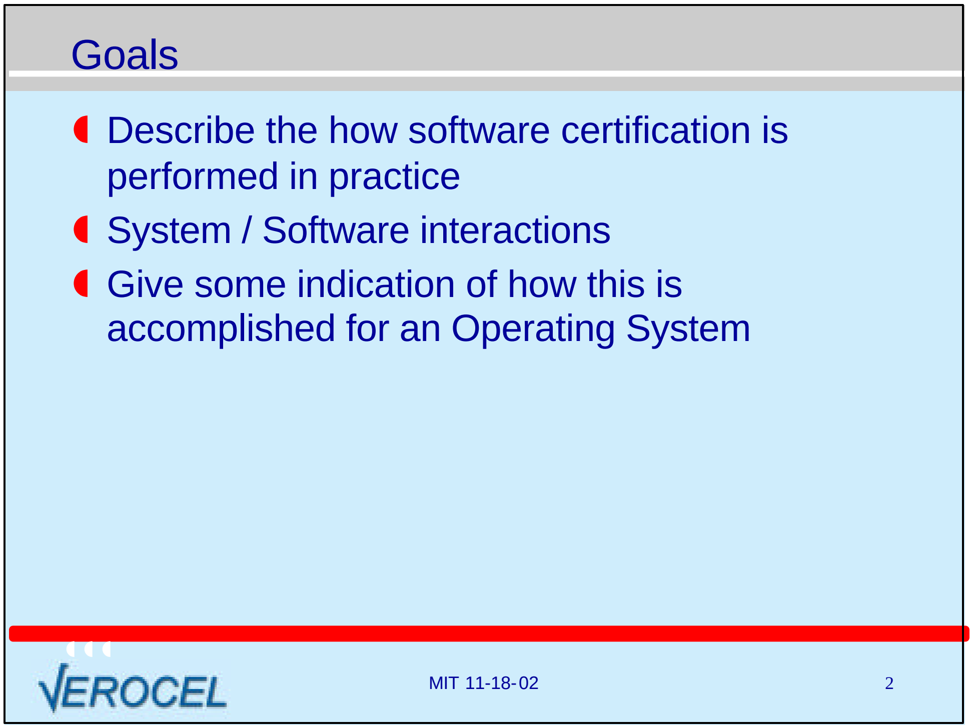
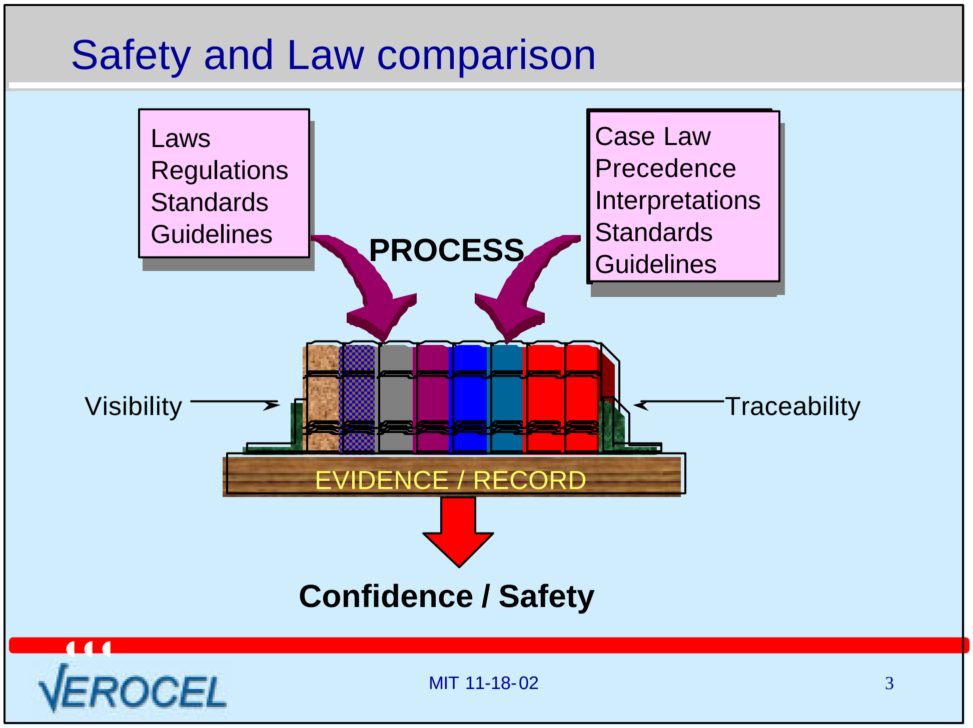
Software Certification In practice

George Romanski
romanski@verocel.com



Goals

- Describe the how software certification is performed in practice
- System / Software interactions
- Give some indication of how this is accomplished for an Operating System



