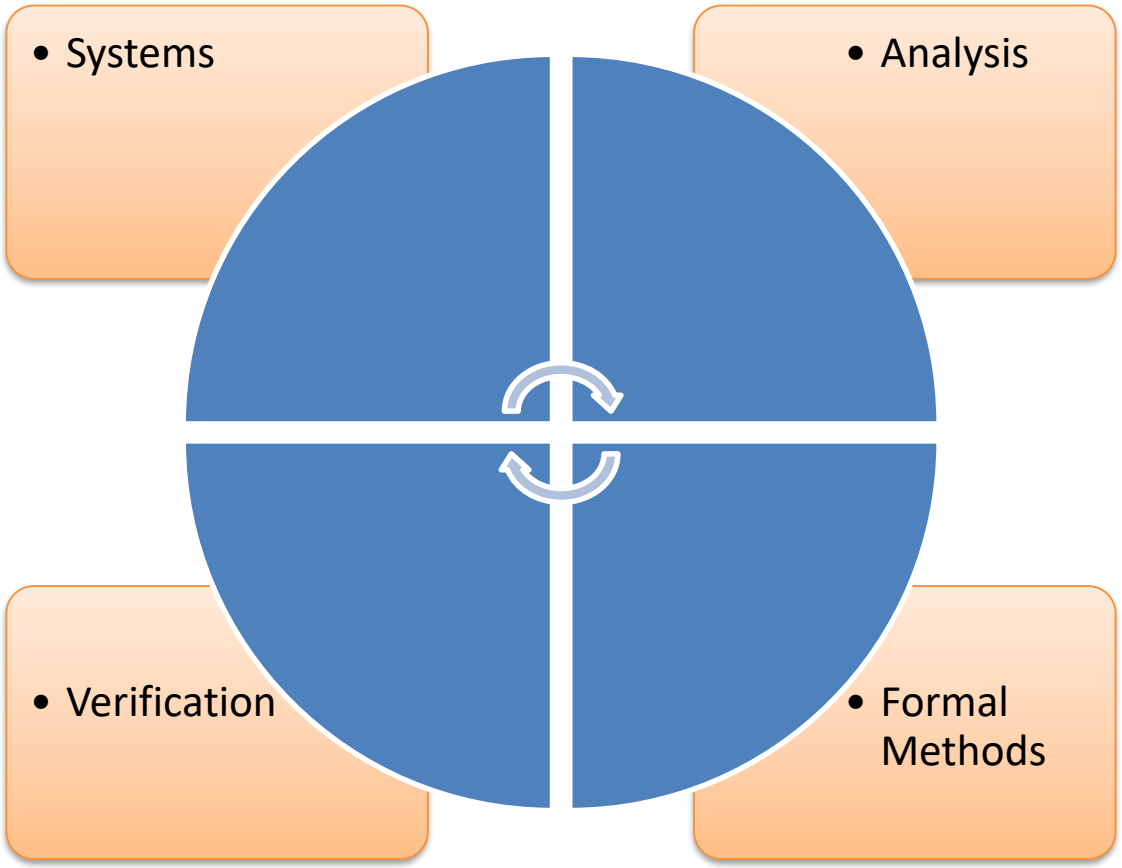
