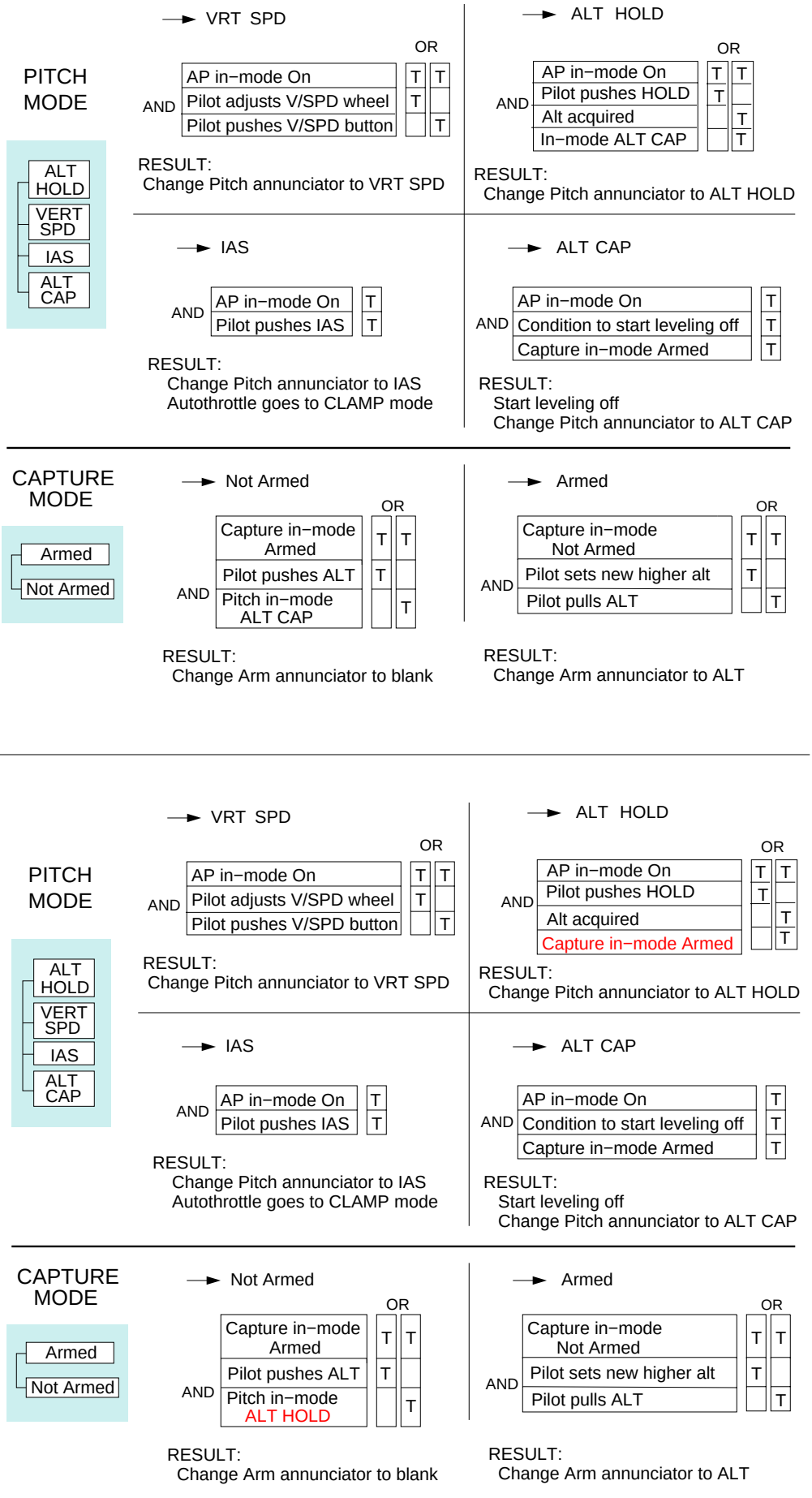

PITCH MODE
ALT HOLD
VERT SPD
IAS
ALT CAP
VRT SPD
OR
AND
AP in−mode On
Pilot adjusts V/SPD wheel
Pilot pushes V/SPD button
RESULT:
Change Pitch annunciator to VRT SPD
ALT HOLD
AND
AP in−mode On
Pilot pushes HOLD
Alt acquired
In−mode ALT CAP
RESULT:
Change Pitch annunciator to ALT HOLD
IAS
AND
AP in−mode On
Pilot pushes IAS
RESULT:
Change Pitch annunciator to IAS
Autothrottle goes to CLAMP mode
ALT CAP
AND
AP in−mode On
Condition to start leveling off
Capture in−mode Armed
RESULT:
Start leveling off
Change Pitch annunciator to ALT CAP
CAPTURE MODE
Armed
Not Armed
Not Armed
AND
Capture in−mode Armed
Pilot pushes ALT
Pitch in−mode ALT CAP
RESULT:
Change Arm annunciator to blank
Armed
AND
Capture in−mode Not Armed
Pilot sets new higher alt
Pilot pulls ALT
RESULT:
Change Arm annunciator to ALT
PITCH MODE
ALT HOLD
VERT SPD
IAS
ALT CAP
VRT SPD
OR
AND
AP in−mode On
Pilot adjusts V/SPD wheel
Pilot pushes V/SPD button
RESULT:
Change Pitch annunciator to VRT SPD
ALT HOLD
AND
AP in−mode On
Pilot pushes HOLD
Alt acquired
Capture in−mode Armed
RESULT:
Change Pitch annunciator to ALT HOLD
IAS
AND
AP in−mode On
Pilot pushes IAS
