Universität Bremen

dfki Deutsches Forschungszentrum für Künstliche Intelligenz
German Research Center for Artificial Intelligence

Systems of High Safety and Security

Lecture 1 from 14.10.26: Einführung

Winter term 2026/27



Serge Autexier, Christoph Lüth

Organisatorisches

Generelles

- Einführungsvorlesung zum Masterprofil S & Q
  Schwerpunkt: IMA-SQ, IMVP-VMC
- 6 ETCS-Punkte
- Vorlesung und Übung:
  - Mi 10 – 12 Uhr (MZH 1450)
  - Mi 12 – 14 Uhr (MZH 1450)
- Material (Folien, Artikel, Übungsblätter) sind auf der Webseite:

  `https://user.informatik.uni-bremen.de/~clueth/lehre/ssq.ws26/`
- Veranstalter:
- Christoph Lüth `clueth@uni-bremen.de`
- Serge Autexier `autexier@uni-bremen.de`

Vorlesung

- Foliensätze als **Kernmaterial**
  - Sind auf Englisch (Notationen!)
- Zusatzmaterial:
  - Vorlesungsnotizen
  - Ausgewählte Fachartikel
- Bücher nur für einzelne Teile der Vorlesung verfügbar:
  - Nancy Leveson: Engineering a Safer World
  - Glynn Winskel: The Formal Semantics of Programming Languages
  - Michael Huth, Mark Ryan: Logic in Computer Science
  - ... weitere im Laufe der Vorlesung

Übungen

- Übungsblätter:
  - "Leichtgewichtige" Übungsblätter, die in der Übung bearbeitet und schnell korrigiert werden können.
  - Übungsblätter vertiefen Vorlesungsstoff.
  - Bewertung gibt schnell Feedback.
- Übungsbetrieb:
  - Übungsgruppen: bis zu **fünf** (idealerweise drei) Studierende
  - Bearbeitung: während der Übung
  - Abgabe: bis zur Vorlesung

Ablauf des Übungsbetriebs

- Abgabe und Korrektur des Übungsbetriebs erfolgt über **gitlab** .
  - Dazu legt pro Gruppe ein Repository an.
  - Ladet uns (clueth, autexier) als Developer ein.
- Für jedes Übungsblatt:
  - Ihr ladet das Übungsblatt herunter (uebung-XX.md) und bearbeitet es elektronisch.
  - Die Lösung wird als Markdown in euer Repo abgelegt (dabei Namen uebung-XX.md nicht verändern; Zusatzmaterial als uebung-XX-...wenn nötig), und **vor dem Abgabezeitpunkt** hochgeladen (push).
  - Nach dem Abgabezeitpunkt laden wir die Abgaben herunter (pull), korrigieren direkt im Markdown, fügen die Bewertung hinzu, und laden die Korrektur wieder hoch (push).
  - Die Datei 00-BEWERTUNG.md enthält die fortlaufenden Bewertungen für die Gruppe.

Prüfungsleistung

- Bewertung der Übungen:
  - A (sehr gut (1.0) – nichts zu meckern, nur wenige Fehler)
  - B (gut (2.0) – kleine Fehler, im großen und ganzen gut)
  - C (befriedigend (3.0) – größere Fehler oder Mängel)
  - Nicht bearbeitet (oder zu viele Fehler)
- Prüfungsleistung:
  - Teilnahme am Übungsbetrieb
  - Eine **Lösung vorstellen** (pro Gruppe)
  - Mündliche Prüfung am Ende des Semesters
    - Einzelprüfung, ca. 20- 30 Minuten
    - Übungsbetrieb (bis zu 10% Bonuspunkte, keine Voraussetzung)

Ziel der Vorlesung

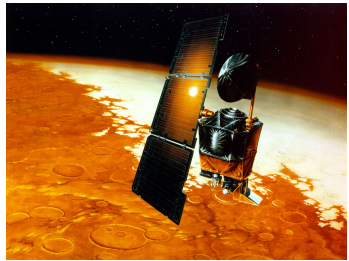
- Methoden und Techniken zur Entwicklung sicherheitskritischer Systeme
- Schwerpunkt: formale Methoden
- Überblick über verschiedene Mechanismen
  - Vertiefung nach Wahl in verschiedenen Veranstaltungen
- Verschiedene Dimensionen
  - Hardware vs. Software
  - Security vs. Safety

