

Problematic in Software Engineering

Alexandre Bergel
abergel@dcc.uchile.cl

Outline

1. Software engineering or Art of programming
2. Historical aspects
3. The Software Crisis
4. Some answers to this crisis



1. Software engineering or
Art of programming

As for any production process, producing software requires
methodologies, techniques and tools

This goes well beyond programming and is commonly referred as
Software Engineering

“Software Engineering” was first coined during
the 1968 NATO conference

The first conference on Software Engineering was in 1973

IEEE TSE appeared in 1975

The NATO conference was held in Garmish

IEEE Transaction on Software Engineering is the leading scientific journal of the field.

Programming is a new discipline that is purely intellectual

It has also been qualified as Art

Strange habits in the field

Donald Knuth authored “The Art of Software Programming”

Probably because Software Engineering is a new discipline that is essentially intellectual property, it is not rare that novice programmers prefers to write his/her own piece of code instead of tacking a similar code written by somebody else.

A software **modifies** the environment of the users

It is therefore **difficult** to obtain **stable requirements**

Software maintenance plays an important rôle

Corrective maintenance: bug fixing

Evolutive maintenance: adapting or perfecting

It is largely recognized by industries that maintenance costs up to 67% of the total cost of a software product. Where 48% are bug fixing. 60% of faults result from mis- and under-specification.