RESULT:
Change Pitch annunciator to IAS
Autothrottle goes to CLAMP mode
ALT CAP
AND
AP in−mode On
Condition to start leveling off
Capture in−mode Armed
RESULT:
Start leveling off
Change Pitch annunciator to ALT CAP
CAPTURE MODE
Armed
Not Armed
Not Armed
AND
Capture in−mode Armed
Pilot pushes ALT
Pitch in−mode ALT HOLD
RESULT:
Change Arm annunciator to blank
Armed
AND
Capture in−mode Not Armed
Pilot sets new higher alt
Pilot pulls ALT
RESULT:
Change Arm annunciator to ALT

Topics for today:

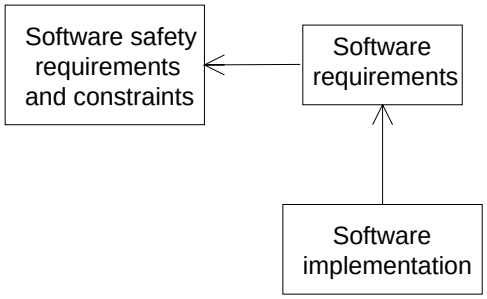
- Testing
- IV&V and Assurance
- Operations
- Management and Culture
  - New standard
  - Safety culture

Verification of Safety

Unlike the fairy tale Rumplestiltskin, do not think that by having named the devil that you have destroyed him. Positive verification of his demise is required.

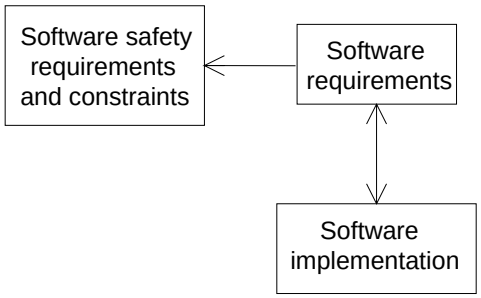
System Safety Handbook for the
Acquisition Manager, U.S. Air Force

Approaches to Verifying Safety

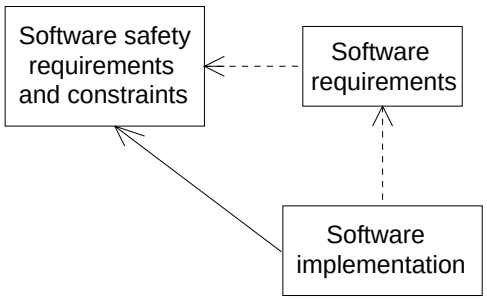


Showing the implementation satisfies the specification and the specification satisfies the safety requirements and constraints is not adequate.