Safety and Law comparison

Laws
Regulations
Standards
Guidelines

Case Law
Precedence
Interpretations
Standards
Guidelines

PROCESS

Visibility

Traceability

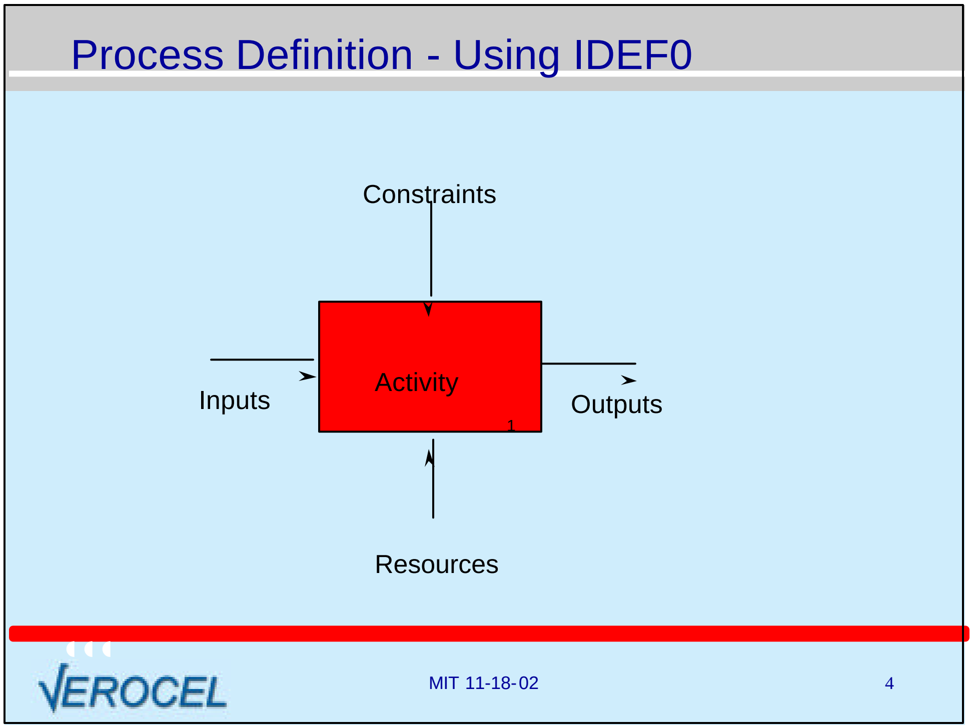
EVIDENCE / RECORD

Confidence / Safety

VEROCEL

MIT 11-18-02

3



Process Definition - Using IDEF0

Constraints

Inputs

Activity

1

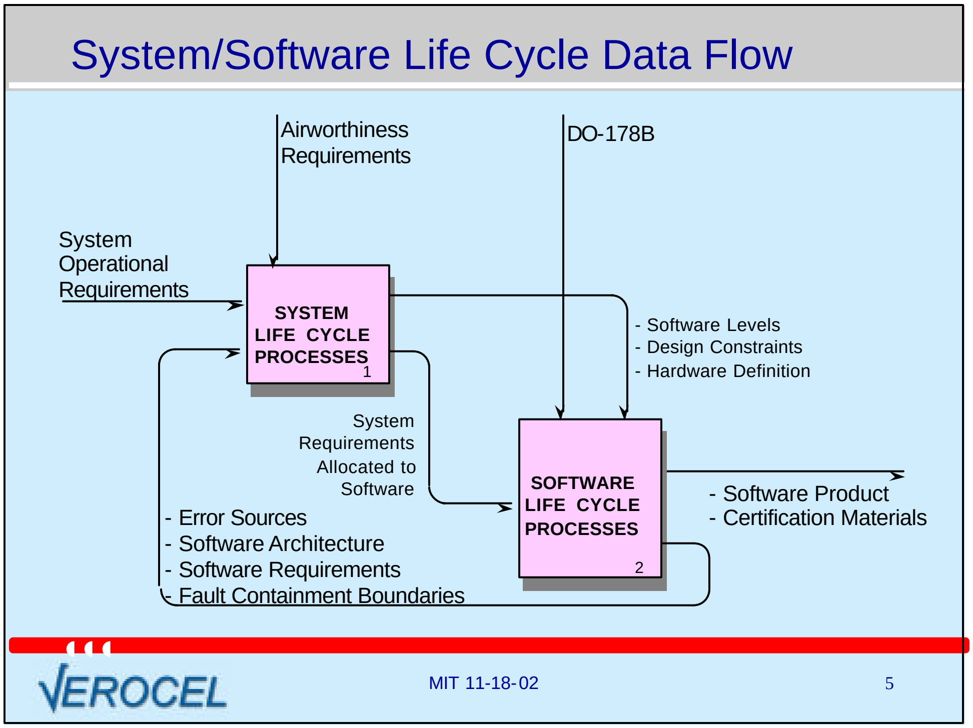
Outputs

Resources

VEROCEL

MIT 11-18-02

4



System/Software Life Cycle Data Flow

Airworthiness Requirements

DO-178B

System Operational Requirements

SYSTEM LIFE CYCLE PROCESSES 1

- Software Levels
- Design Constraints
- Hardware Definition

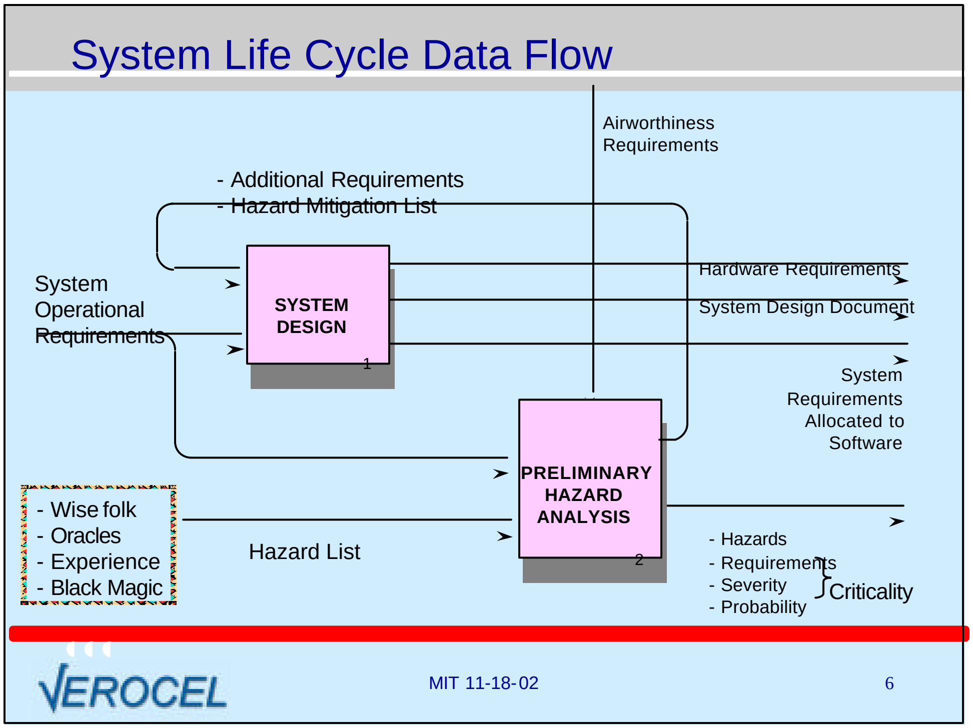
System Requirements Allocated to Software

SOFTWARE LIFE CYCLE PROCESSES 2

- Software Product
- Certification Materials

- Error Sources
- Software Architecture
- Software Requirements
- Fault Containment Boundaries

VEROCEL MIT 11-18-02 5



System Life Cycle Data Flow

Airworthiness Requirements

- Additional Requirements
- Hazard Mitigation List

System Operational Requirements

SYSTEM DESIGN 1

Hardware Requirements

System Design Document

System Requirements Allocated to Software

PRELIMINARY HAZARD ANALYSIS 2

- Wise folk
- Oracles
- Experience
- Black Magic

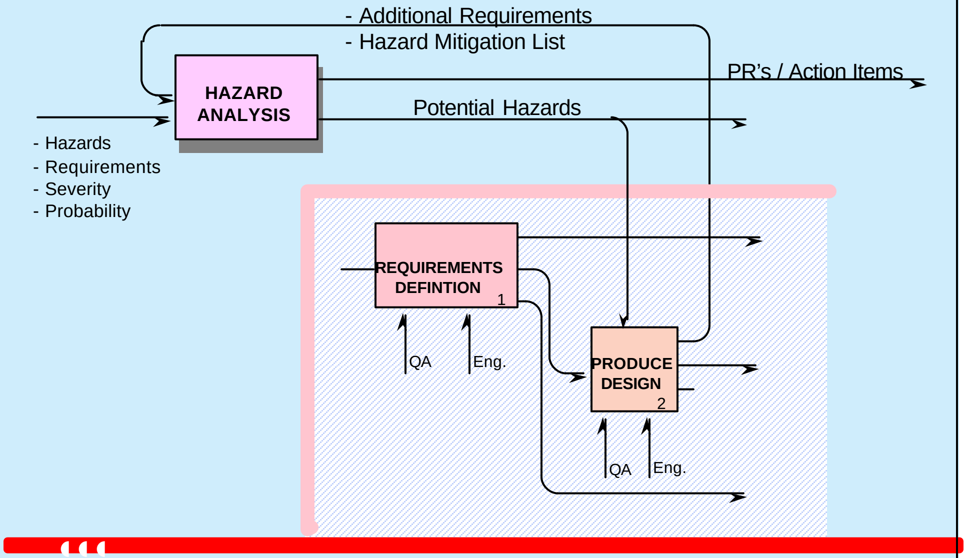
Hazard List

- Hazards
- Requirements
- Severity
- Probability

} Criticality

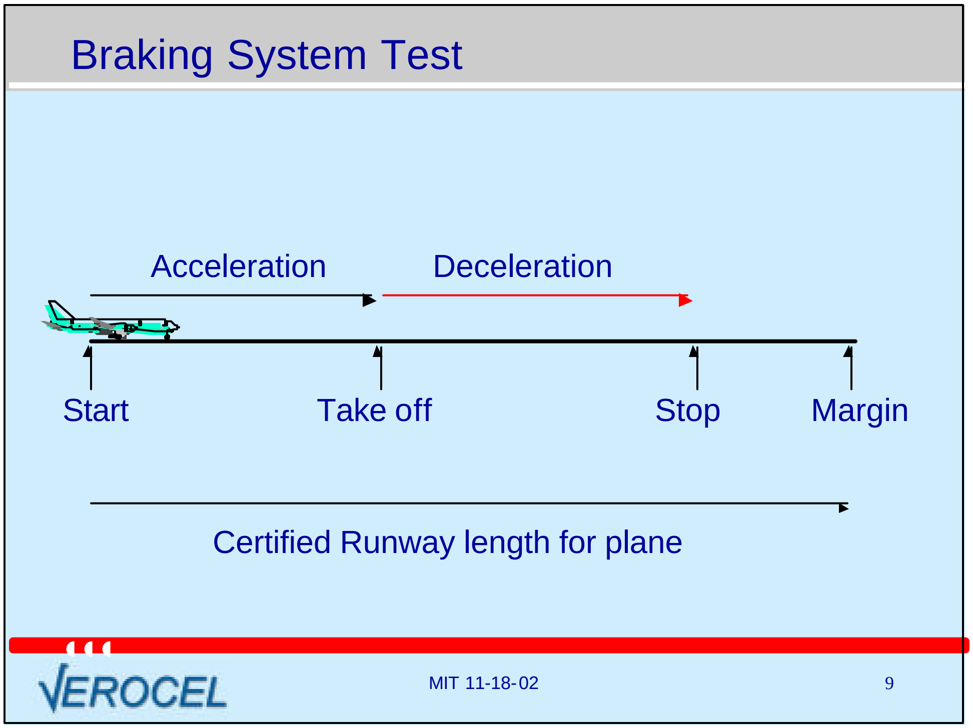
VEROCEL MIT 11-18-02 6

Subsystem Hazard Analysis



Braking System

- Boeing 777 – 14 wheels
- Each wheel has a braking system
- Each braking system has a computer (MC68332)
- Computer software verified independently
- System Test is required



