Software Engineering

What is Software?

30 August, 2016

What is Software Engineering?

The IEEE Std. 610.12-1990's Standard Glossary of Software Engineering Terminology defines 'software engineering' as follows:

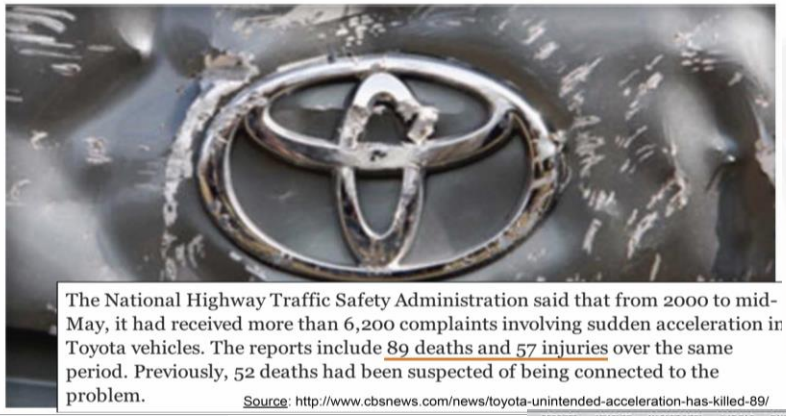
(1) the application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software: that is, the application of engineering to software.

(2) the study of approaches as in (1).

Poor SW Quality Threatens Society

By CBSNEWS / AP / May 25, 2010, 7:08 PM

Toyota "Unintended Acceleration" Has Killed 89

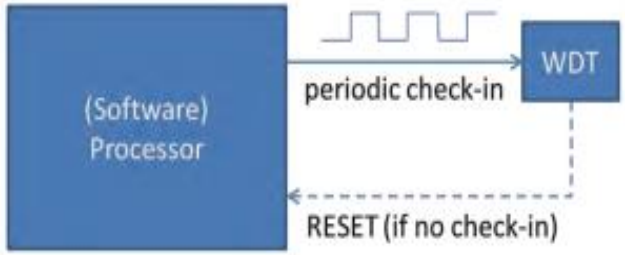


The National Highway Traffic Safety Administration said that from 2000 to mid-May, it had received more than 6,200 complaints involving sudden acceleration in Toyota vehicles. The reports include 89 deaths and 57 injuries over the same period. Previously, 52 deaths had been suspected of being connected to the problem.

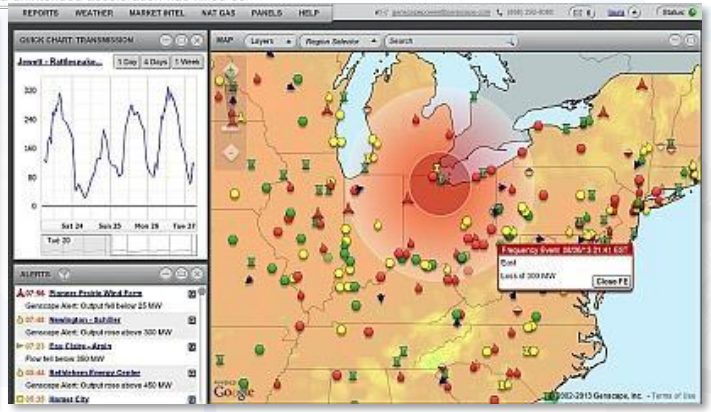
Source: http://www.cbsnews.com/news/toyota-unintended-acceleration-has-killed-89/