2. Historical Aspects

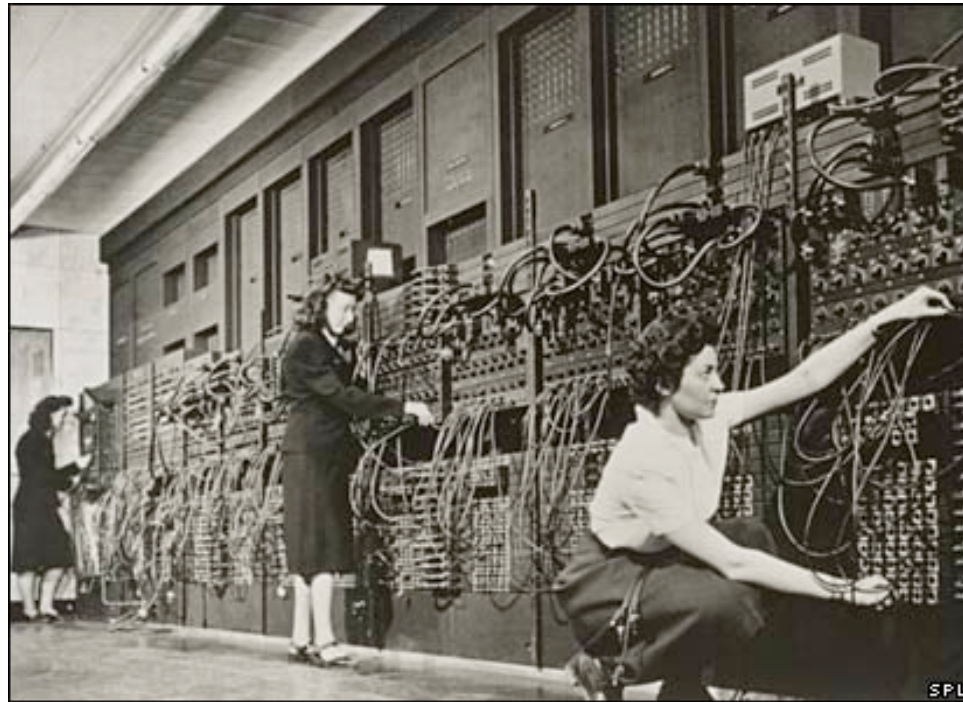
Programming appeared during 1945-1958

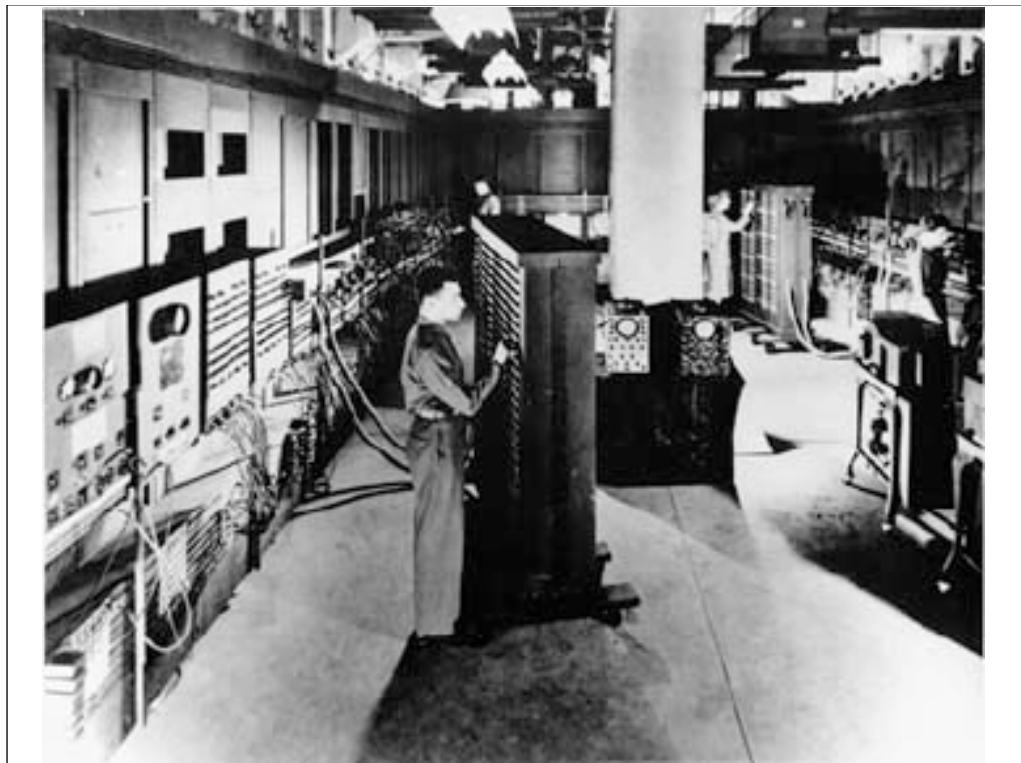
Essentially for engineer in mathematics

Computation performed on huge machines with low
power

experimental: ENIAC, EDVAC

commercial: SEA CAB500, BULL GAMMA3, UNIVAC, IBM 701





Development during 1958-1975

Major development

scientific computing with Fortran

business, finance, administrative with Cobol

Multi-user systems appears

IBM 360, VAX, Bull Gecos

Effort were essentially focused on hardware

Many programs developed at that time were concerned with the 2000-year bug

1975-1990

Important **development of ICT systems** essentially due to improvement of database and networks

Price of hardware decrease while **cost of software increase**

Decentralization of systems

Software gains an **important place** in ICT systems

Crisis

3.The Software Crisis

The Software Crisis

Formally identified during the NATO conference in 1968

Term used to describe

the impact of rapid increases in computer power and

the complexity of the problems which could be tackled

In essence, it refers to the difficulty to write correct, understandable and verifiable computer programs

The roots are complexity, expectations and changes

The term "software crisis" was coined by F. L. Bauer at the first NATO Software Engineering Conference in 1968 at Garmisch, Germany.





Tony Hoare: quicksort algorithm, first axiomatic basis for programming, notation and Communicating Sequential Processes



Edsger W. Dijkstra

“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

An early use of the term is in Edsger Dijkstra's 1972 ACM Turing Award Lecture

The causes of the software crisis were linked to the overall complexity of the software process and the relative immaturity of software engineering as a profession.

The crisis manifestation

Projects running over-budget

Projects running over-time

Software was very inefficient

Software was of low quality

Software often did not meet requirements

Projects were unmanageable and code difficult to maintain

Software was never delivered

Few examples

Bill of 0 French Francs

False missile attack (Nov 1979)

Anti-patriot missile (Gulf War 1991)

Lost spacial ship toward Venus (60's), Mars (1999)

Shutdown of ATT (1990)

Ariane 5 (1996)

French Post (1998)

Bug 2000 Year

A non exhaustive list: <http://www.cs.tau.ac.il/~nachumd/horror.html>

Taming the software crisis

Various processes and methodologies have been developed

... with varying degrees of success

4. Some answers to this crisis

Mastering the development process

Development phases

requirements, specification, schedule, implementation, testing, ...

Intermediary steps

prototypes, documentations

Criteria for changing product stages

Managing projects

A lot of effort before starting to program (requirements, features analysis, ...)

CASE Methodology

CASE = Computer Aided Software Engineering

“Method and quality”

ADA language (1983)

advanced compiler

reusable library

A number of large research projects are founded:

Europe: ESPRIT, Alvey, Software Factory

USA: ADA, Stars, SDI

Japon: 5th Generation (ICOT)

The ADA language was requested by the DOD (North American Department of Defense) leads this rationalization of Software Development with a

Object Paradigm

During the 1990, the Crisis is still here!

Structured programming did not scale as expected

what is true for 50.000 LOC is not true for 5.000.000

Software are more and more complex

Obsolescence is a real problem

Simula and Smalltalk pioneered the notion of Object

More recent approaches

A number of techniques emerged

Component based programming

Generative programming

Aspect-oriented programming

Scripting languages

Today, we know the Software Crisis better

Its impacts and syndrome are better controlled