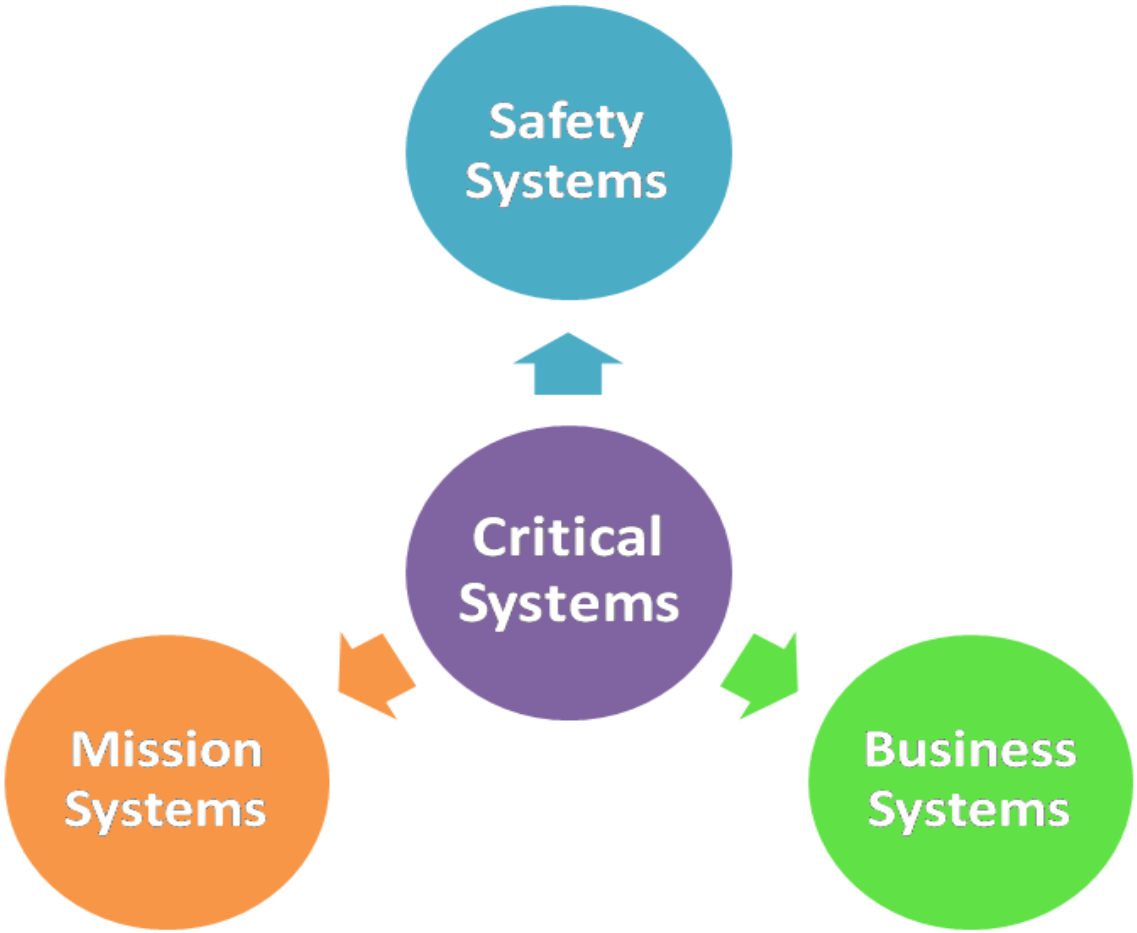
Formal Methods in Blockchain

