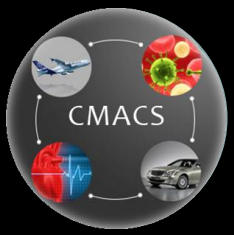
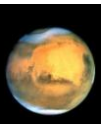
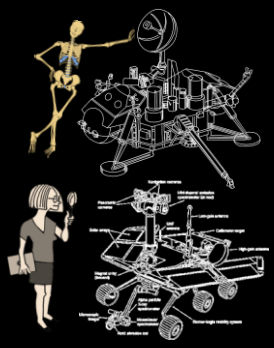
analysis of complex software

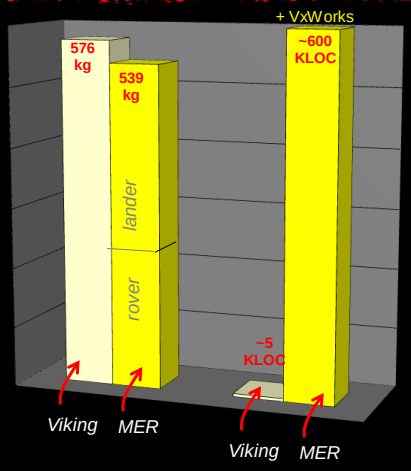
gerard holzmann
jpl laboratory for reliable software
gh@jpl.nasa.gov



+30 [Viking (1976) and MER (2004)]







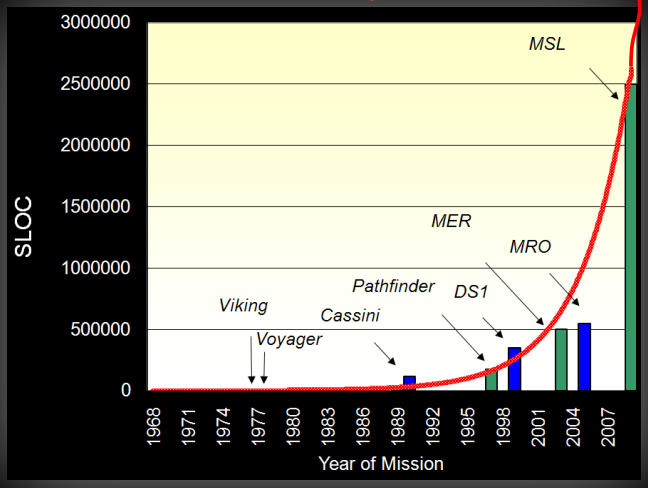
same destination (mars)
same nr of instruments (~8)
similar landed mass (~550kg)
same # person-months

two *orders of magnitude* more code

2

aerospace software

robotic spacecraft



the code size has grown

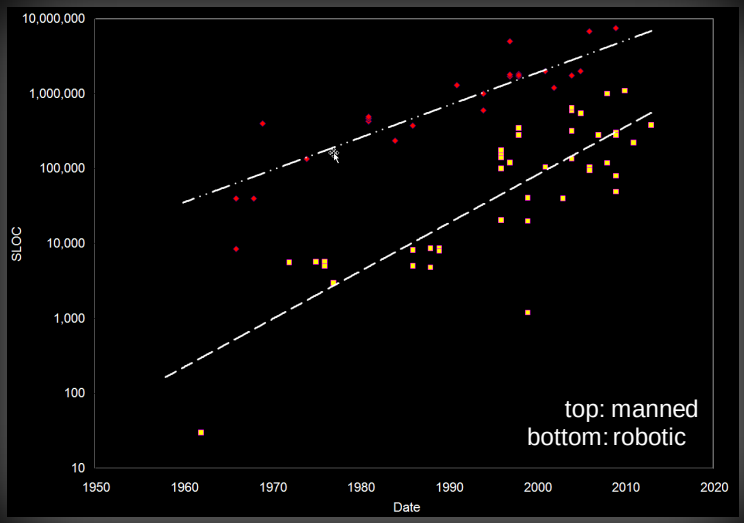
after a 2-year postponement of the mission in 2009:

software grows with time

3

it is a general trend

(all aerospace software – military, commercial, space)



4

automotive software

example: the Mercedes-Benz S-Class 2003

- ***more than 50 embedded controllers***
- ***more than 600,000 lines of code***

IBM claims that approximately 50 percent of car warranty costs are now related to electronics and their embedded software, costing automakers in the United States around $350 and European automakers €250 per vehicle in 2005.

In 2005, Toyota voluntarily recalled 160 000 of its 2004 and some early 2005 model year Prius hybrids because of a software problem that caused the car to suddenly stall or shut down. The time needed to repair the software was estimated at about 90 minutes per vehicle, or about 240 000 person-hours. Even at cost, that is a lot of money.

- ***thousands of signals***
- ***network of three bus systems***

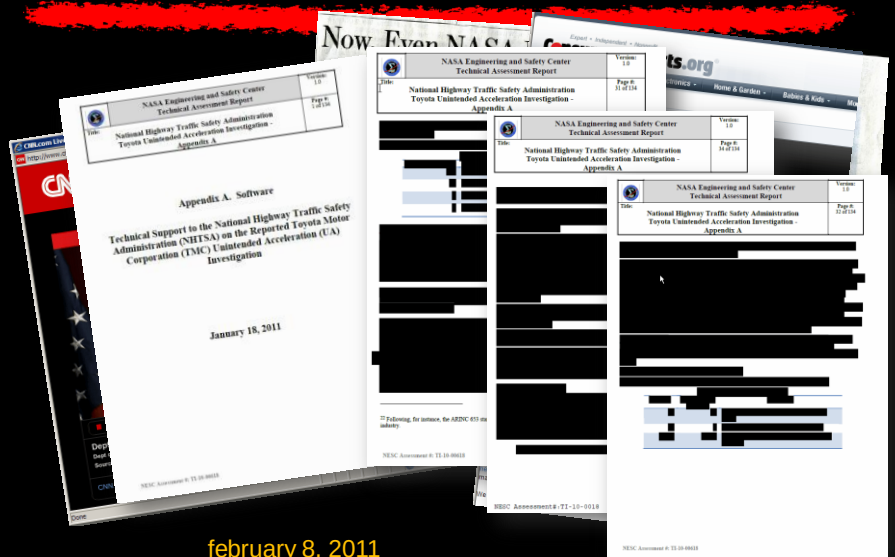
Proc. 25th Int. Conf. on Software Eng., 2003, Portland, Oregon, pp 498-503

http://www.spectrum.ieee.org/feb09/7649

5

unintended acceleration

our investigation of possible software causes



february 8, 2011

6

our plan

- *objective:* scale logic model checking techniques to handle complex software applications (as used in *automobiles*, *spacecraft*, *power plants*)
  - *significantly* improve over currently used methods for software testing
- *method:* leverage grid/cloud/multi-core verification techniques and search randomization techniques
  1. develop new efficient algorithms, and prove them correct
  2. *make them trivial to use by any software developer*
  3. evaluate them on complex mission-critical spacecraft software
- *metric: quantifiable* improvements over the currently used verification methods for complex software systems in the target domain (aerospace & automotive)

logic model checking

swarm verification

tackle complex systems

7

thank you!

LARS
NASA/JPL Laboratory for Reliable Software

8