Software can do more than is in the requirements.



This solves the unintended function problem but is impractical.



Verification of the implementation directly against the safety requirements and constraints.

Testing for Safety

Testing for safety starts from system safety design constraints.

Goal is to show that software will not do anything that will lead to violating the constraints.

- Test for hazardous outputs.
- Test effectiveness of any specific safety design features.
  (Write them first)
- Focus on what software will do that it is not supposed to do.
- Like any testing, specify early in development cycle and evolve as design and hazard analysis evolves.
- Test as you fly

Testing for Safety (2)

Need to test:

- Critical functions for hazardous behavior.
- Boundary conditions.
- Special features (e.g., firewalls) upon which protection of critical functions based.
- Incorrect and unexpected inputs, input sequences, and timing.
- Reaction of software to system faults and failures (environmental stress testing)
- Go/No−Go and fail−safe modes.
- Operator interface.

Role of Software Safety Engineer

- Review test plans.
- Recommend tests based on hazard analyses, safety standards, checklists, previous accidents and incidents, interface analyses
- Specify conditions under which test is to be conducted.
- Review test results for any safety−related problems that were missed in analysis or other testing.
- Monitor tests for unexpected failure modes and hazardous states.

Limitations of Dynamic Analysis

- Testing for safety is essentially intractible.
- Accidents occur only infrequently and usually in ways. not anticipated by designer or tester.
- Testing usually doesn’t find requirements errors.

Uses for Dynamic Analysis

- Limitations don't mean don't test—just that there is a limit to the confidence that can be acquired through dynamic analysis.
- Also limits to static analysis. Need testing to verify:
  - Accuracy of model to constructed system
  - Satisfaction of assumptions underlying math techniques
  - Things not covered by static analysis, e.g., performance.
- Test planning can use the results of analysis.

Formal Verification

- Will not make complex systems simple. There is no magic here.
- Limitations:
  - May be same size as program or larger.
  - Often difficult to construct.
  - Therefore, likely to contain errors.
  - Some limited aspects can be mathematically proven.

    Are these the errors most like to be found in testing?
    Are they likely to cause accidents?
- Probably little effect on safety unless aimed at safety constraints.

Independent Verification and Validation (IV&V)

- Greatest value of IV&V (and V&V) lies in interaction between developer and IV&V organizations.
  - Constant feedback ensures that quality are safety are built into system from the beginning.
  - Timeliness of findings is inversely proportional to separation of IV&V team from development team.
- Three dimensions:
  1. Orientation
  2. Scope
  3. Independence

Independent Verification and Validation (IV&V)

- Orientation
  - Development Process vs. Product
  - Usually need both
- Scope
  - Comprehensive
  - Focused (concentrate on functions most critical for safety)
  - Limited (cursory monitoring or process or limited testing)

Independent Verification and Validation (IV&V)

- Independence (most misunderstood aspect)
  - Technical Independence
    - Use personnel not involved in development
    - Need knowledge of system or background that allows them to learn quickly
    - Must formulate their understanding of problem and their proposed solution without influence from developers
    - Use or develop own set of test and analysis tools separate developer's tools (sometimes not practical)
  - Managerial
    - Responsibility vested in organization outside contractor and program organizations that develop software
  - Financial
    - Control of budget vested in organization outside contractor and program teams.