Dr. Raju Halder

Formal Methods for Analysis and Verification



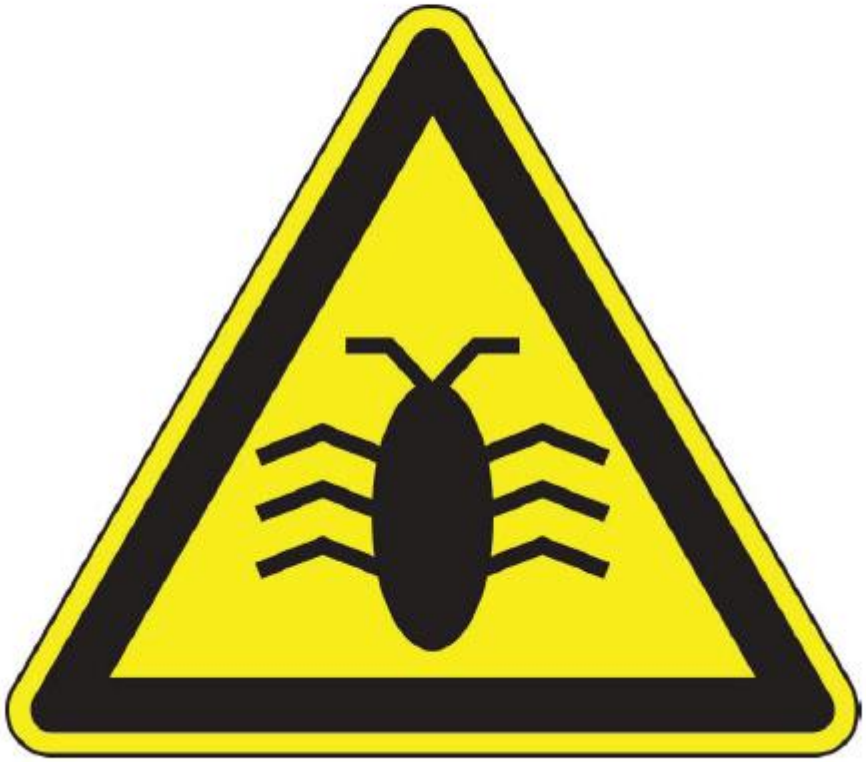
Critical Systems and Formal Methods



Critical Systems

- Safety-critical systems
  - Failure results in loss of life, injury or damage to the environment;
  - Chemical plant protection system;
- Mission-critical systems
  - Failure results in failure of some goal-directed activity;
  - Spacecraft navigation system;
- Business-critical systems
  - Failure results in high economic losses;
  - Customer accounting system in a bank;

Bugs are Frequent in Software



Bugs are Frequent in Software

Bugs are Frequent in Software

Bugs are Frequent in Software

Bugs are Frequent in Software

Software bugs may cause big troubles…

On 4 June 1996, the maiden flight of the Ariane 5 launcher ended in a failure. Only about 40 seconds after initiation of the flight sequence, at an altitude of about 3700 m, the launcher veered off its flight path, broke up and exploded. »

The failure of the Ariane 5.01 was caused by the complete loss of guidance and attitude information 37 seconds after start of the main engine ignition sequence, 30 seconds after lift-off.

This loss of information was due to specification and design errors in the software of the inertial reference system.

https://www.youtube.com/watch?v=PK_yguLapgA

Software bugs may cause big troubles...

A conversion from 64-bit floating point to 16 bit integer with a value larger than possible. The overflow caused a hardware trap

FAILURES IN COMPLEX SYSTEMS: Some Real Examples

- **Six accidents by the Therac-25**
  - Deaths of 6 Patients for massive overdoses of radiation
  - Some instances operators repeated overdoses because machine display indicated no dose given

- **The failure of the Patriot missile during the Gulf war**
  - 24 deaths due to calculation in error
  - The systems internal clock was measured in tenths of seconds and the actual time was reported by multiplying the internal clock's value with a 24-bit fixed-point register.

- **The loss of Mars Climate Orbiter**
  - due to a unit error (Navigation team was calculating metric units while the ground calculations were in imperial unit)

- **Many More …..**

Remarks by Bill Gates

17th Annual ACM Conference on Object-Oriented Programming, Seattle, Washington, November 8, 2002

"… When you look at a big commercial software company like Microsoft, there's actually as much testing that goes in as development.

We have as many testers as we have developers.

Testers basically test all the time, and developers basically are involved in the testing process about half the time…

Remarks by Bill Gates

17th Annual ACM Conference on Object-Oriented Programming, Seattle, Washington, November 8, 2002

"… We've probably changed the industry we're in. We're not in the software industry; we're in the testing industry, and writing the software is the thing that keeps us busy doing all that testing."

"…The test cases are unbelievably expensive; in fact, there's more lines of code in the test harness than there is in the program itself. Often that's a ratio of about three to one."