Braking System Test

Acceleration

Deceleration

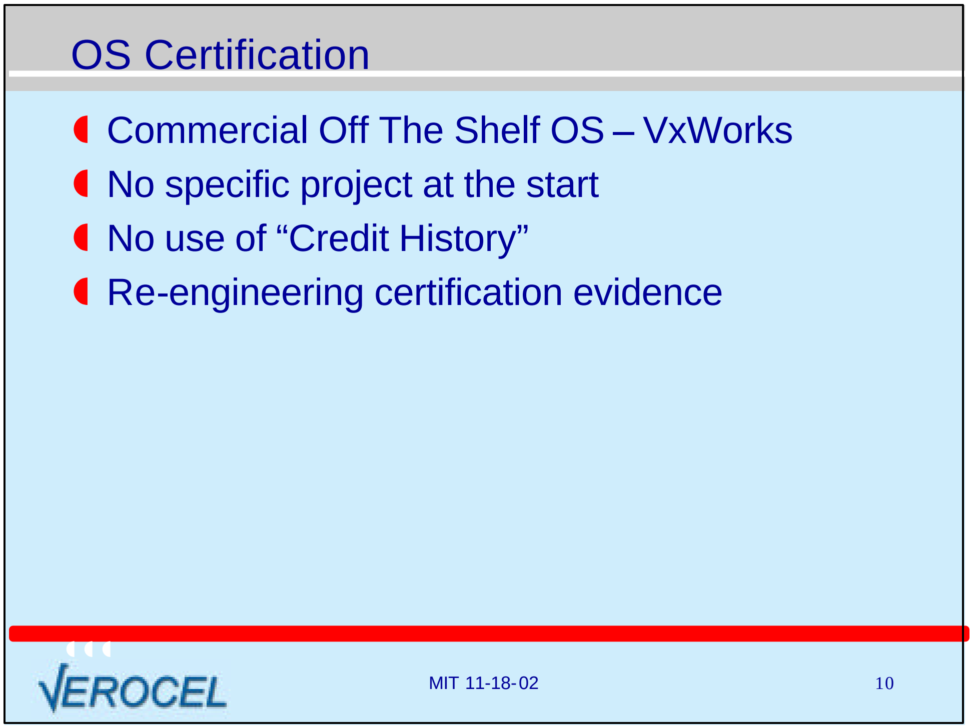
Start

Take off

Stop

Margin

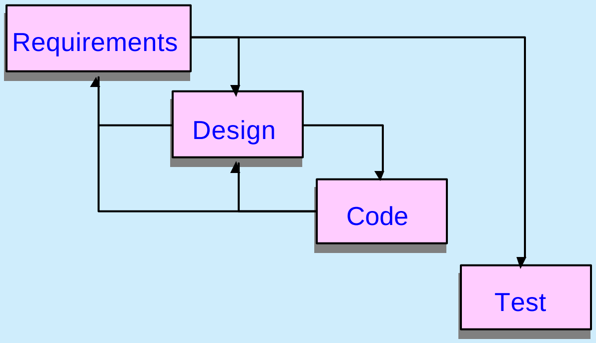
Certified Runway length for plane



OS Certification

- Commercial Off The Shelf OS – VxWorks
- No specific project at the start
- No use of “Credit History”
- Re-engineering certification evidence

Waterfall model



Requirements

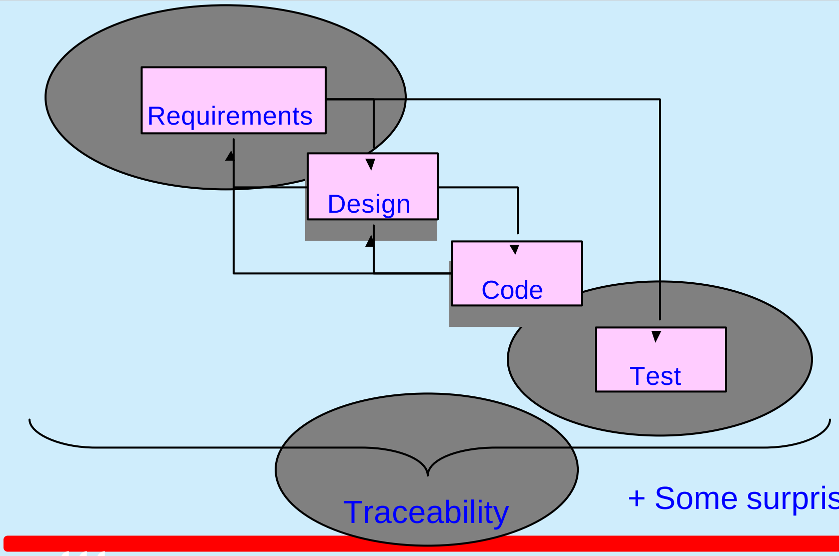
Design

Code

Test

Traceability

Focus on – Requirements, Test, Traceability



Requirements

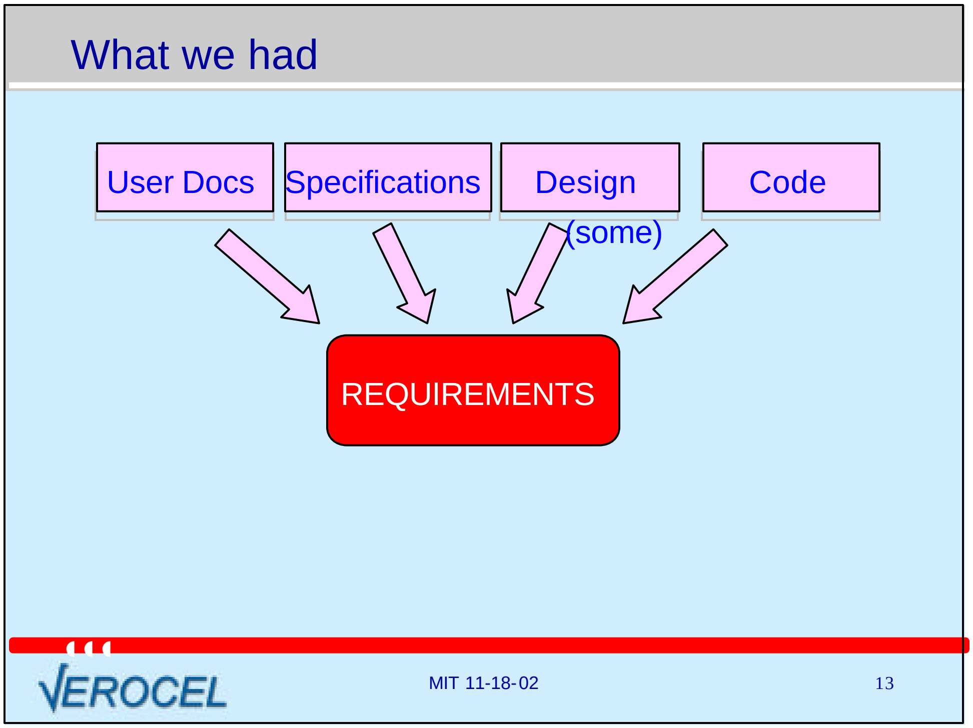
Design

Code

Test

Traceability

+ Some surprises



What we had

User Docs

Specifications

Design

Code

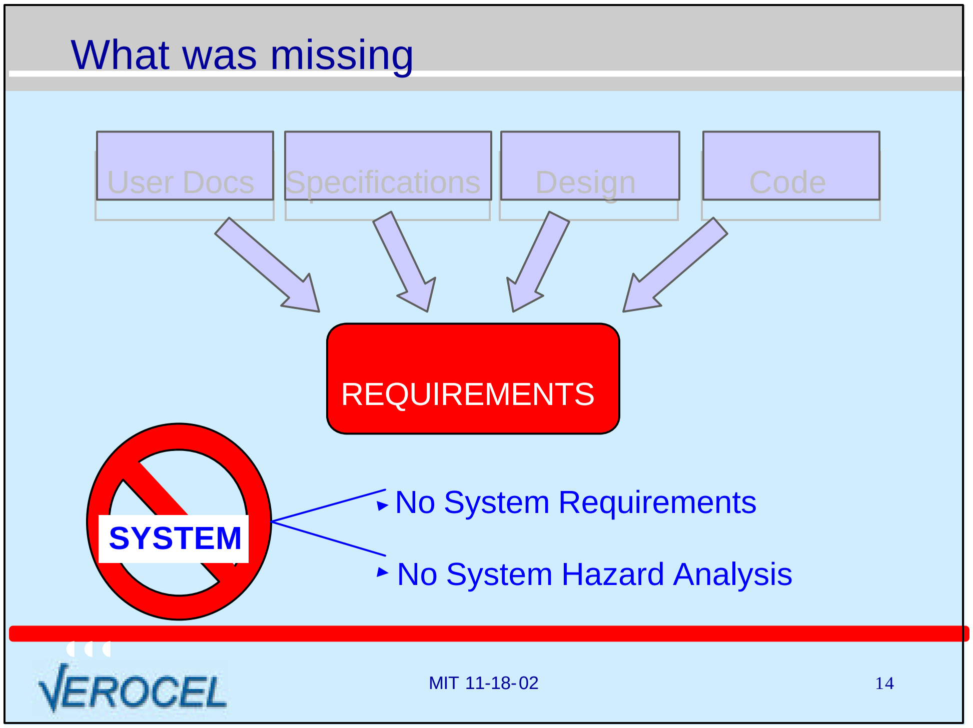
(some)

