FORMAL METHODS IN SOFTWARE ENGINEERING

TODAY'S AGENDA

- Introduction to the Course
- Objectives of the course
- Basic Concepts

COURSE INTRODUCTION

- **Course Title:** Formal Methods
- **Total Credits:** 3 Credits

3 Credits Theory

- **Pre-requisite:** Discrete Structures

Course Learning Outcomes

- **CLO 1:** Describe the costs and benefits of formal methods.
- **CLO 2:** Construct formal models of sequential software systems
- **CLO 3:** Implement sequential software systems based on formal models.

RECOMMENDED BOOKS

1-Concise Guide to Formal Methods, Theory, Fundamentals and Industry Applications, 2017, Springer

2-Using Z Specification, Refinement, and Proof by Jim Woodcock and Jim Davies, 2000

GRADING SCHEME

- Quizzes:10%
- Assignments: 10%
- Mid Term:30%
- Final:40%
- Project/Presentation:10%

CLASSROOM CODE

6kupler

FORMAL METHODS

Formal method is a way to takes the specification (written in natural language) and converts it into its mathematical equivalent.

It is normally used in the SDLC Analysis and Design stages.

- Formal method will also bring to light all different probable perspective to any given variables and functions that could have been hidden behind the English language.

Formal Method

- Formal method is branch of software engineering, in which we analyze software systems.
- Develop a program in a way that each step leads to a final solution, follow proper method to make sure that we do not take wrong steps.
- The Encyclopedia of Software Engineering defines formal methods in the following manner:
  - *Formal methods used in developing computer systems are mathematically based techniques for describing system properties. Such formal methods provide frameworks within which people can specify, develop, and verify systems in a systematic, rather than ad hoc manner*.

Formal Method definition

- *A method is formal if it has a sound mathematical basis, typically given by a formal specification language. This basis provides a means of precisely defining notions like consistency, completeness, and more relevantly specification, implementation and correctness.*

- Correctness, the property that an abstract model fulfills a set of well defined requirements.
- Consistency, to be consistent, facts stated in one place in a specification should not be contradicted in another place.
- Used to specify programs, what the system is suppose to do.
- Used for constructing programs.
- Used to verify the program.

Why Formal Methods are required

- History of software
  - Softwares encountered notorious bugs that were the cause of financial lose and deaths of many people.
  - Famous bugs are
    - Therac-25
      - Computerized radiation therapy machine called the Therac-25. Killed many people, controller could not stop radiation due to software bug.
    - AT & T long distance breakdown bug 1990
      - Ill placed break statement in the code, caused the 1/3rd of entire American network to go down for 9 hours.

Why Formal Methods are required

- Patriot Missile Failure Gulf war
  - This missile hit the own American troops a software defect in 1991. Killing 28 people and many injured
- Pentium bug
  - Software error in microcode of Pentium microprocessor, which resulted in error of floating point calculation problems. Intel had to take back all the Pentiums, and it caused huge loss.

Lessons Learnt

- Testing programs was not enough to find all bugs
- Pre and Post conditions may help
  - But identifying exhaustive conditions is not straightforward as well
- What about larger programs?
  - Windows 8????

Program testing can be a very effective way to show the presence of bugs, but it is hopelessly inadequate for showing their absence.

Edsgar W. Dijkstra

Computer Arithmetic is Imprecise

- Example
  - $[[((X^{1/2})^{1/2}]^2]^2 = X$

- $85^{1/2} = 9.219544457292887$
- $9.219544457292887^{1/2} = 3.036370276710811$
- $3.036370276710811^2 = 9.219544457289$
- $9.219544457289^2 = 85.00000000000005$

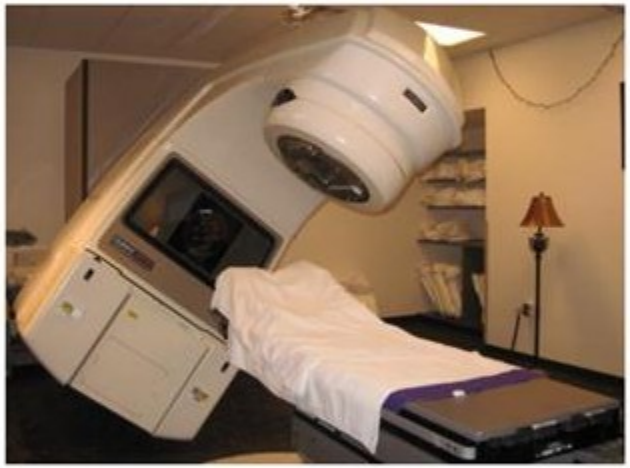
Safety-Critical Systems

- Accuracy is Extremely Important
  - Failure can cause loss of life or severe injury

Software

Faulty systems can be disastrous

- Therac-25



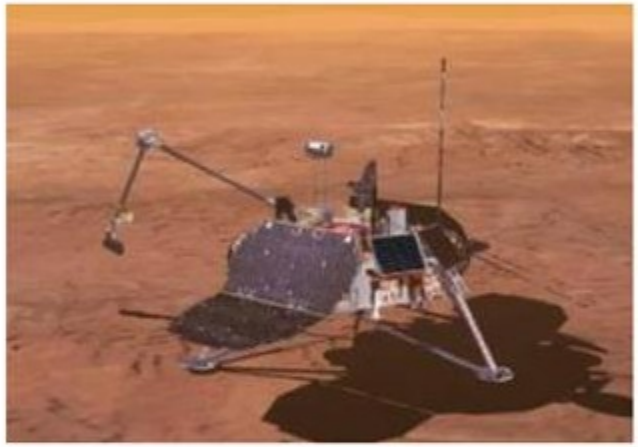
- Software Bug in a Cancer Therapy Machine
- 3 Deaths and 3 severe injuries between 1985-87