System Dependability

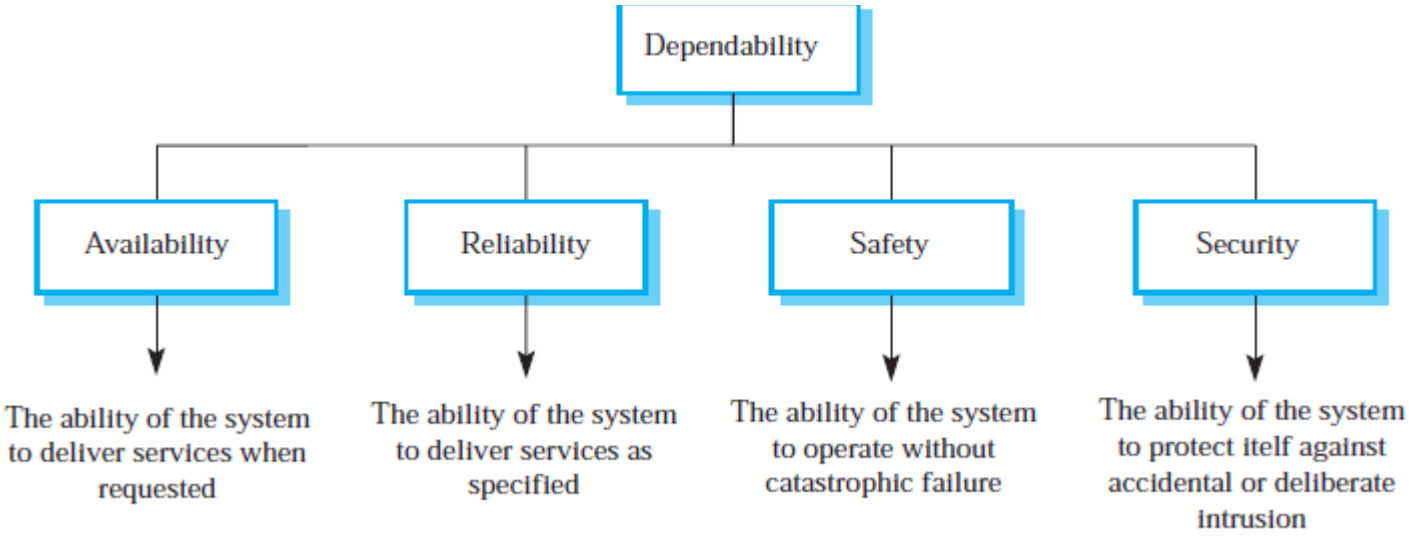
- For many software-intensive system (especially, critical systems), the most important system-property is the dependability of the system.
- User trust a system that is dependable.
- Dependability of a system reflects the user's degree of trust in the system.
- It reflects the extent of the user confidence that it will operate as user expects and that it will not 'fail' in normal use

System Dependability

- Dependability is a non-functional requirement.
- Usefulness and trustworthiness are not the same thing. A system does not have to be trusted to be useful.

Principal properties of dependability:

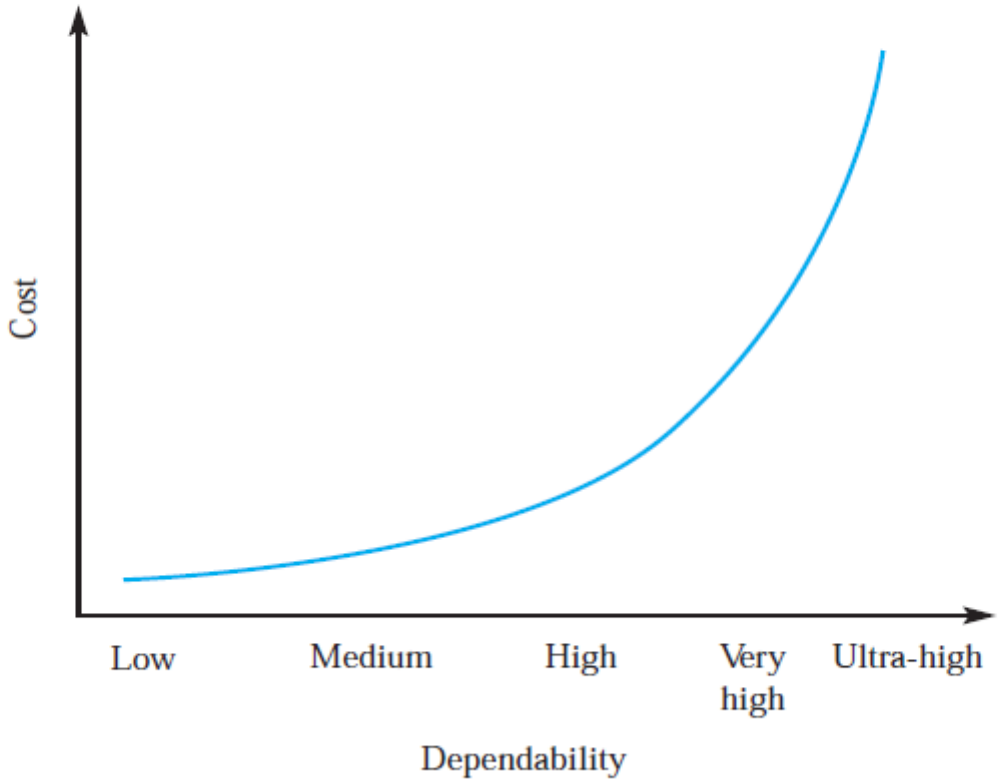
- Principal dimensions of dependability are:
  - Availability;
  - Reliability;
  - Safety;
  - Security
  - Others ...



Other dependability properties

- Repairability
  - Reflects the extent to which the system can be repaired in the event of a failure
- Maintainability
  - Reflects the extent to which the system can be adapted to new requirements;
- Survivability
  - Reflects the extent to which the system can deliver services whilst under hostile attack;
- Error tolerance
  - Reflects the extent to which user input errors can be avoided and tolerated.

Costs of increasing dependability



Some of the more typical functional requirements include:

- Business Rules
- Transaction corrections, adjustments and cancellations
- Administrative functions
- Authentication
- Authorization levels
- Audit Tracking
- External Interfaces
- Certification Requirements
- Reporting Requirements
- Historical Data
- Legal or Regulatory Requirements

Some typical non-functional requirements are:

- Performance – for example Response Time, Throughput, Utilization, Static Volumetric
- Scalability
- Capacity
- Availability
- Reliability
- Recoverability
- Maintainability
- Serviceability
- Security
- Regulatory
- Manageability
- Environmental
- Data Integrity
- Usability
- Interoperability

User's need	User's concern	Non-functional requirement
Function	1. Ease of use 2. Unauthorised access 3. Likelihood of failure	1. Usability 2. Security 3. Reliability
Performance	1. Resource utilisation 2. Performance verification 3. Ease of interfacing	1. Efficiency 2. Verifiability 3. Interoperability
Change	1. Ease of repair 2. Ease of change 3. Ease of transport ? 4. Ease of expanding or upgrading capacity or performance ?	1. Maintainability 2. Flexibility 3. Portability 4. Expandability

Formal Methods

“ Formal methods are mathematical approaches to software and system development which support the rigorous specification, design and verification of computer systems.” [Fme04]

“[they]… exploit the power of mathematical notation and mathematical proofs. “ [Gla04]

What you will learn?

- Techniques for representing programs
- Defining Syntax and Semantics of Programs
- Techniques for Analyzing and Verifying Programs
- Applications of these techniques
- You may learn how to formally (mathematically) shape your thought in report/thesis/paper !
- You may feel more comfortable with ToC and Compiler!