REQUIREMENTS

VEROCEL

MIT 11-18-02

13



What was missing

User Docs

Specifications

Design

Code

REQUIREMENTS

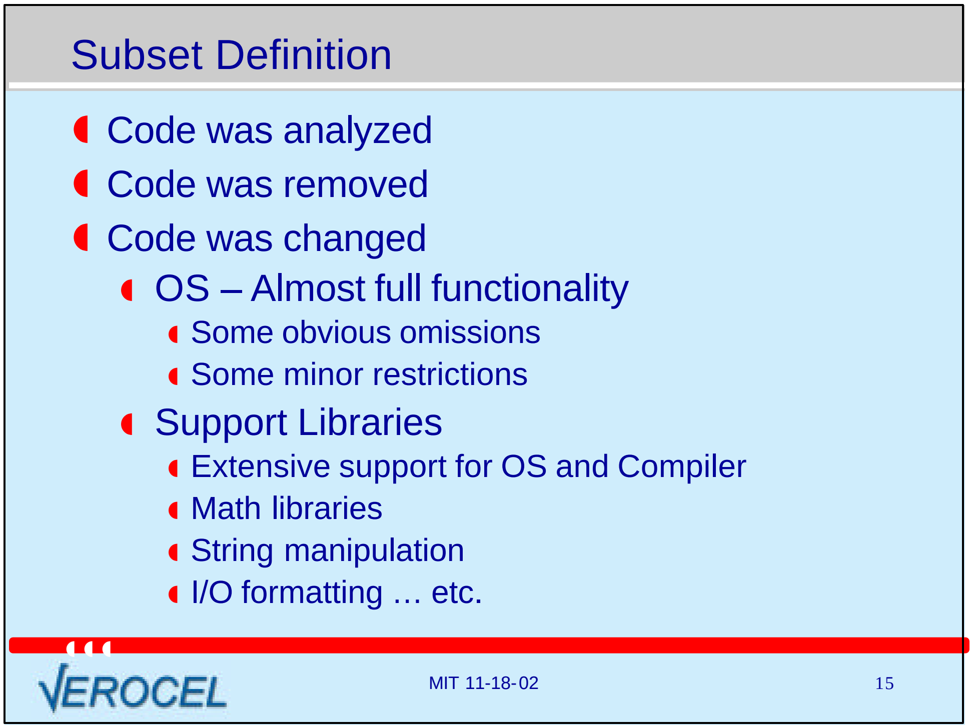
SYSTEM

- No System Requirements
- No System Hazard Analysis

VEROCEL

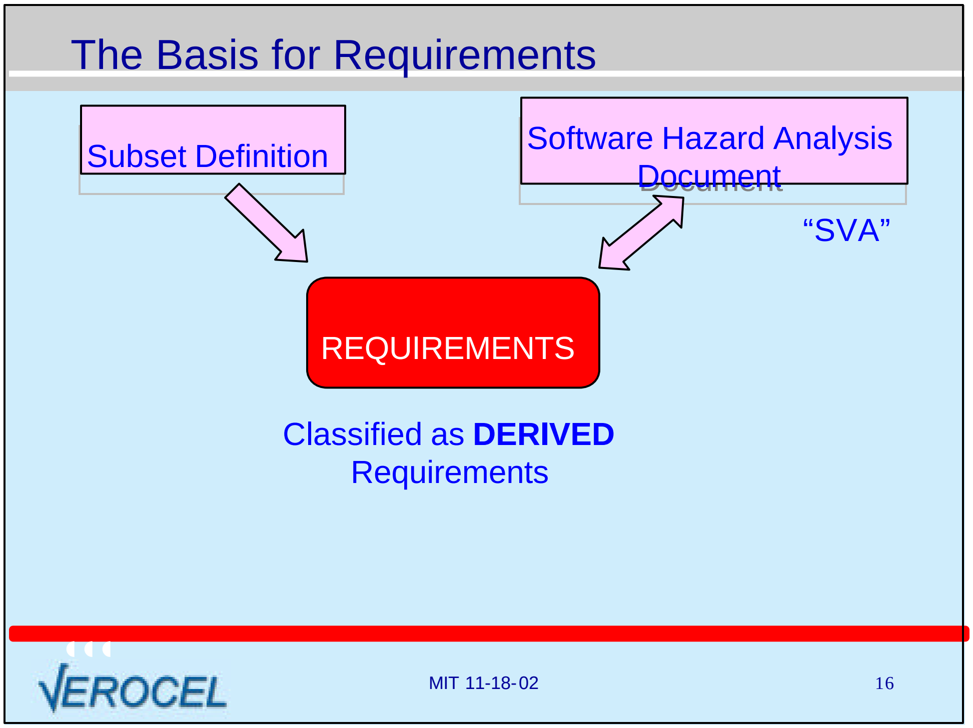
MIT 11-18-02

14



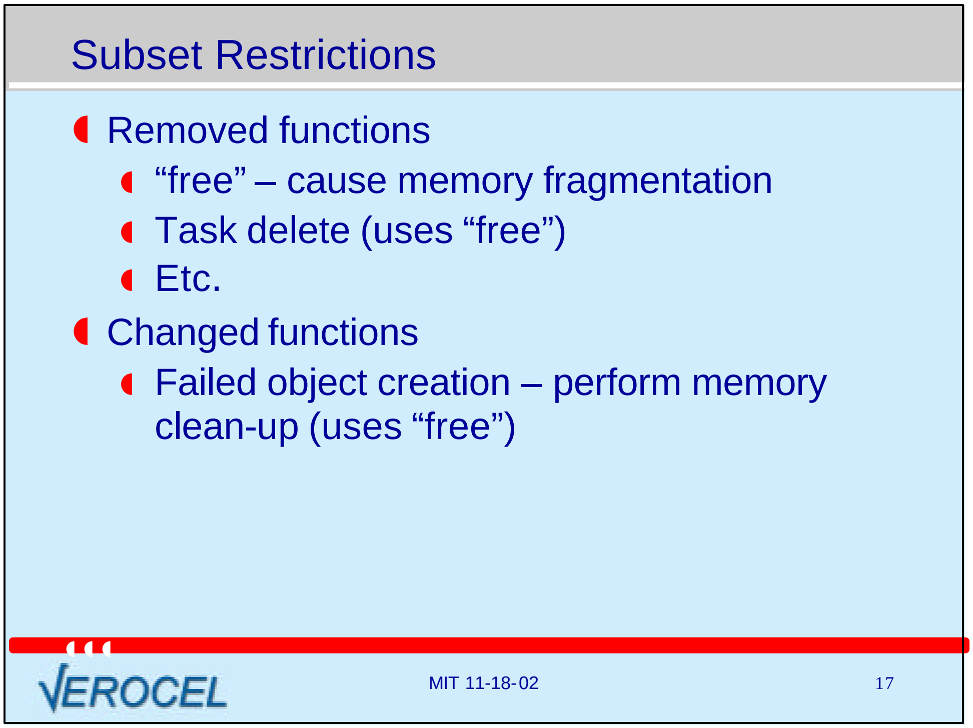
Subset Definition

- Code was analyzed
- Code was removed
- Code was changed
  - OS – Almost full functionality
    - Some obvious omissions
    - Some minor restrictions
  - Support Libraries
    - Extensive support for OS and Compiler
    - Math libraries
    - String manipulation
    - I/O formatting … etc.