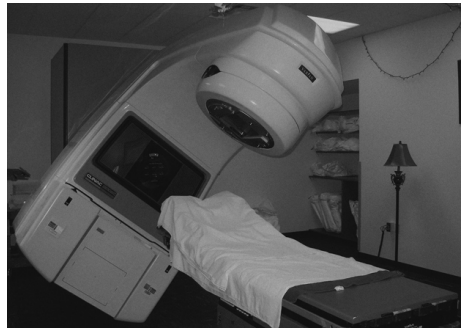
> ***Formal Methods*** *refers to mathematically rigorous techniques and tools for the specification, design and verification of software and hardware systems.*
>
> — NASA, `https://shemesh.larc.nasa.gov/fm/fm-what.html`

Overview

Famous Software Disasters



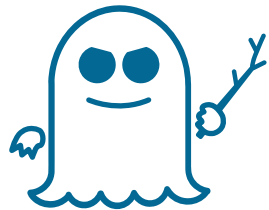




Famous Hardware Disasters







Intel Pentium FDIV Bug (1995), Meltdown and Spectre vulnerabilites (2018)

Famous Software Disasters: Ariane 5

- Ariane 5 exploded on its virgin flight (Ariane Flight 501) on 4.6.1996.
- How could that happen?



What Went Wrong With Ariane Flight 501?

1. Self-destruction due to instability;

What Went Wrong With Ariane Flight 501?

1. Self-destruction due to instability;
2. Instability due to wrong steering movements (rudder);

What Went Wrong With Ariane Flight 501?

1. Self-destruction due to instability;
2. Instability due to wrong steering movements (rudder);
3. On-board computer tried to compensate for (assumed) wrong trajectory;

What Went Wrong With Ariane Flight 501?

1. Self-destruction due to instability;
2. Instability due to wrong steering movements (rudder);
3. On-board computer tried to compensate for (assumed) wrong trajectory;
4. Trajectory was calculated wrongly because own position was wrong;

What Went Wrong With Ariane Flight 501?

1. Self-destruction due to instability;
2. Instability due to wrong steering movements (rudder);
3. On-board computer tried to compensate for (assumed) wrong trajectory;
4. Trajectory was calculated wrongly because own position was wrong;
5. Own position was wrong because positioning system had crashed;

What Went Wrong With Ariane Flight 501?

1. Self-destruction due to instability;
2. Instability due to wrong steering movements (rudder);
3. On-board computer tried to compensate for (assumed) wrong trajectory;
4. Trajectory was calculated wrongly because own position was wrong;
5. Own position was wrong because positioning system had crashed;
6. Positioning system had crashed because transmission of sensor data to ground control failed with integer overflow;

What Went Wrong With Ariane Flight 501?

1. Self-destruction due to instability;
2. Instability due to wrong steering movements (rudder);
3. On-board computer tried to compensate for (assumed) wrong trajectory;
4. Trajectory was calculated wrongly because own position was wrong;
5. Own position was wrong because positioning system had crashed;
6. Positioning system had crashed because transmission of sensor data to ground control failed with integer overflow;
7. Integer overflow occurred because values were too high;

What Went Wrong With Ariane Flight 501?

1. Self-destruction due to instability;
2. Instability due to wrong steering movements (rudder);
3. On-board computer tried to compensate for (assumed) wrong trajectory;
4. Trajectory was calculated wrongly because own position was wrong;
5. Own position was wrong because positioning system had crashed;
6. Positioning system had crashed because transmission of sensor data to ground control failed with integer overflow;
7. Integer overflow occurred because values were too high;
8. Values were too high because positioning system was integrated unchanged from predecessor model, Ariane-4;

What Went Wrong With Ariane Flight 501?

