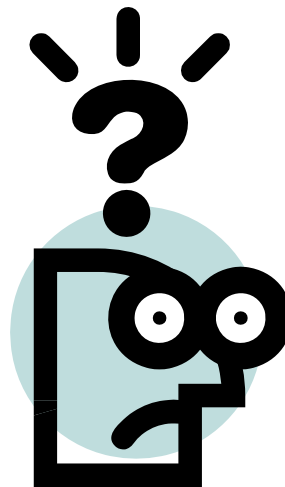
## Executive summary

- Manufacturers are expected to use current best practices for software engineering
  - “Published” and “credible” coding conventions
  - Three year rule and reassessments
- Worst practices are prohibited
  - I.e., practices that are known risk factors for latent software faults and unverifiable code
- Defensive programming is required
- Use of state-of-the-art programming languages and standards facilitates compliance

## Impact of changes

- Resolved controversy over prescriptive requirements on programming style
- More flexibility for manufacturers
- Pressure to migrate to state-of-the-art programming languages and standards
- Should get more reliable, higher integrity software
- Costs
  - Legacy code must be cleaned up and reinforced to meet the same requirements
  - More experience and judgment required of test labs

# Warning: Programming jargon



## 6.4.1 Software engineering practices

- 6.4.1.1 Scope
  - Application logic
  - Not COTS, third-party logic, or border logic
- 6.4.1.2 Selection of programming languages
  - High-level language with support for structured programming—which includes block-structured exception handling
  - Compromise—C can be retrofit with extension package to add exceptions
  - Interpreted languages like Java are allowed

## 6.4.1 Software engineering practices

- 6.4.1.3 Selection of general coding conventions
  - Workmanship, security, integrity, testability, maintainability
  - “Published”
  - “Credible”
- 6.4.1.4 Software modularity and programming
  - Orthogonality of design
  - Size limit

## 6.4.1 Software engineering practices

- 6.4.1.5 Structured programming
  - Unstructured programming is forbidden
- 6.4.1.6 Comments
  - Optional; defer to coding conventions
- 6.4.1.7 Executable code and data integrity
  - Risky practices forbidden
  - Protect against tampering
  - Monitor transfer quality of I/O operations (e.g., burning coasters)

## 6.4.1 Software engineering practices

- 6.4.1.8 Error checking
  - Defend against garbage input from outside
  - Prevent buffer overflows, numeric overflows, stack overflows...
  - Validate inputs to each unit
  - Masking errors is prohibited
  - Diagnostics and health monitoring
  - Detect or prevent violations of election integrity (e.g., accumulation of negative votes)

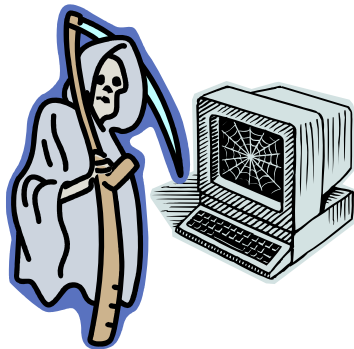
## 6.4.1 Software engineering practices

- 6.4.1.9 Recovery
  - Protect integrity of all recorded votes and audit log information
  - Controlled recovery to known good state
  - Allows diagnostic state prior to recovery



## 6.5 Archival[ness] requirements

- Records last at least 22 months in temperatures up to 40 °C and humidity up to 85 %



## Related requirements

- Part 2 Req. 4.4.8-C Operations manual, procedures to ensure archivalness
  - The manufacturer SHALL detail the care and handling precautions necessary for removable media and records to last 22 months etc.
- Part 3 Req. 4.1-B Review of COTS suppliers' specifications
  - Test lab shall verify that the media are not being used out-of-spec

## Impact of changes (archivalness)

- Responsive to complaints about thermal paper going off
- Ambient conditions specified
  - End users should not have to resort to extreme measures to preserve records for the statutory period
- More test lab scrutiny of data sheets for media used
  - Actually supposed to last 22 months in ambient conditions

# Next VVSG Training

## Chapter 7: Reqs. by Voting Activity

December 2007

David Flater

National Institute of Standards and Technology

dflater@nist.gov

## 7 Requirements by Voting Activity

- 7.1 Election Programming
- 7.2 Ballot Preparation, Formatting, and Production
- 7.3 Equipment Setup for Security and Integrity [L&A testing]
- 7.4 Opening Polls
- 7.5 Casting
- 7.6 Closing Polls
- 7.7 Counting
- 7.8 Reporting

## 7.1 Election Programming

- The EMS shall support election definition
  - Definition of political and administrative subdivisions, a.k.a. reporting contexts, corresponding to precincts, districts, etc.
  - Ballot definition (contests and candidates)
- The EMS shall support all of the claimed voting variations
- The EMS shall provide for the distribution of these definitions to the rest of the system as needed

## 7.2 Ballot Preparation, Formatting, and Production

- The EMS shall support the definition of ballot configurations and ballot styles
  - Subject to many subrequirements
- The EMS shall provide for the distribution of these definitions to the rest of the system as needed

## Terms

- **Ballot configuration:** Set of contests in which voters of a particular group (e.g., political party and/or election district) are entitled to vote
- **Ballot style:** Concrete presentation of a particular ballot configuration
- A given ballot configuration may be realized by multiple ballot styles, which may differ in the language used, the ordering of contests and contest choices, etc.