Change control

- Evaluate all changes for potential effect on safety
- Update all safety−related documentation
- Need to plan during development to make this feasible

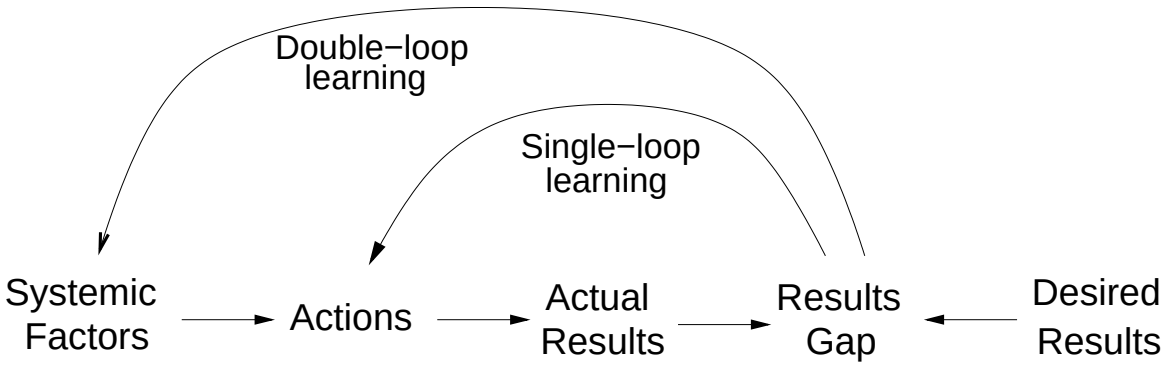
Performance Monitoring and Audits

- Data Collection
  - Use assumptions of analysis as preconditions for operations
  - Check whether operations have deviated
  - Just looking at incidents is not enough
- Data Analysis
  - Always perform root cause analysis
  - Do not assume problems are caused by hardware
  - Look at why operators made mistakes or deviated from written procedures (maybe it is system design that needs to be changed)

Performance Monitoring and Audits (2)

- Information Dissemination and Use
  - Integrate information into decision making tools and environment
  - Blame and discipline can lead to lack of reporting of problems
  - Distinction between a fixing orientation and a learning one

Single loop vs. Double loop learning



Organization/Management/Culture

It is not enough to talk about ‘‘absolute safety’’ and of ‘‘zero accidents.’’ There must also be proper organisation and management to ensure that actions live up to words.

British Department of Transport
Investigation into the Clapham Junction Railway Accident

Management

The new NASA software safety standard contains everything I would have said about management.

- What you have learned in this class provides the information you will need to implement it.
- Two minor comments:
  - 5.15.3 Project has final sign−off authority for waivers with concurrence of S&MA director and software safety expert.

    CAIB report recommended an Independent Technical Authority be responsible for all waivers.
  - Section 7: I would require root cause analysis rather than "recommend" it.

Safety Culture and Management

Definitions of "culture"

- A shared set of norms and values
- A way of looking at and interpreting the world and events around us (our mental model) and taking action in a social context.
- An ongoing, proactive process of reality construction (Morgan)
  - Organizations are socially constructed realities that rest as much in the heads of members as in sets of rules and regulations.
  - Sustained by belief systems that emphasize the importance of rationality: the "myth of rationality"

    "helps us to see certain patterns of action as legitimate, credible, and normal, and hence to avoid the wrangling and debate that would arise if we were to recognize the basic uncertainty and ambiguity underlying many of our values and actions."

Safety Culture and Management (2)

- Safety culture is subset of culture that reflects general attitude and approaches to safety and risk management.
- Trying to change culture without changing environment in which it is embedded is doomed to failure
- Simply changing organizational structures may lower risk over short term, but superficial fixes that do not address the set of shared values and social norms are likely to be undone over time.
- "Culture of denial"
  - Risk assessment unrealistic and credible risks and warnings are dismissed without appropriate investigation.