- Air France Flight 447 crash into the Atlantic ocean

- Inaccurate measurement of airspeed
- Resulted in a loss of 228 lives

System Analysis Accuracy

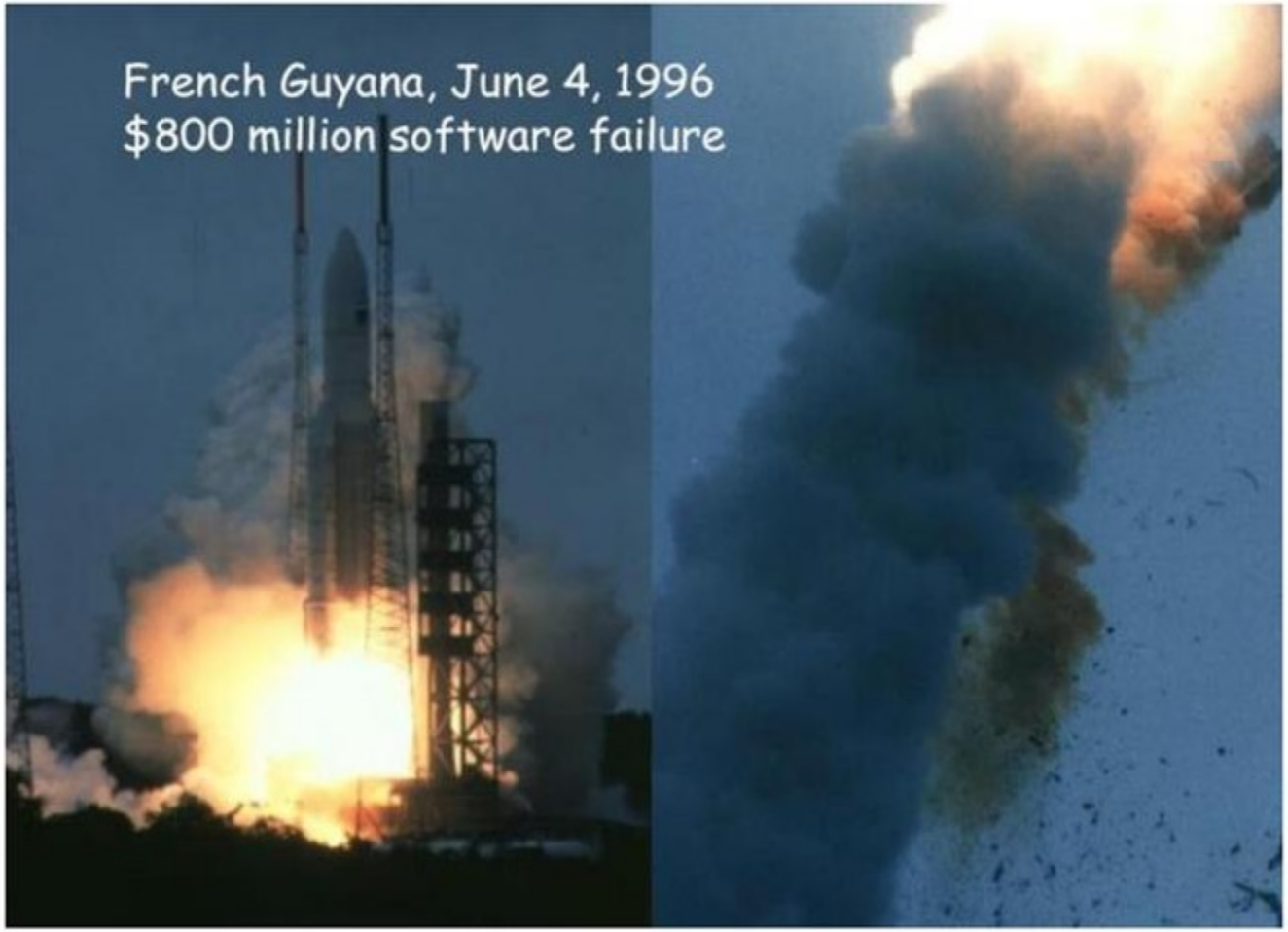
- Faulty systems can be disastrous
  - Mars Polar Lander



  - Engine shutdown due to spurious signals that gave false indication that spacecraft had landed Mars
  - Resulted in a loss of US $370M in 1999
  - Mars Climate Orbiter

  - Conversion error from English units to metric units
  - Resulted in a loss of US $125M in 1999

Faulty systems can be disastrous



Software still running on Safety-Critical Systems

Formal Verification Methods

- Based on Mathematical techniques
  - Construct a computer-based mathematical model of the program (*implementation*)
  - Use mathematical reasoning to check if the implementation satisfies the properties of interest (*specifications*) in a computerized environment

Formal Verification Research

- Requires strong understanding of
  - Logic
  - Mathematics
  - Computer Programming
  - Understanding of the system that requires to be analyzed
- A number of open research issues BUT very challenging!

Scientists Quotes

Software engineers want to be real engineers.

Real engineers use mathematics.

Formal methods are the mathematics of software engineering.

Therefore, software engineers should use formal methods.

(Mike Holloway, NASA)