## Why this makes sense

- Most functional requirements on vote capture and tabulation are independent of style issues such as which ballot position a given contest choice appears in or which language it is written in
- Nevertheless, there are functional requirements on the capability of the voting system to produce and tabulate different ballot styles
  - Auto-layout capability
  - Association of ballot styles with political parties in primary elections
  - Correct mapping from ballot positions back to contest choices
- So we need both terms

## 7.3 Equipment Setup for Security and Integrity [L&A testing]

- “The purpose of logic and accuracy testing is to detect malfunctioning and misconfigured devices before polls are opened. It is not a defense against fraud.”
- Built-in self-test and diagnostics
- Integrity checks on ballot styles and software
- Able to run test ballots
- Calibrations
- No side-effects
- Readiness reports (→ Part 1 Section 7.8.2)

## 7.4 Opening Polls

- Designated functions to open polls
- Guard against accidental or unauthorized poll opening
- Guard against doing things in the wrong order
- Check that L&A testing was done
- Check that L&A test result was “OK”
- Leave no ambiguity whether polls are currently open

## 7.5 Casting

- 7.5.1 Issuance of voting credentials and ballot activation
- 7.5.2 General voting functionality
- 7.5.3 Voting variations
- 7.5.4 Recording votes
- 7.5.5 Redundant records
- 7.5.6 Respecting limits
- 7.5.7 Procedures required for correct system functioning

## 7.5 Casting

- 7.5.2 General voting functionality
  - Capture votes
  - No advertising
- 7.5.3 Voting variations
  - Support all of the claimed voting variations
- 7.5.4 Recording votes
  - Record them correctly
  - All records shall be consistent with the feedback to the voter
  - Cast is committed

## 7.5 Casting

- 7.5.5 Redundant records
  - DREs shall record and retain at least two machine-countable copies of each cast vote record on physically separate media
- 7.5.6 Respecting limits
  - If the next ballot could overflow a counter, STOP
- 7.5.7 Procedures required for correct system functioning
  - One voter, one ballot
  - Assignment of the correct ballot style
  - Prevent tampering
  - Early voting considerations

## 7.6 Closing Polls

- Designated functions to close polls
- No access to cast vote records before close of polls
- No voting after close of polls
- No reopening of polls
- Post-election reports (→ Part 1 Section 7.8.3)

## 7.7 Counting

- 7.7.1 Integrity
- 7.7.2 Voting variations
- 7.7.3 Ballot separation
- 7.7.4 Misfed ballots
- 7.7.5 Accuracy
- 7.7.6 Consolidation
- 7.7.7 Procedures required for correct system functioning



## 7.7 Counting

- 7.7.1 Integrity
  - Detect and prevent ballot style mismatches
  - Optical scanners must deal with ballots that are oriented incorrectly
- 7.7.2 Voting variations
  - Support all of the claimed voting variations
- 7.7.3 Ballot separation
  - For CCOS, outstacking of ballots on various conditions
  - For PCOS, separation of ballots containing write-ins
  - ECOS react to marginal marks and overvotes as equipment malfunctions

## 7.7 Counting

- 7.7.4 Misfed ballots
  - Ability to clear misfeed
  - **Indicate status of misfed ballot**
- 7.7.5 Accuracy
  - Special cases for optical scanners
  - General benchmark handles everything else

## 7.7 Counting

- 7.7.6 Consolidation
  - Single report for polling place
  - 5-minute requirement for DREs
- 7.7.7 Procedures required for correct system functioning
  - Clearing ballots that get stuck between the reader and the ballot box

## 7.8 Reporting

- 7.8.1 General reporting functionality
- 7.8.2 Audit, status, and readiness reports
- 7.8.3 Vote data reports
- 7.8.4 Procedures required for correct system functioning

## 7.8 Reporting

- 7.8.1 General reporting functionality
  - Timestamps
  - Reporting is non-destructive
- 7.8.2 Audit, status, and readiness reports
  - Pre-election reports = record of the election definition
  - Status and readiness reports = record of conditions at open of polls
  - Able to turn event logs into reports

## 7.8 Reporting

- 7.8.4 Procedures required for correct system functioning
  - Ballot accounting
  - Label unofficial reports as unofficial

# Next VVSG Training

## Chapter 8: Reference Models

December 2007

David Flater

National Institute of Standards and Technology

dflater@nist.gov

# Process Model (Pt. 1 Sec. 8.1)

- Informative
- Provides context for product requirements
- Not directly used by any product requirements