A "watchdog timer" is hardware to auto-reset software

- Healthy software should periodically "check-in" to prevent reset

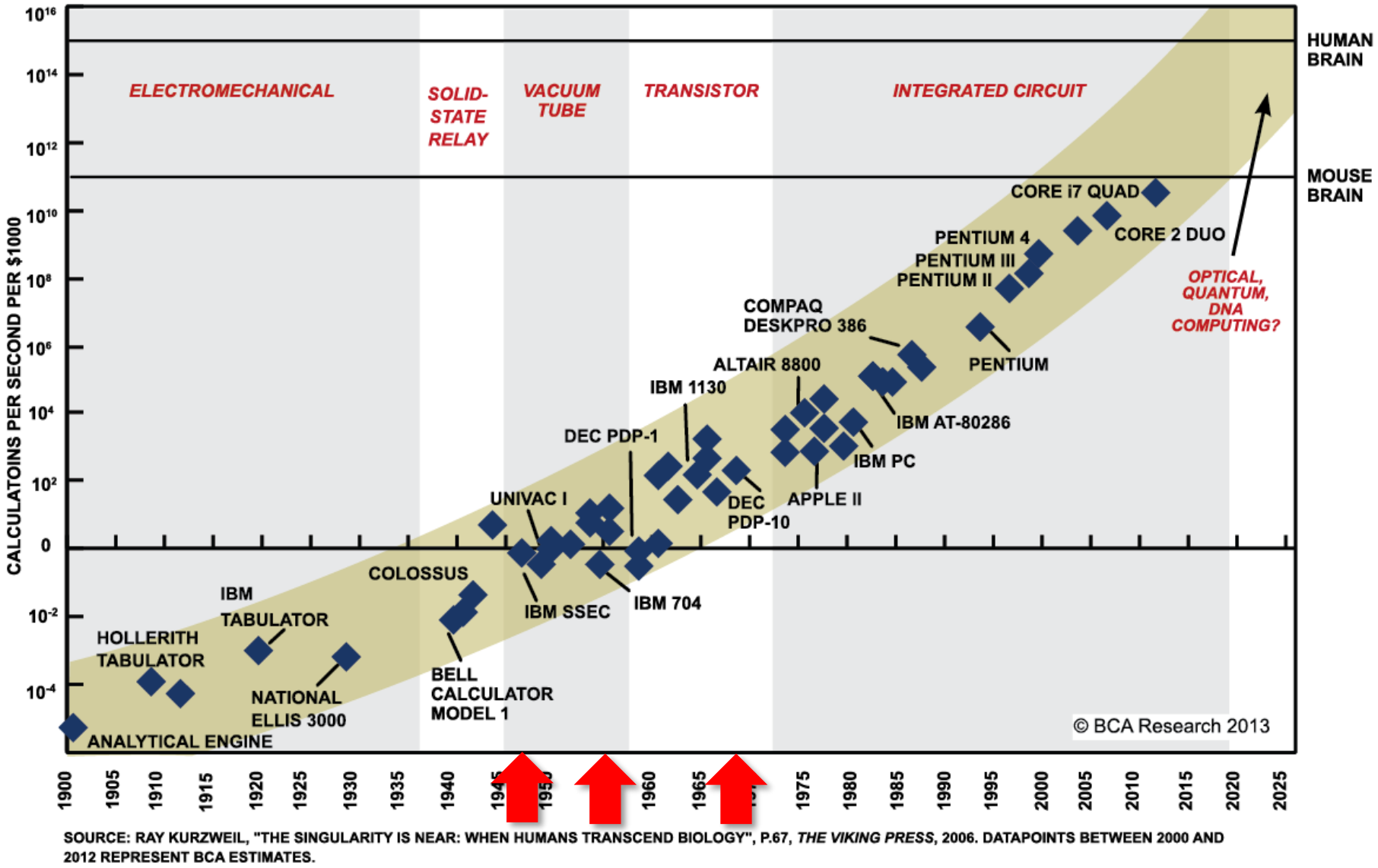


With multiple tasks, health of all tasks must be checked

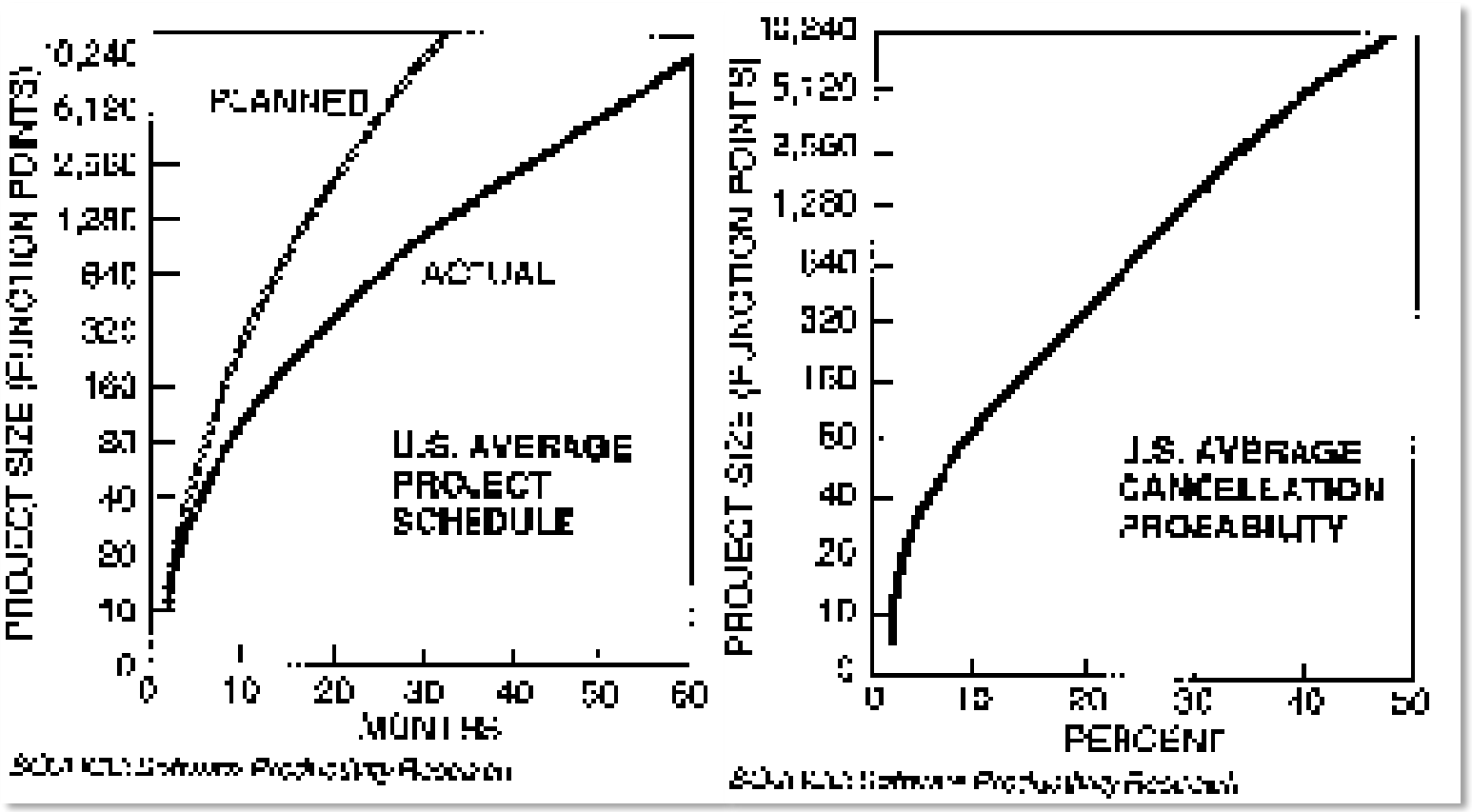


P. Koopman: A Case Study of Toyota Unintended Acceleration and Software Safety, 2014

M. Barr: Killer Apps-Embedded Software's Greatest Hit Jobs, 2014



