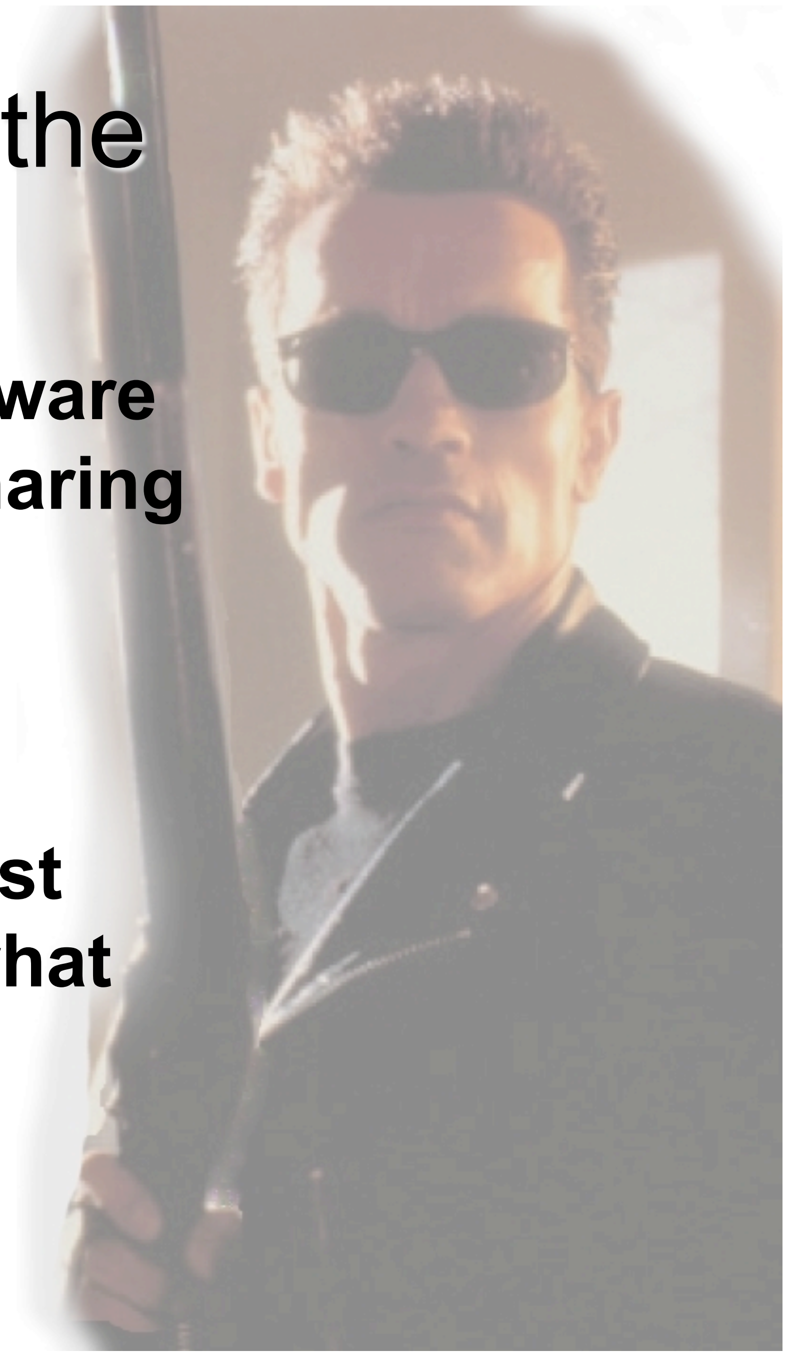
Building Trust in Software

Dr. Paul E. Black

paul.black@nist.gov

KC7PKT



Software Can Make the Hardware Your Ally

- **i.e., properly written software on radios can enforce sharing protocols and following (external) policies.**

- **So, how can we build trust that the system will do what we want?**

Trust in Software Comes From Three Sources:

Develop. Process
Execution
Analysis
CoolClips.com

Trust Begins With Good Process

- **Trustworthy software must be developed with care, for instance:**
  - **Validate requirements**
  - **Simplify the system architecture**
  - **Design compliance into the software**
  - **Prove a trust argument during development**
  - **Train programmers**
  - **Program with helpful languages**