# Vote-Capture Device State Model (Pt. 1 Sec. 8.2)

- Informative, but important
- Clarifies concepts for early voting
  - Overnight suspension of early voting is not the same as closing the polls
- Distinguishes “activated” (active period) from “in use” (voting session)
- Referenced by access control requirements

## Logic Model (Pt. 1 Sec. 8.3)

- Normative
- Rigorously defines the correct results from counting votes
- Referenced by reporting requirements (shall report the following results)
- Used in logic verification
  - Higher level of assurance than operational testing alone

## Counting logic as modelled

- Cumulative voting
  - N-of-M voting = special case with at most 1 vote per contest choice
    - 1-of-M voting = special case with  $N = 1$
    - Yes/no question = special case with yes/no as the only choices
- Ranked order voting not handled; see Part 1 Section 7.7.2.5



# Related requirements

- Part 1 Req. 6.1-B Verifiably correct vote recording and tabulation
- Part 1 Req. 6.3.2-A Satisfy integrity constraints
- Documentation requirements
  - Part 2 Req. 3.4.7.2-F TDP, inductive assertions
  - Part 2 Req. 3.4.7.2-G TDP, high-level constraints
  - Part 2 Req. 3.4.7.2-H TDP, safety of concurrency
- Part 3 Section 4.6 Logic Verification

# Logic verification

## Part 3 Section 4.6



# What is logic verification?

- Formal characterization of software behavior within a carefully restricted scope
- Proof that this behavior conforms to specified assertions (i.e., votes are reported correctly in all cases)
- Complements [falsification] testing
- C.f. “inductive assertions,” “Hoare logic,” “program proving”

# Motivation

- TGDC Resolution #29-05, "Ensuring Correctness of Software Code"
- Higher level of assurance than operational testing alone
- Clarify objectives of source code review

## How it works

- Manufacturer specifies pre- and post-conditions for each callable unit
- Manufacturer proves assertions regarding tabulation correctness
- Testing authority reviews, checks the math, and issues findings
  - Pre- and post-conditions correctly characterize the software
  - The assertions are satisfied



# Compromise #1

- Scope of verification limited to core logic
- **Core logic:** Subset of application logic that is responsible for vote recording and tabulation
- Limited scope = limited assurance; unlimited scope = impracticable

## Compromise #2

- Programming language does not have formally specified semantics
- A formal proof cannot be mandated
- Do what is feasible
  - Formality where possible
  - Informal arguments where not
  - Limitations on complexity to make correctness intuitively obvious
- Still better than operational testing alone

# Impact

- Another document for manufacturers to produce
- Skill level: computer science undergraduate
- Higher level of assurance than operational testing alone
- Possible fear, loathing, claims of infeasibility
  - This is not Common Criteria EAL 7
  - This is not the general case (arbitrary software)
  - Limited both in scope and in rigor

# Logic verification non-issue

- Rice's theorem: In the general case (arbitrary software), nontrivial properties are undecidable
- *This is not the general case*
- Vote counting uses very simple math and logic
- All voting system designs must preserve the ability to demonstrate that votes will be counted correctly

## Part 2 Documentation Requirements

- 1 Introduction
- 2 Quality Assurance & Configuration Management
- 3 Technical Data Package
- 4 Voting Equipment User Documentation
- 5 Test Plan
- 6 Test Report
- 7 Public Information Package

# Quality Assurance and Configuration Management

- Next VVSG
  - ISO 9000/9001 (QA) and ISO 10007 (CM) standards provide the framework for the requirements
  - Manufacturer must deliver a well defined Quality Manual detailing how the processes and procedures required by the VVSG are being implemented

## Part 3 Testing Requirements

- 1 Introduction
- 2 Conformity Assessment Process
  - Informative description
  - General requirements on test labs, ground rules
  - COTS validation, initial and final builds
- 3 Introduction to General Testing Approaches
- 4 Documentation & Design Reviews
  - Inspections
- 5 Test Methods
  - Operational tests
  - OEVT

## Definition Guidelines (ISO 10241)

- A definition shall contain, or start with, the nearest superordinate term.
- A definition shall only contain delimiting characteristics. Other information should be put in an informative note.
- Use already defined terms as components in the definitions.
- A definition shall not be too narrow or too broad.
- Avoid circular definitions.

Summarized from "Introduction to Terminology," Bernd G. Wenzel, EuroSTEP GmbH, 1997.



## Definition Guidelines (ISO 10241)

- A definition shall not begin with an expression such as “term used to describe ....”
- Unless there is a specific reason, a definition shall not begin with an article.
- A definition shall have the same grammatical form (e.g., verb, adjective, or noun) as the term. The grammatical form shall be indicated whenever there is a risk of misunderstanding.
- Unless there is a specific reason, a definition shall consist of a single phrase.