SOURCE: RAY KURZWEIL, "THE SINGULARITY IS NEAR: WHEN HUMANS TRANSCEND BIOLOGY", P.67, *THE VIKING PRESS*, 2006. DATAPOINTS BETWEEN 2000 AND 2012 REPRESENT BCA ESTIMATES.


PLANNED
ACTUAL
U.S. AVERAGE
PROJECT
SCHEDULE
0
10
20
30
40
50
60
MONTHS
U.S. AVERAGE
CANCELLATION
PROBABILITY
PERCENT

The NATO Software Engineering Conference, 1969

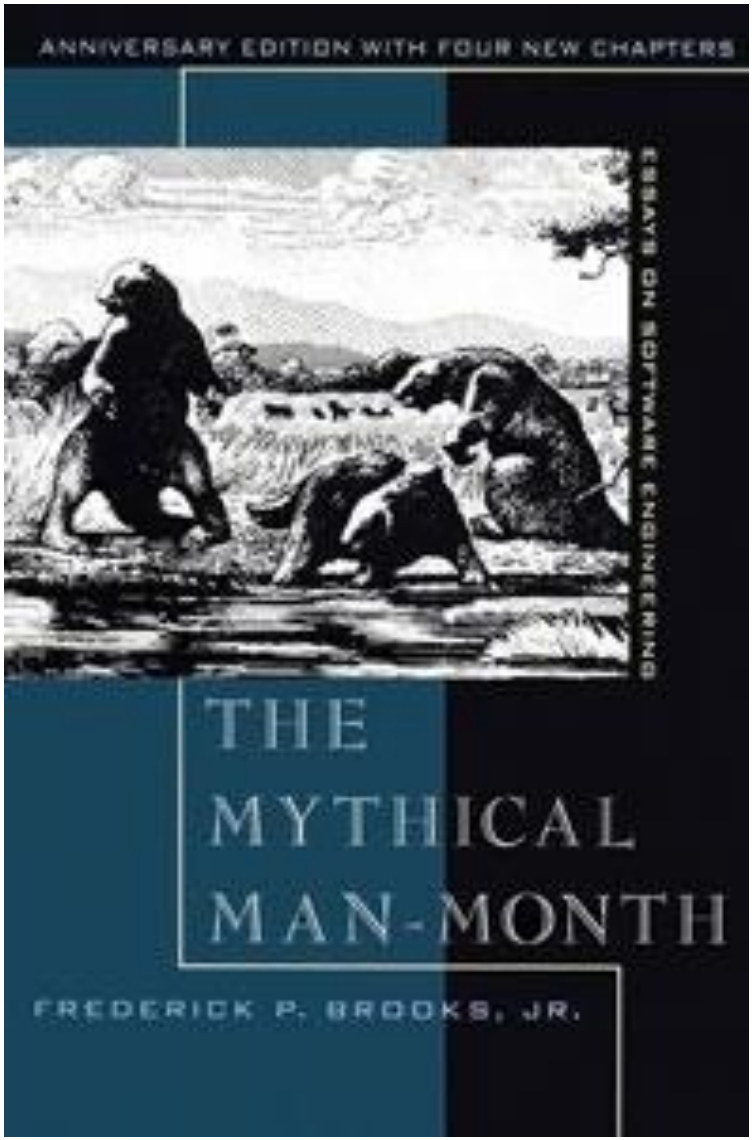
The major cause of the **software crisis** is that the machines have become several orders of magnitude more powerful!