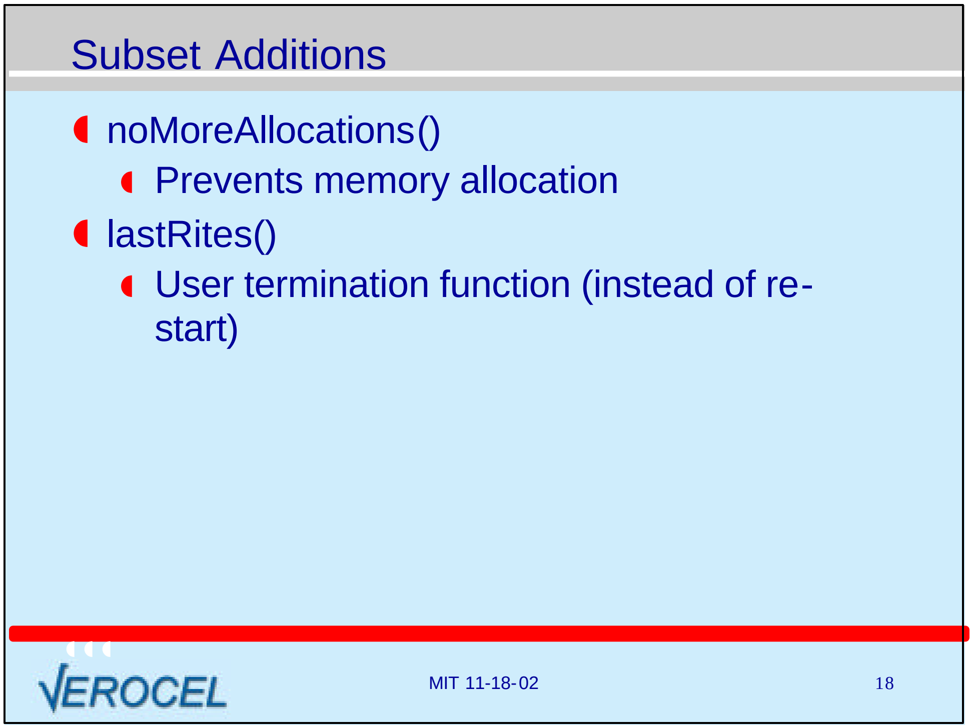
The Basis for Requirements

Classified as **DERIVED** Requirements



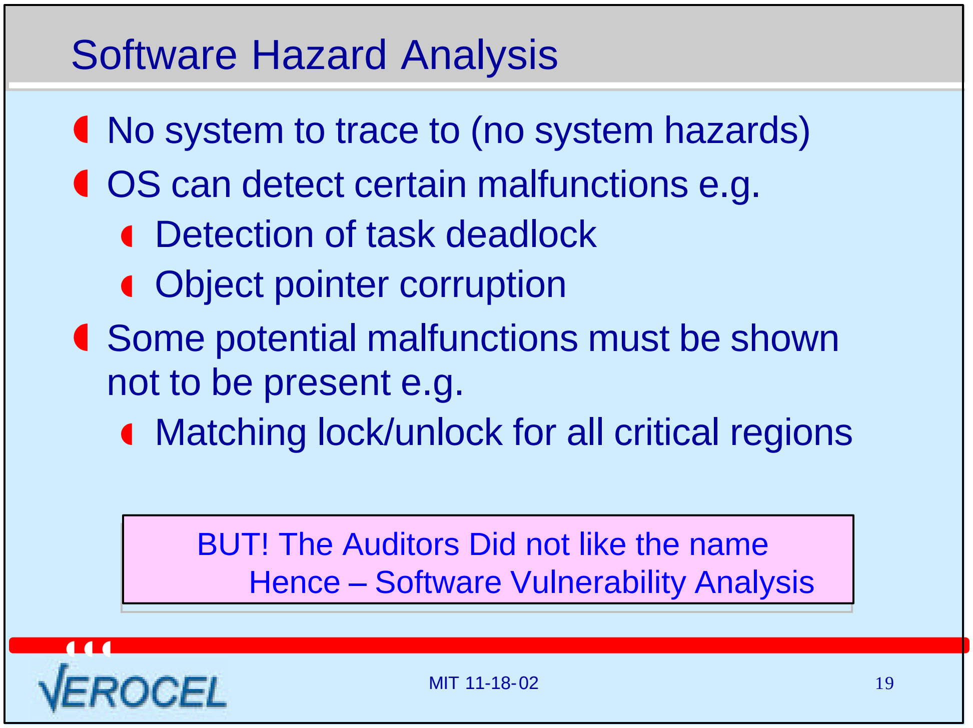
Subset Restrictions

- Removed functions
  - “free” – cause memory fragmentation
  - Task delete (uses “free”)
  - Etc.
- Changed functions
  - Failed object creation – perform memory clean-up (uses “free”)



Subset Additions

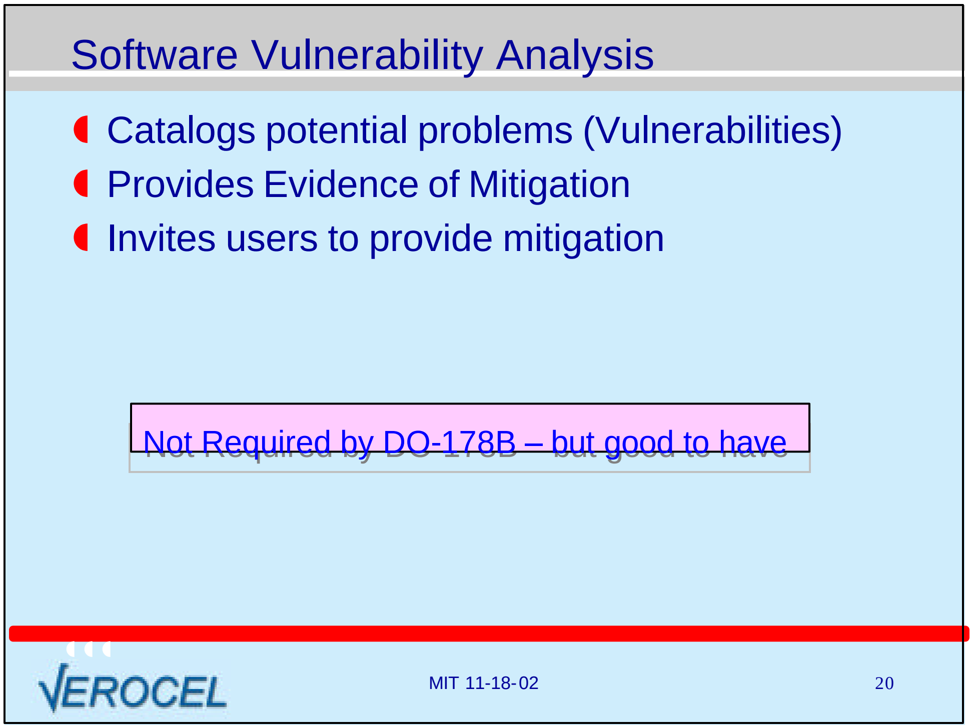
- noMoreAllocations()
  - Prevents memory allocation
- lastRites()
  - User termination function (instead of re-start)



Software Hazard Analysis

- No system to trace to (no system hazards)
- OS can detect certain malfunctions e.g.
  - Detection of task deadlock
  - Object pointer corruption
- Some potential malfunctions must be shown not to be present e.g.
  - Matching lock/unlock for all critical regions

BUT! The Auditors Did not like the name
Hence – Software Vulnerability Analysis

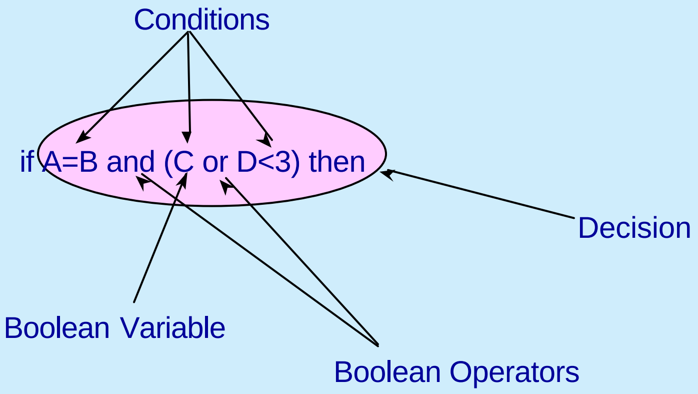


Software Vulnerability Analysis

- Catalogs potential problems (Vulnerabilities)
- Provides Evidence of Mitigation
- Invites users to provide mitigation

