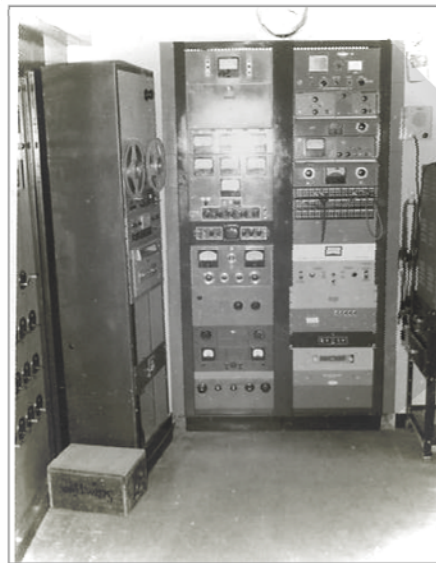


# Why Not Create Standard Tests?

Rob Andzik  
AMERGINT Technologies  
GSAW 2012

© 2012 by AMERGINT Technologies. Published by The Aerospace Corporation with permission.

- **Technology Has Changed Satellite Ground Systems**
  - Large, heavy equipment is being replaced with software
  - System capabilities are increasing exponentially
  - We continue to push for more standardization
- **Have Our Processes and Testing Capabilities Kept Pace?**
  - Are systems being delivered on time and under budget?
  - Do our systems consistently meet requirements?



Maybe we need to consider  
changing our approach to better  
leverage the changing technologies

- **Verification Testing**

- Procedures strictly focus on the identified system requirements
- Generally lacking in areas like reliability, robustness and completeness of implementation
- Test procedures are developed late and often used only for the initial delivery

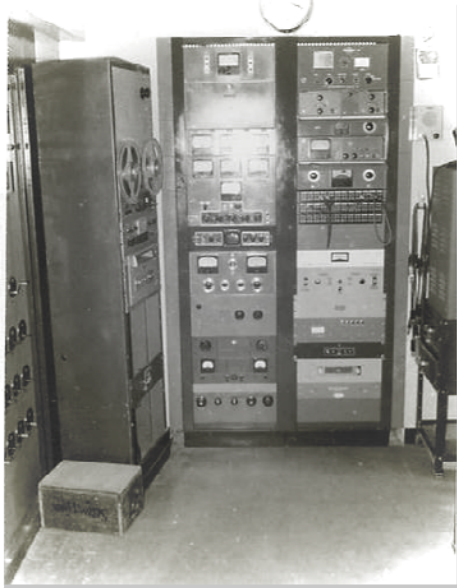
- **Processes**

- Ensure tasks are always done the same way and as prescribed
- General assumptions
  - The processes actually apply to the system being delivered
  - Certification to a certain level guarantees quality products

- **Some Unintended Consequences**

- High cost both in money, time and resources
- Burdensome processes can further increase cost and stifle innovation
- This approach maps poorly to COTS products

- **Software-Based Systems Have Seemingly Unlimited Capacity**
  - This can be both a blessing and a curse
    - The overall complexity and required validation is **not** reduced
    - Rarely is there any reuse of legacy tests
  - New products & technologies often have more capabilities
    - More requirements are mapped to COTS products
    - More functionality can be handled inside the 'black box'



How do we  
validate this?

**GROUND  
SYSTEM  
SOFTWARE**

Reports

GUIs

Databases

Security

3<sup>rd</sup> Party Tools

Custom

Requirements



- **The Impact Of Standards On Requirements and Technology**

- Products and knowledge evolve around the standards (Good)
- Greatly simplify requirement specifications (Good)

- Example:

The system shall process CCSDS Space Packets per the following:

CCSDS 732.0-B-2

CCSDS 133.0-B-1

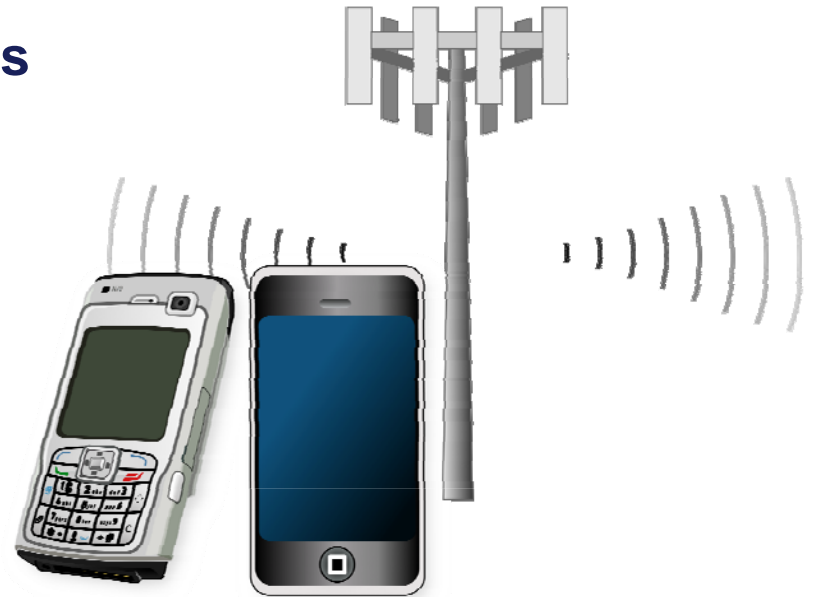
- **How Do We Validate These Type Of Requirements? (Yikes)**

- These ‘simple’ requirements carry a lot of weight
- What about statements like ‘if used’, ‘if implemented’?
  - These are necessary in the standards world, but can present implementers with as many as  $N^2$  potential combinations
- Programs create their own interpretation and acceptance criteria

Spec	Pages	# Shall Statements	# Conditional Statements
732.0-B-2	87	> 200	> 50
133.0-B-1	49	> 100	> 30

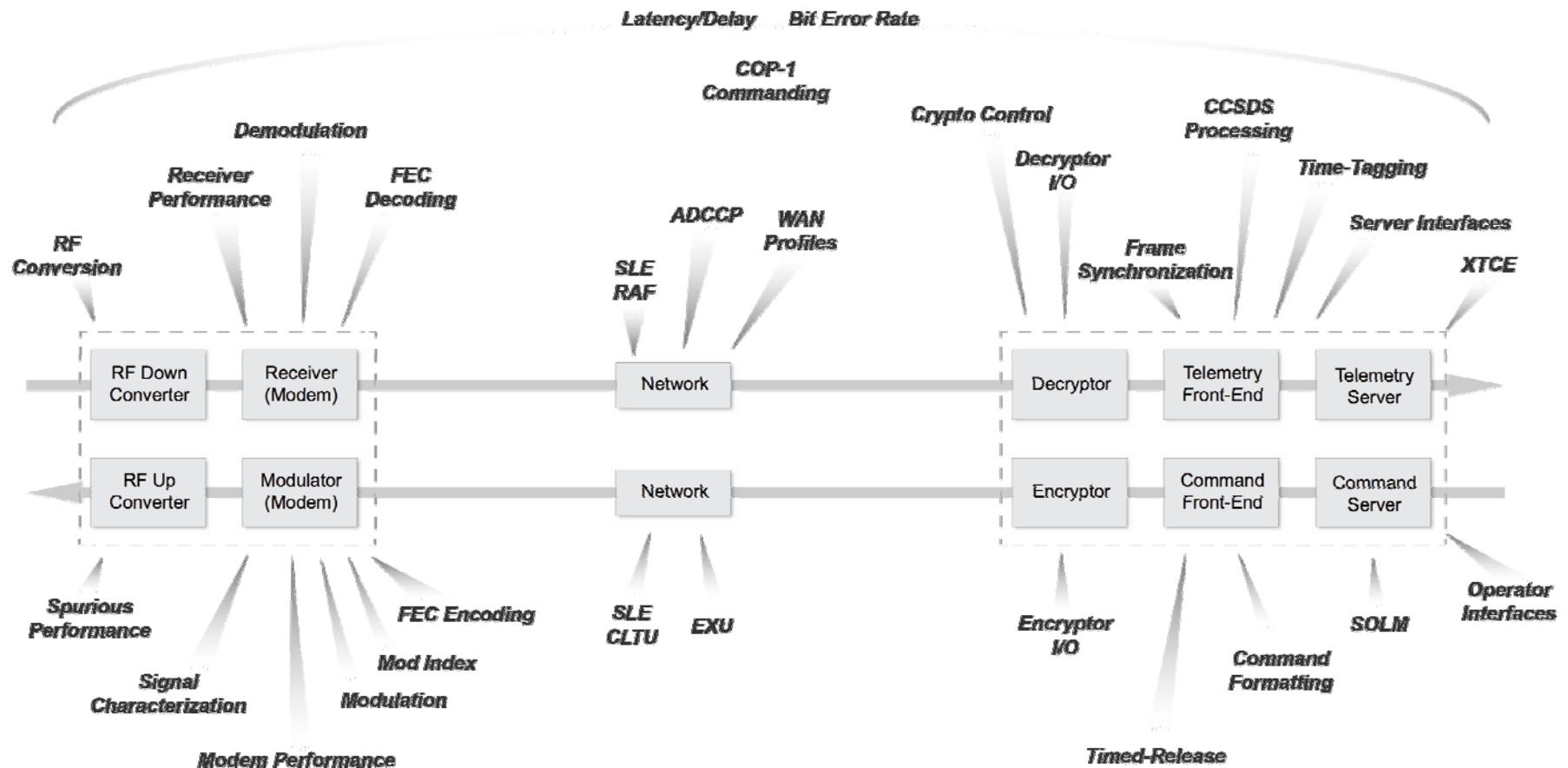
Often validated using the ‘trust me or prove me wrong’ approach