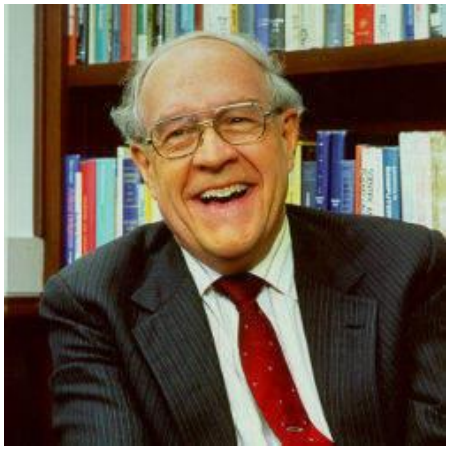
To put it quite bluntly: as long as there were no machines, programming was no problem at all; when we had a few weak computers, programming became a mild problem, and now we have gigantic computers, programming has become an equally gigantic problem.

- Edsger Dijkstra, The Humble Programmer (EWD340), Communications of the ACM

It is (software) not made with a clean fabrication process. … What se need is "software engineering".

- Friedrich L. Bauer





* most chapters can be found at Google Books

What is Software? (1/2)

- A part of a computer system, that consists of instructions to operate hardware for providing certain functionalities
  - traditional definition by John Tukey at 1958
  - what is clear distinction between S/W and H/W?
- "Software" in other domains

소프트웨어는 명령어와 기능을 하면 software 가 된다. 순서와 절차가 있기 때문에 software 라고 볼 수 있다.

What is Software? (2/2)

소프트웨어의 특징

- An artifact that represents intellectual processes for a certain functionality as an executable/interpretable form
  - Software is an abstraction of mental activities

인간의 뇌 활동, abstract하는 과정임

- Characteristics of software
  - software deals with problems in human society
  - software is often complicated
  - software often has a structure
  - software is mostly a logical entity and has no physical form

Characteristics of Large Software System [Brooks'86]

- Complexity
  - no two parts are alike. 다른 소프트웨어가 똑같을 수가 없다.
  - the complexity of a software artifact increases non-linearly with the size of the artifact.
  - there is no single way to abstract software artifacts perfectly.
    소프트웨어를 추상화 하는 방법이 한가지만 있는 것이 아님
- Conformity 세상에 있는 것들을 다룬다
  - software artifacts need to confront arbitrariness of many human artifacts and human society.
- Changeability
  - software artifacts are intangible.
  - software artifacts are constantly subject to pressure of change.
- Malleability 상황에 따라 유연하게 변화헤애 한다
  - Software artifacts can quickly become extremely complicated and expensive to change correctly.

The Nature of Software Engineering

- The system specification is mostly unknown at beginning.
- Software engineering involves multiple persons building multiple versions of software.
- Software engineering is to manage software changes.
- Software engineering is to generate not only a program itself but also associated documentation, libraries, data, tools that are needed to make the program useful.

소프트웨어 엔지니어링이라는 분야에 코딩만 있는 것이 아니라 문서화 등 다양한 분야가 있는 것이다

- Software engineering is essentially modelling large and complex intellectual processes, which requires intellectual effort, creativity and time.

Role of Software Engineer

- The software engineer must develop skill to build a variety of models and to reason about those models in order to guide choices of many trace-off at software development.

- Software engineering itself is "software", and thus, software engineers are asked to manage their processes in a systematic and automated methodologies.

소프트웨어를 왜 그렇게 만들었냐(했느냐)에 대해 말해줄 수 있어야 함.
프로세스를 관리해야 하고 엔지니어 자체가 알아서 자동화된 방법론을 적용해야 한다 automated methodologies

References

- Leon J. Osterweil, What is software?, Automated Software Engineering, 15, 2008.
- Frederick P. Brook Jr., No Silver Bullet, IFIP World Computing Conference, 1986.
  - Chapter 16 of *The Mythical Man-Month*, 1995 ed.
- Roger Pressman and Bruce Maxim, *Software Engineering: A Practitioner's Approach*, 8/e, Chapter 1.
- Carlo Ghezzi et al., Fundamentals of Software Engineering, 2/e, 2002.
- Ian Sommerville, *Software Engineering*, 10/e., Chapter 1.