Not Required by DO-178B – but good to have

Conditions/Decisions



Conditions

if A=B and (C or D<3) then

Decision

Boolean Variable

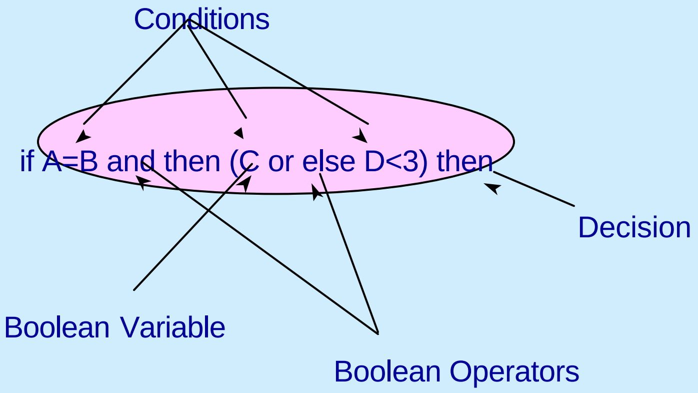
Boolean Operators

VEROCEL

MIT 11-18-02

21

Short –Circuit Forms



Conditions

if A=B and then (C or else D<3) then

Decision

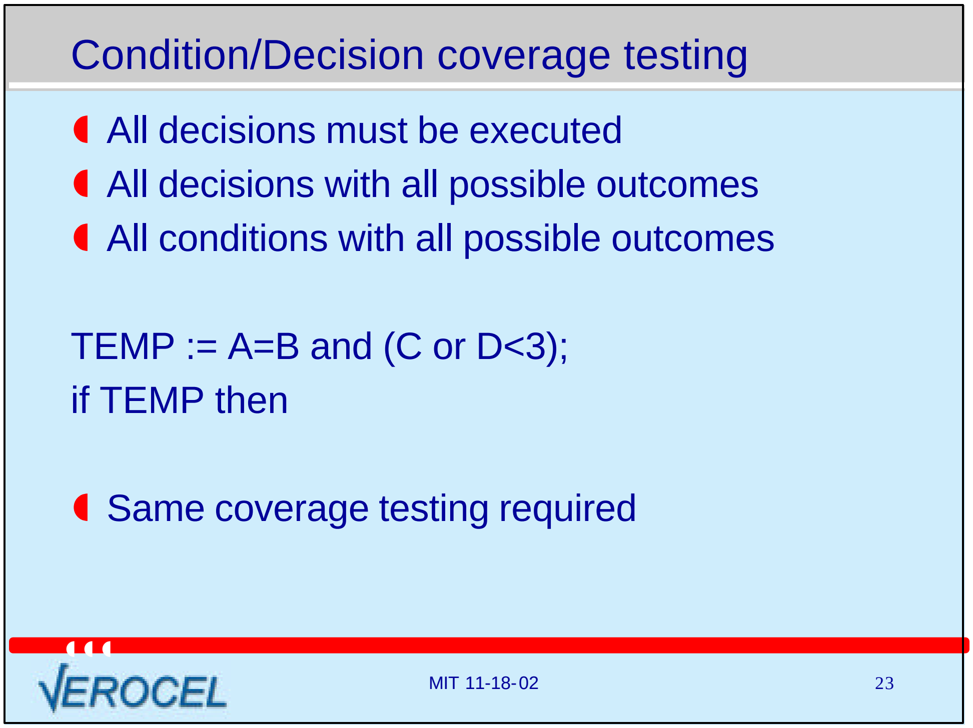
Boolean Variable

Boolean Operators

VEROCEL

MIT 11-18-02

22

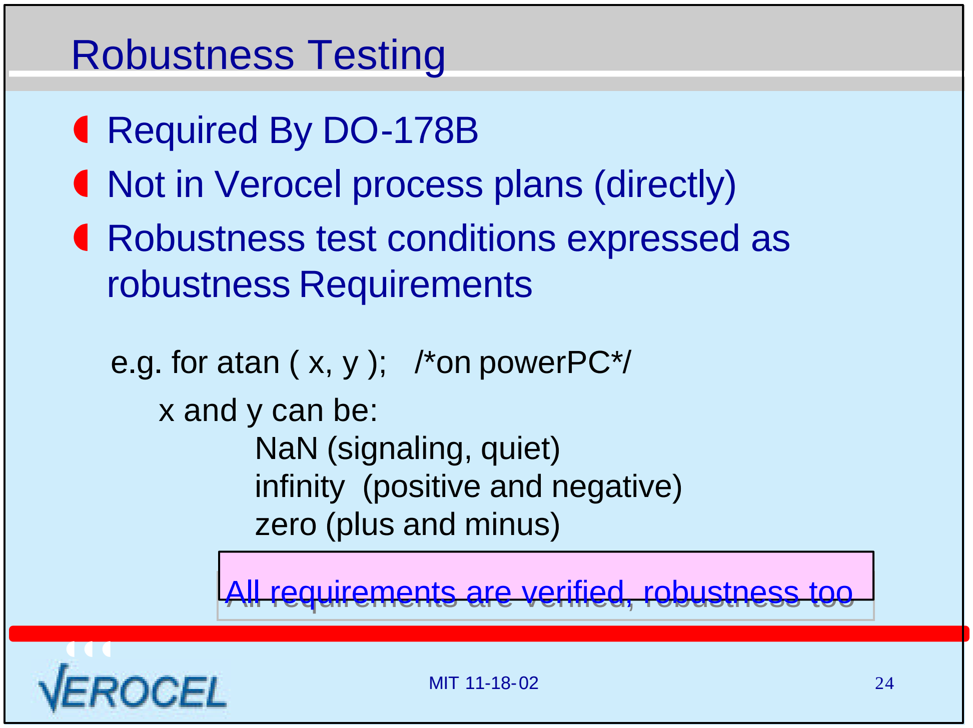


Condition/Decision coverage testing

- All decisions must be executed
- All decisions with all possible outcomes
- All conditions with all possible outcomes

TEMP := A=B and (C or D<3);
if TEMP then

- Same coverage testing required



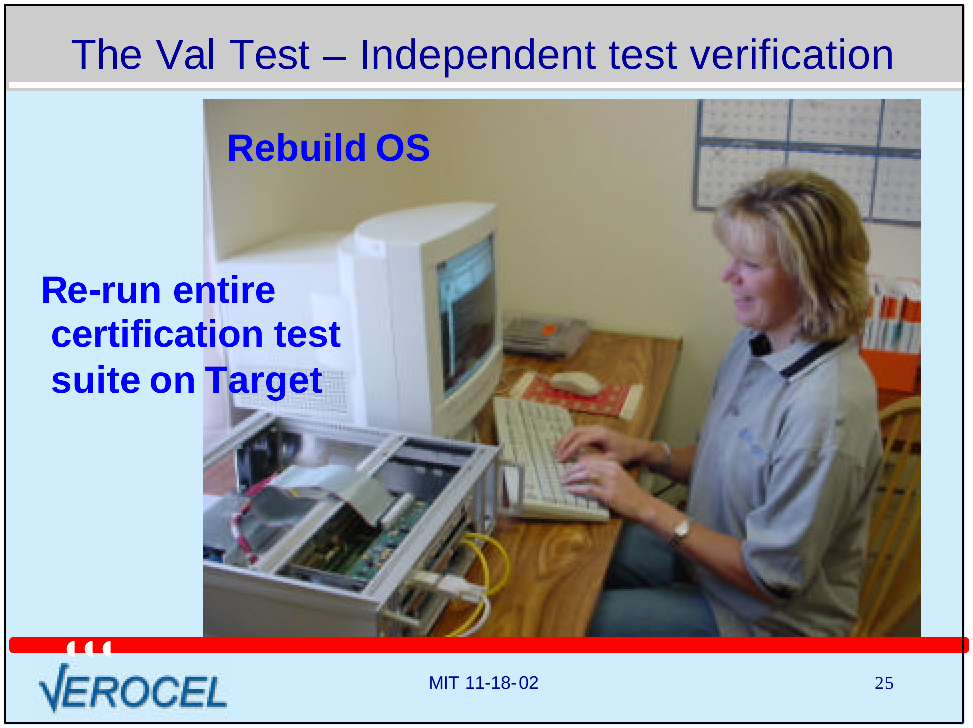
Robustness Testing

- Required By DO-178B
- Not in Verocel process plans (directly)
- Robustness test conditions expressed as robustness Requirements

e.g. for atan (x, y); /*on powerPC*/

x and y can be:
NaN (signaling, quiet)
infinity (positive and negative)
zero (plus and minus)

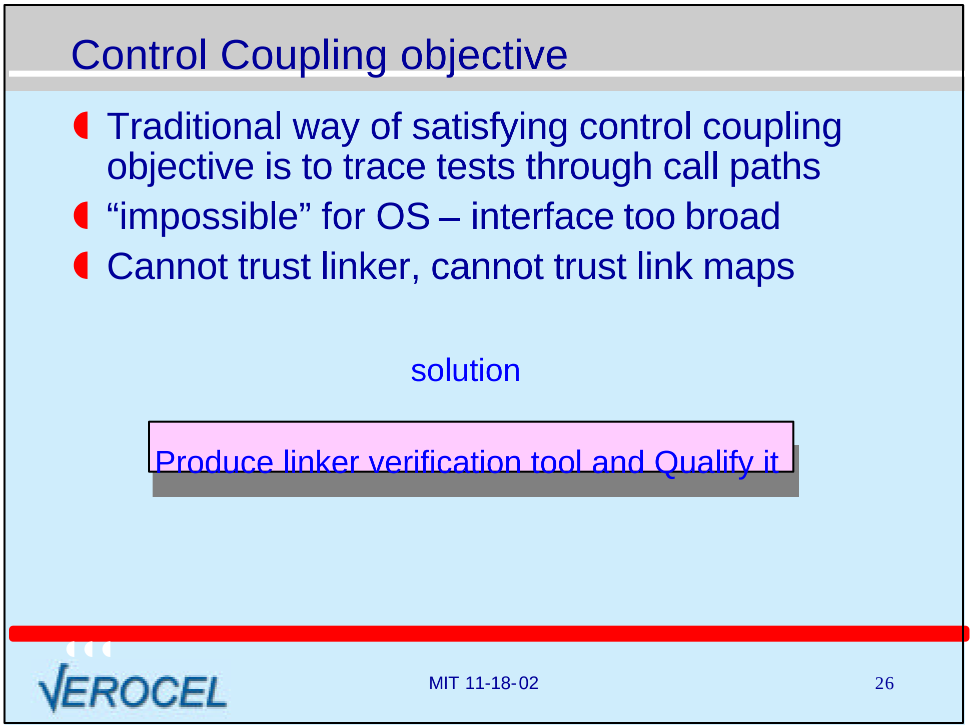
All requirements are verified, robustness too