Safety Culture and Management (3)

- Need to bring operational practices and values into alignment
  - Identify desired organizational safety principles and values
  - Establish organizational infrastructure to achieve values and sustain them over time.
    - Understand why operational practices have deviated from stated principles.
    - Make adjustments
    - Institute protections against future misalignments
- Goal is to create a culture and organizational infrastructurw that can resist pressures against applying good safety engineering practices and procedures
- No one single safety culture in a large organization

Organizational Structure and Culture

- Structure drives behavior
- Where should safety activities be put?
  - Safety permeates every part of development and operations (e.g., engineering design, risk management, IV&V, quality assurance, operational performance monitoring, maintenance)
    - Need not be located in one place but common methods and approach will strengthen these disciplines
    - If distributed, need a clear focus and coordinating body
  - Basic principles:
    1. System Safety needs a direct link to decision makers and influence on design−making (influence and prestige).
    2. System Safety needs to have independence from project management (but not engineering)
    3. Direct communication channels are neededto most parts of the organization (oversight and communication)

Organizational Structure and Culture (2)

- Influence and prestige
  - Weak matrix structure and placement only in assurance org. has diminished influence and prestige and thus impact on decision−making.
- Independence
  - Project manager "purchases" safety
    - Raises conflict of interest problems
    - Limited to what and how much project manager wants and can afford.
  - Independent safety reviews and alternative reporting channels between levels have been taken over by Project Office.

    e.g. SSRP
  - Independent authority inside and outside program and projects

---

Organizational Structure and Culture (3)

- Independent Technical Authority (CAIB)
  - Inside program but independent of Program Manager and schedule/budget concerns
  - Outside program to authorize and provide:

    Tailoring or relaxing of safety standards

    Amount and type of safety to be applied to program

    Standards creation

    External safety review

    e.g. WSESRB

Organizational Structure and Culture (4)

- Oversight and Communication
  - Oversight vs. insight (transition usually done poorly)
  - Responsibility for safety cannot be delegated to contractors
  - Use of working groups for communication
    - Very effective in DoD
    - Different groups at different levels
    - Responsible for coordinating safety efforts at each level, reporting status of outstanding safety issues, providing information to other levels and to external review boards.

Organizational Subsystems and Social Interaction

- Lots of subsystems affect safety culture
  - Communication systems, information systems, reward and reinforacement systems, hiring and promotion, learning and feedback systems, career development, complaint and conflict resolution systems, etc.
- Subsystems intertwined with social interaction processes
  - Leadership, teamwork, negotiations, problem solving, decision−making, partnership, entrepreneurship, etc.
- Will consider just two that are very important
  - Communication and leadership
  - Safety information systems

Communication and Leadership

- Need culture of openness and honesty where everyone's voice is valued.
  - Employees need to feel they will be supported by management
- Need to create incentives and reward structures that encourage proper tradeoffs between safety and other goals
- Informal rules (social processes) as well as formal processes must support safety policy.
- Managers need to display leadership on safety issues and eliminate barriers to dissenting opinions.
- Simply creating new communication systems not enough
  - Need to change management style
  - Need to change attitudes about communication about safety

---

Safety Information Systems

- Good decision making about risk is dependent on having appropriate information.
  - Without it, decisions made on basis of past success and unrealistic risk assessment.
- Need a culture that values the sharing of knowledge gained from experience
- Reports have said NASA does not have such a learning culture
  - Learning across centers and even programs often problematic
  - Data often not collected
  - Data that is collected is often filtered and inaccurate
  - Little analysis and summarization of causal data
  - Information not provided to decision makers in way that is meaningful and useful to them
    - Should be integrated into the decision−making environment
- Operational experience should provide feedback on process

Capability and Skills

- NASA losing system safety skills
- Difficulties in moving from data to information to action
  e.g., seeing the O−rings and foam as problems.
  - "Hindsight Bias"
    - Vaughan − Normalization of Deviance
      - Says some risks had come to be seen as "normal"
      - Never defines what "normal" is
      - Problem really was that some factors had come to be seen as acceptable risk without adequate supporting data
    - Tufte − problem was in display of data
      - Easy to see what to display only after know the answer

---

Capability and Skills (2)

- Need to evaluate decisions in context made
  - With respect to information available at the time decision made (not afterward)
  - Along with organizational factors influencing interpretation of the data and information
- After an accident, always easy to separate signal from noise
  - Good system safety engineering can be used to do this before an accident.
  - But need an culture that allows it to operate effectively.
- Extrapolation from limited data

Conclusions

We pretend that technology, our technology, is something of a life force, a will, and a thrust of its own, on which we can blame all, with which we can explain all, and in the end by means of which we can excuse ourselves.

