Reliability/Dependability

• NATO 1968: impending “crisis” perceived
• Hardware was becoming more reliable through increased redundancy
• Software was becoming less reliable through increased complexity
  – For some applications, failure = death
  – But for others, failure is tolerated by users
• There will always be some bugs left!
  – A need for “fault tolerance”

Reliability/dependability

• Results:
  We have somewhat standard techniques / processes now and some people even use them.
  Highly reliable systems have been built.
• Open problem: Quantifying increase in confidence.
  How much do they help, and how much confidence do you need?
Reverse engineering

• Design info is known to be necessary for system evolution (and all systems evolve)
• But may be unavailable for legacy systems.
  – Sometimes you can interrogate the creators
  – Sometimes all you have left is the artifact
• Some of the human effort could be assisted
  – A need for “program comprehension”

Reverse engineering

• Results:
  – Some tools that assist in human understanding of programs via parsing and visualization.
• Open problem:
  – People don’t use them much yet. They use simpler things like grep. Slow adoption, or failure to address the “right” problems?
  – Remove the need for last-ditch efforts by maintaining/carrying design documentation?