- **Standard Tests Are Common In Other Industries**
  - Provide a “litmus test”, verifying compliance with a specification
  - Examples: mobile phone industry, manufacturing, etc.
- **Software-Based Solutions Provide New Options For Testing**
  - Test Instruments can be defined in software
  - Need for custom test hardware is greatly reduced
  - Technologies like Virtual Machines provide longevity well beyond hardware and OS life-cycles
- **Standard Tests Take Various Forms**
  - Integrated Test System
  - Virtual Machine
  - Platform Independent Application
  - Sample Data Set
  - Even Standard Test Procedures



- **Customer/Government Furnished Tests**
  - Tests for various standards, interfaces etc. would be developed once
  - Applicable tests provided to projects, sub-contractors, vendors
  - Pay for test development once across many programs
- **Standard Tests Directly Validate Requirements**
  - Many system requirements are mapped directly to the standard tests
  - Test captures results and creates a Certificate of Conformance
  - Reduces effort spent developing/reviewing/executing test procedures
- **Added Benefit: Reduced Development & Maintenance Costs**
  - Standard tests could validate more than the specification
    - e.g. robustness, completeness, consistency, etc.
  - Standards tests can be used during development activities
  - Reduce confusion, rework and improve quality right from the start

# Targets For Standardized Tests



Standards

Common Interfaces

Data

Even Long-Duration Programs Would Benefit By Developing 'Standard' Tests



- **Ownership, Authority, Visibility**
  - Who decides what the standard test will test?
  - How do we develop and maintain these standard tests?
  - How do we ‘advertise’ the standard tests?
- **Cost Of Developing Tests**
  - Developing robust, automated tests is costly
  - Today 20-50% project costs go to test efforts
    - Minimal value is retained in today’s model
- **How To Handle Unique Features**
  - What do we do if the standard test doesn’t quite fit?
  - Designing the test to be ‘right sized’ is critical

- **Overall Cost Reduction**
  - Tests can be shared between programs
  - Less cost developing each program's tests
  - Achieve Value Persistence through reusable tests
- **Improved Quality and Consistency**
  - Standard Tests could be more comprehensive
  - Reduces interpretations and confusion
- **Could Potentially Have Effect Outside Of Test**
  - Standard tests can be provided at the start of the project and used during development
- **How do we get there?**
  - Start requiring automated tests to be delivered with the system
  - Develop a couple small, atomic tests around standards

# Why Not Create Standard Tests?

Rob Andzik  
AMERGINT Technologies  
[andzik@amergint.com](mailto:andzik@amergint.com)