The Val Test – Independent test verification

Rebuild OS

Re-run entire certification test suite on Target



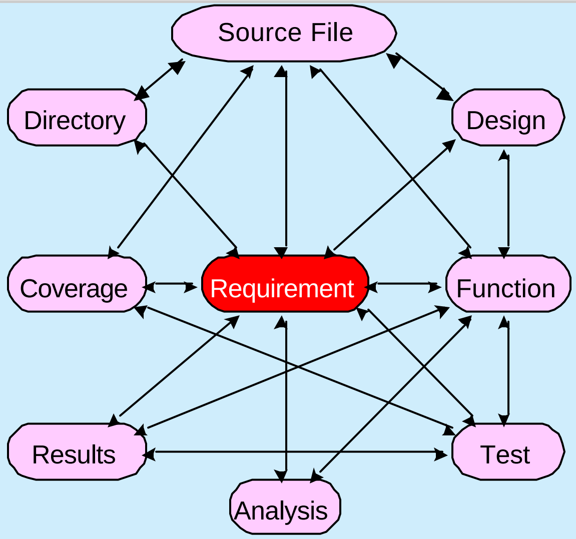
Control Coupling objective

- Traditional way of satisfying control coupling objective is to trace tests through call paths
- “impossible” for OS – interface too broad
- Cannot trust linker, cannot trust link maps

solution

Produce linker verification tool and Qualify it

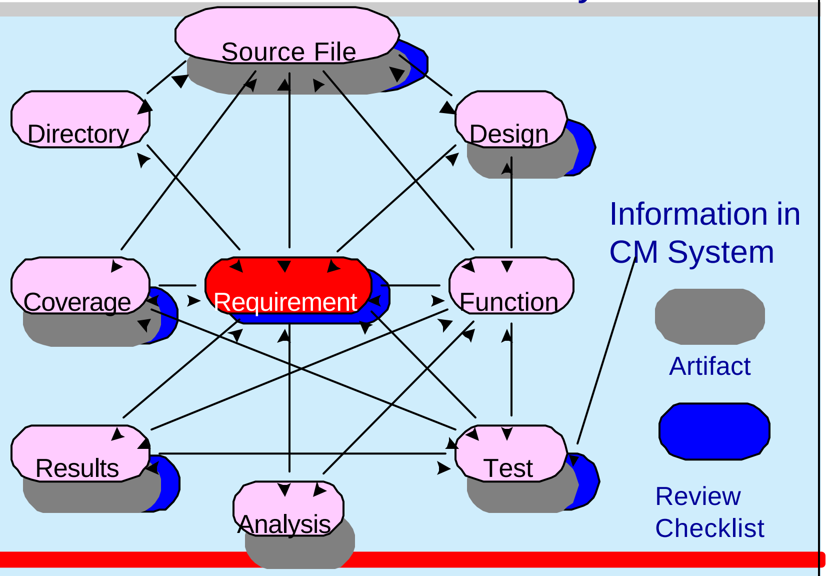
Requirements are Central to Traceability



Source File
Directory
Design
Coverage
Requirement
Function
Results
Test
Analysis

MIT 11-18-02

Artifacts/Reviews are held in CM system



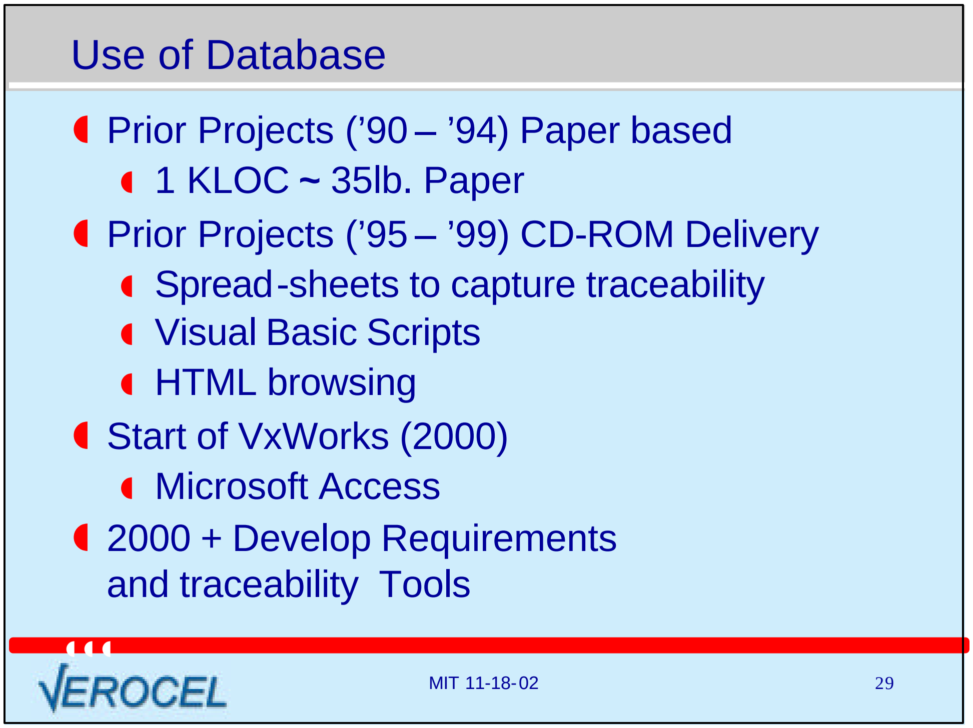
Source File
Directory
Design
Coverage
Requirement
Function
Results
Test
Analysis

Information in CM System

Artifact

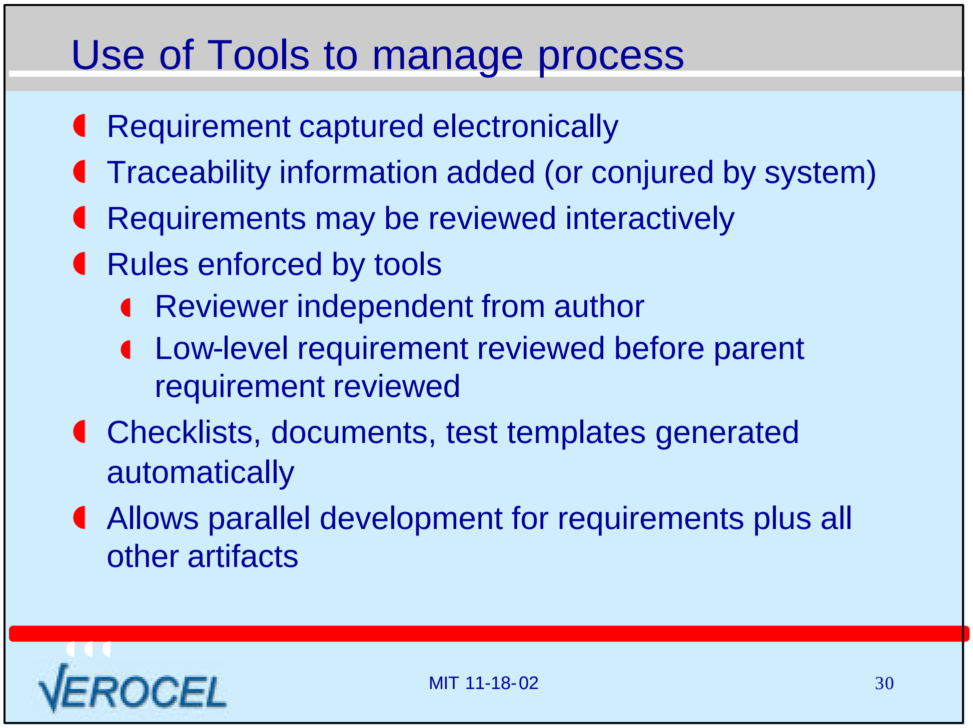
Review Checklist

MIT 11-18-02



Use of Database

- Prior Projects ('90 – '94) Paper based
  - 1 KLOC ~ 35lb. Paper
- Prior Projects ('95 – '99) CD-ROM Delivery
  - Spread-sheets to capture traceability
  - Visual Basic Scripts
  - HTML browsing
- Start of VxWorks (2000)
  - Microsoft Access
- 2000 + Develop Requirements and traceability Tools