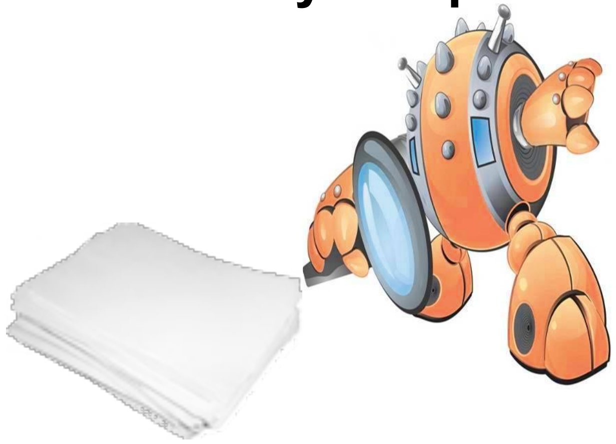
Analysis Builds Trust in Software

- **There are two general kinds of software analysis:**
  - **Static analysis**
    - **e.g. design review, code review, and scanner tools**
    - **examines code**
  - **Testing (dynamic analysis)**
    - **e.g. simulations, fault injection, and test beds**
    - **runs code**

Static Analysis and Testing Complement Each Other

Static Analysis

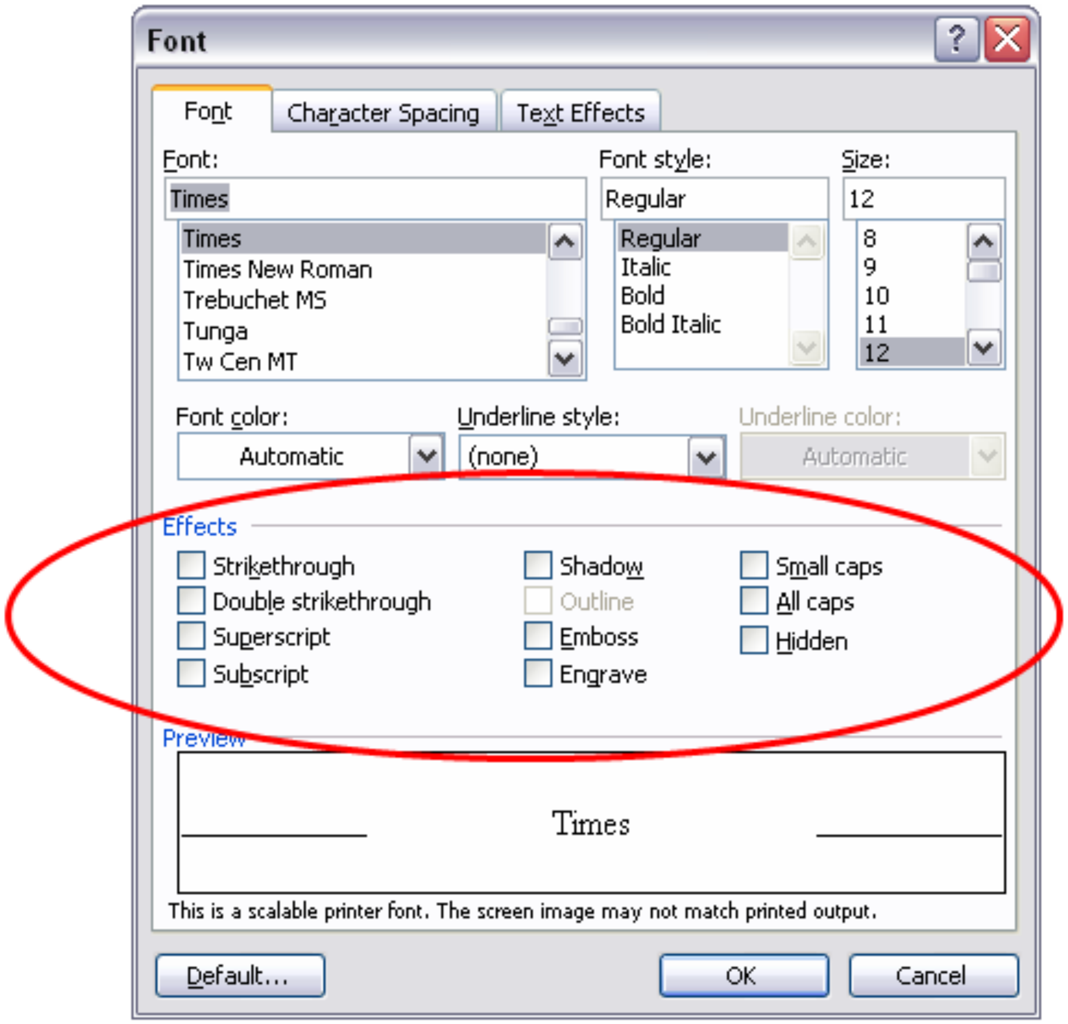
- Handles unfinished code
- Higher level artifacts
- Can find backdoors, e.g., no exclusions when frequency 105.7 entered
- Potentially complete