T. Cuyler Young
Man in Nature

System Safety Process

Safety must be specified and designed into the system and software from the beginning.

- Program/Project Planning
  - Develop policies, procedures, etc.
  - Develop a system safety plan
  - Establish management structure, communication channels, authority, accountability, responsibility
  - Create a hazard tracking system
- Concept Development
  - Identify and prioritize system hazards
  - Generate safety−related system requirements and and design contraints

System Safety Process (2)

- System Design
  - Apply hazard analysis to design alternatives
    - Determine if and how system can get into hazardous states
    - Eliminate hazards from system design if possible
    - Control hazards in system design if cannot eliminate
    - Identify and resolve conflicts between design goals
  - Trace hazard causes and controls to components (hardware, software, and human)
  - Generate component safety requirements and design constraints from system safety requirements and constraints

System Safety Process (3)

- System Implementation
  - Design safety into components
  - Verify safety of constructed system
- Configuration Control and Maintenance
  - Evaluate all proposed changes for safety
- Operations
  - Incident and accident analysis
  - Performance monitoring
  - Periodic audits

Software Safety Tasks

- Establish software safety management structure, authority, responsibility, accountability, communication channels, etc.
- Develop a software hazard tracking system and link to. system hazard tracking system.
- Trace identified system hazards and system safety design constraints to software interface.
- Translate identified software−related hazards and constraints into requirements and constraints on software behavior.
- Evaluate software requirements with respect to system safety design constraints and other safety−related criteria.

Software Safety Tasks (2)

- Design software and HMI to eliminate or control hazards. Design safety into the software.

  Defensive programming
  Assertions and run−time checks
  Separation of critical functions
  Elimination of unnecessary functions
  Exception handling
  etc.

- Analyze the behavior of all reused and COTS software for safety (conformance with safety requirements and constraints)

- Trace safety requirements and constraints to the code. Document safety−related design decisions, design rationale, and other safety−related information.

Software Safety Tasks (3)

- Perform special software safety analyses
  e.g.
  human−computer interaction and interface
  formal or informal walkthroughs or proofs
  interface between critical and non−critical software

- Plan and perform software safety testing. Review test results for safety issues. Trace identified hazards back to system level.

- Analyze all proposed software changes for their effect on safety.

- Establish feedback sources. Analyze operational software and relate to hazard analysis and documented design assumptions.

Conclusions

- There are no simple solutions. Requires:
  - Time and effort
  - Special knowledge and experience
- Our most effective tool for making things safer is simplicity and building systems that are intellectually manageable.
- Complacency is perhaps the most important risk factor.
- Safety and reliability are different—don't confuse them.

Conclusions (2)

- The safety of software cannot be evaluated by looking at it alone. Safety can be evaluated only in the context of the system in which it operates.
- Building safety into a system will be more effective than adding protection devices onto a completed design.
- The job of the system safety engineer is to identify safety constraints and ensure they are enforced in design and operations. System safety must work closely with the system designers.
- Placing all responsibility for safety on human operators does not ensure safety. It merely provides a scapegoat.

Conclusions (3)

- Our technology must be used by humans. Human error can be reduced by appropriate design.

- The earlier safety is considered in development, the better will be the results.

- To prevent accidents, we will need to remove systemic factors.

  Concentrating only on technical issues and ignoring managerial and organizational deficiencies will not result in effective safety programs.

- Safety is a system problem and can only be solved by experts in different disciplines working together.

A life without adventure is likely to be unsatisfying, but a life in which adventure is allowed to take whatever form it will, is likely to be short.