Use of Tools to manage process

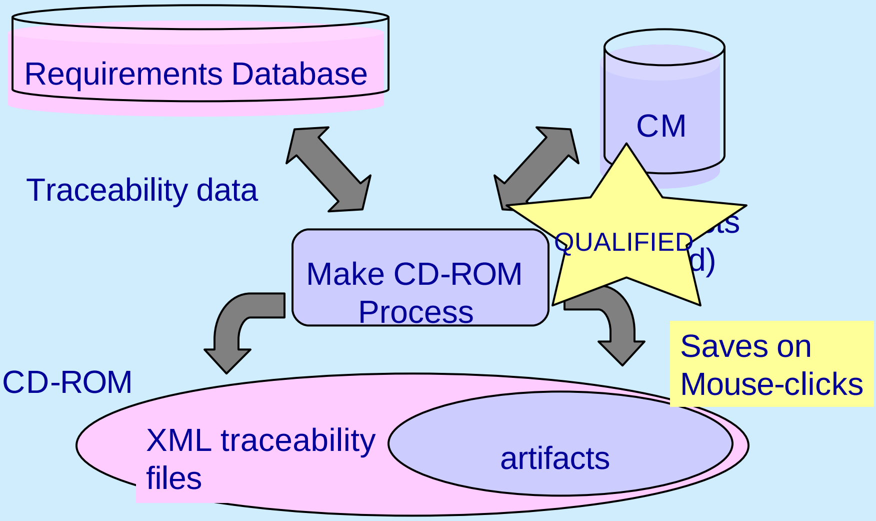
- Requirement captured electronically
- Traceability information added (or conjured by system)
- Requirements may be reviewed interactively
- Rules enforced by tools
  - Reviewer independent from author
  - Low-level requirement reviewed before parent requirement reviewed
- Checklists, documents, test templates generated automatically
- Allows parallel development for requirements plus all other artifacts

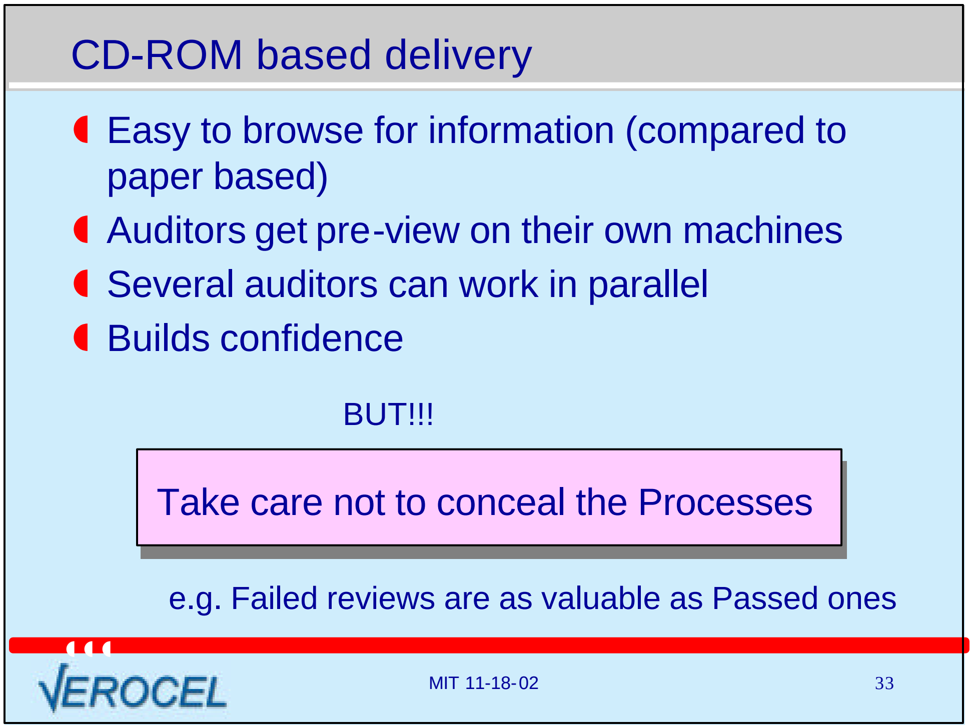
The Numbers – VxWorks/Cert – no BSP

Lines of Code	12,000 *
Requirements	1,300
Test Files	720
Lines of test	48,000
Review Files	2,900
XML Traceability Files	14,600

* Includes many support libraries

Delivery medium – CD-ROM





CD-ROM based delivery

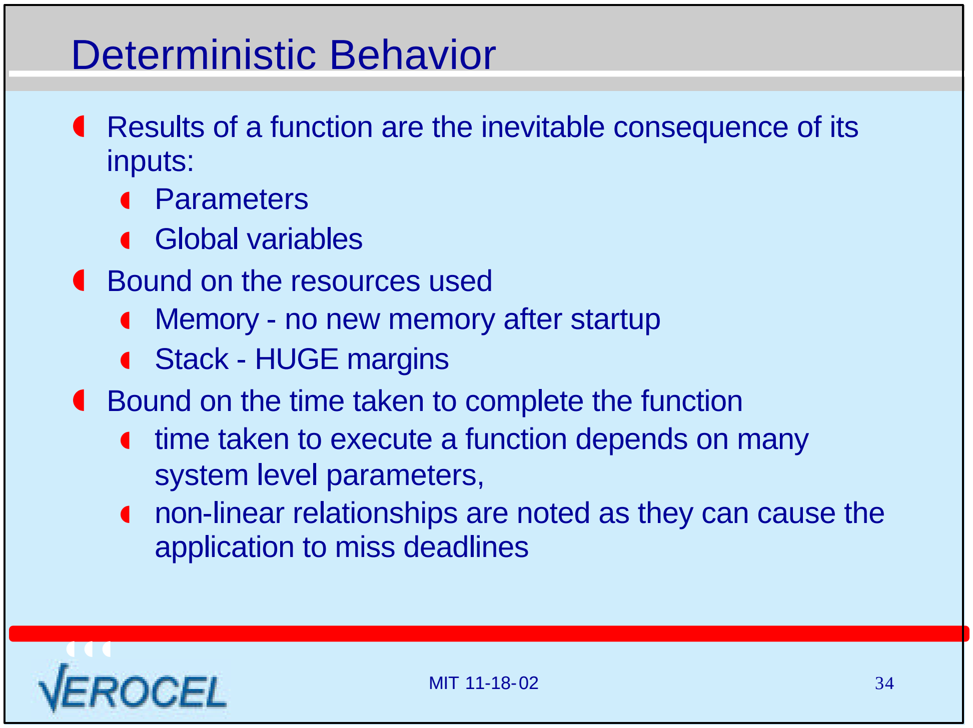
- Easy to browse for information (compared to paper based)
- Auditors get pre-view on their own machines
- Several auditors can work in parallel
- Builds confidence

BUT!!!

Take care not to conceal the Processes

e.g. Failed reviews are as valuable as Passed ones

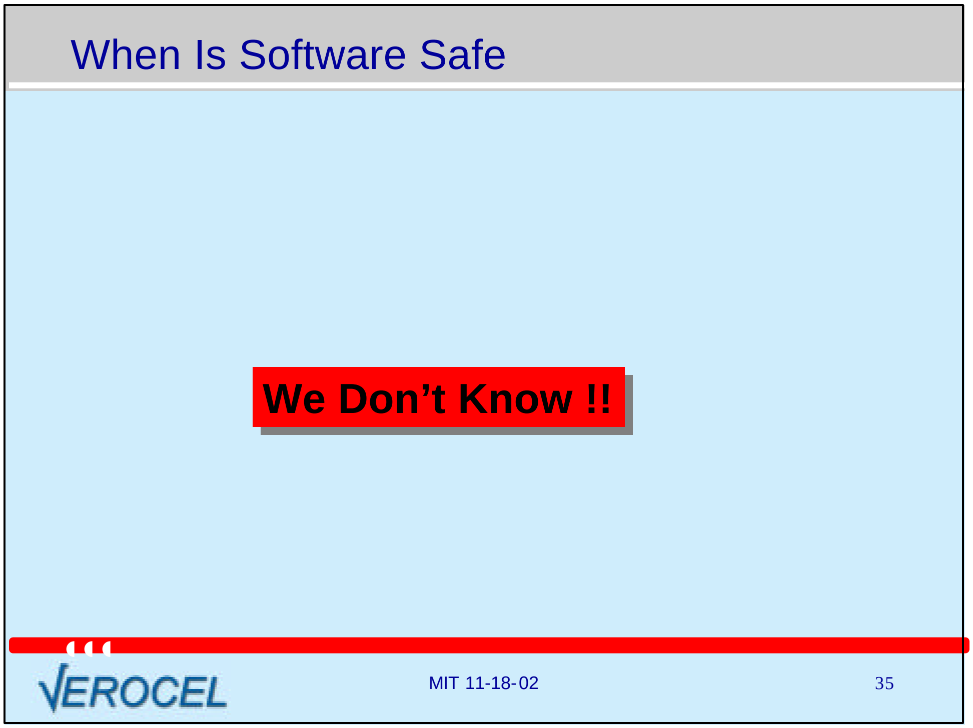
VEROCEL MIT 11-18-02 33



Deterministic Behavior

- Results of a function are the inevitable consequence of its inputs:
  - Parameters
  - Global variables
- Bound on the resources used
  - Memory - no new memory after startup
  - Stack - HUGE margins
- Bound on the time taken to complete the function
  - time taken to execute a function depends on many system level parameters,
  - non-linear relationships are noted as they can cause the application to miss deadlines

VEROCEL MIT 11-18-02 34



When Is Software Safe

We Don't Know !!

What is our best guess about the safety

- When applicable processes have been followed
- When we have verified the code "from within"
- When this has been checked
  - and checked
  - and checked
  - and checked
  - and checked
  - and checked
  - and checked