Testing

- Code not needed, e.g., embedded systems
- Covers end-to-end or system tests
- Assess as-installed
- Has few(er) assumptions

Use Combinatorial Testing



Font

Font | Character Spacing | Text Effects

Font: Times
- Times
- Times New Roman
- Trebuchet MS
- Tunga
- Tw Cen MT

Font style: Regular
- Regular
- Italic
- Bold
- Bold Italic

Size: 12
- 8
- 9
- 10
- 11
- 12

Font color: Automatic

Underline style: (none)

Underline color: Automatic

Effects

☐ Strikethrough	☐ Shadow	☐ Small caps
☐ Double strikethrough	☐ Outline	☐ All caps
☐ Superscript	☐ Emboss	☐ Hidden
☐ Subscript	☐ Engrave	

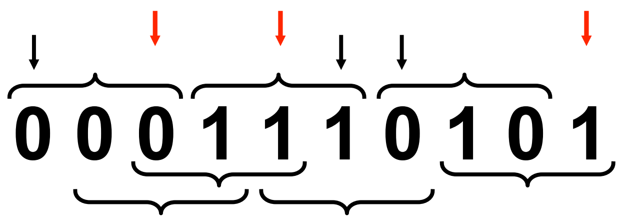
Preview

Times

This is a scalable printer font. The screen image may not match printed output.

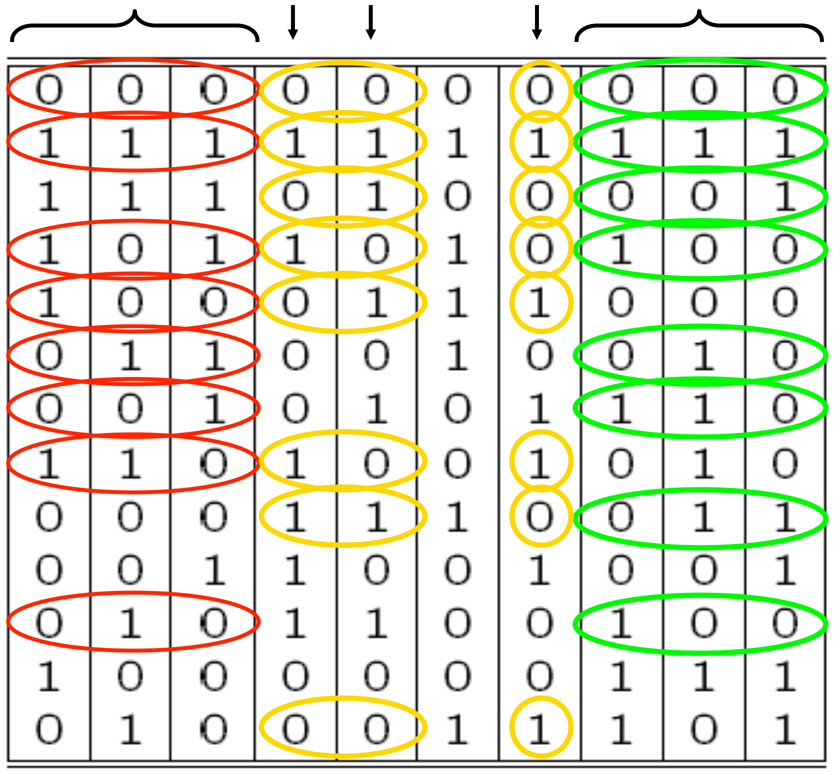
How Many Tests Would It Take?

- There are $\begin{bmatrix}10\\3\end{bmatrix}$ = 120 3-way interactions.
- Naively 120 x 2^3 = 960 tests.
- Since we can pack 3 triples into each test, we need no more than 320 tests.
- But each test exercises many triples:



We oughta be able to pack a lot in one test, so what’s the smallest number we need?

All Triples Take Only 13 Tests!



0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1	1	1
1	1	1	0	1	0	0	0	0	1
1	0	1	1	0	1	0	1	0	0
1	0	0	0	1	1	1	0	0	0
0	1	1	0	0	1	0	0	1	0
0	0	1	0	1	0	1	1	1	0
1	1	0	1	0	0	1	0	1	0
0	0	0	1	1	1	0	0	1	1
0	0	1	1	0	0	1	0	0	1
0	1	0	1	1	0	0	1	0	0
1	0	0	0	0	0	0	1	1	1
0	1	0	0	0	1	1	1	0	1