1. Self-destruction due to instability;
2. Instability due to wrong steering movements (rudder);
3. On-board computer tried to compensate for (assumed) wrong trajectory;
4. Trajectory was calculated wrongly because own position was wrong;
5. Own position was wrong because positioning system had crashed;
6. Positioning system had crashed because transmission of sensor data to ground control failed with integer overflow;
7. Integer overflow occurred because values were too high;
8. Values were too high because positioning system was integrated unchanged from predecessor model, Ariane-4;
9. This assumption was not documented because it was satisfied tacitly with Ariane-4.

What Went Wrong With Ariane Flight 501?

1. Self-destruction due to instability;
2. Instability due to wrong steering movements (rudder);
3. On-board computer tried to compensate for (assumed) wrong trajectory;
4. Trajectory was calculated wrongly because own position was wrong;
5. Own position was wrong because positioning system had crashed;
6. Positioning system had crashed because transmission of sensor data to ground control failed with integer overflow;
7. Integer overflow occurred because values were too high;
8. Values were too high because positioning system was integrated unchanged from predecessor model, Ariane-4;
9. This assumption was not documented because it was satisfied tacitly with Ariane-4.
10. Positioning system was redundant, but both systems failed (systematic error).

What Went Wrong With Ariane Flight 501?

1. Self-destruction due to instability;
2. Instability due to wrong steering movements (rudder);
3. On-board computer tried to compensate for (assumed) wrong trajectory;
4. Trajectory was calculated wrongly because own position was wrong;
5. Own position was wrong because positioning system had crashed;
6. Positioning system had crashed because transmission of sensor data to ground control failed with integer overflow;
7. Integer overflow occurred because values were too high;
8. Values were too high because positioning system was integrated unchanged from predecessor model, Ariane-4;
9. This assumption was not documented because it was satisfied tacitly with Ariane-4.
10. Positioning system was redundant, but both systems failed (systematic error).
11. Transmission of data to ground control was even unnecessary!

What is Safety and Security?

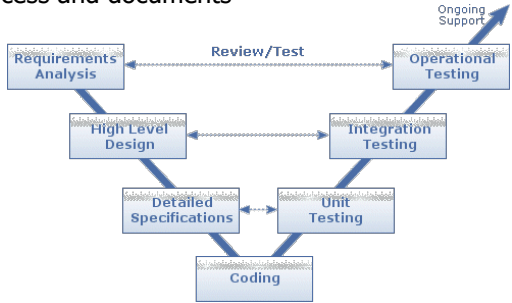
- **Safety** is ensured if product achieves acceptable levels of risk or harm to people, business, software, property or the environment in a specified context of use.
- Threats from “inside”
  - Avoid malfunction of a system – this concerns both hardware and software
  - E.g. planes, cars, railways
- Threats from “outside”
  - Protect product against force majeure (“acts of god”, “höhere Gewalt”)
  - E.g. Lightening, storm, floods, earthquake, fatigue of material, loss of power

What is Safety and Security?

- **Security** is ensured if product is protected against potential attacks from people, environment etc.
- Protection against threats from "outside"
  - Analyze and counteract the abilities of an attacker (also called malicious agent).
- Protection against threats from "inside"
  - Monitor activities of own personnel, to prevent
    - selling of sensitive company data
    - insertion of Trojans during HW/SW design
    - involuntary misuse
- In this context: "cybersecurity" (not physical security)

Software Development Models

- Definition of software development process and documents
- Examples:
  - Waterfall Model
  - **V-Model**
  - Model-Driven Architectures
  - Agile Development



- Motivation:
  - A well-defined development process is more likely to result in a high-quality product than a chaotic process
  - "Process quality ensures product quality"

Verification and Validation (V&V)

- **Verification**: have we built the system right?
  - i.e. correct with respect to a reference artefact
    - specification document
    - reference system
    - model

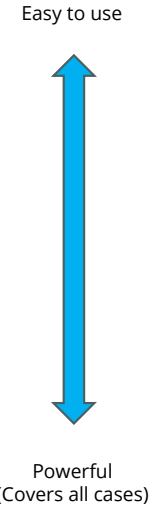
Deutsch: Korrektheit

- **Validation**: have we built the right system?
  - i.e. effective (or adequate) for its intended operation?

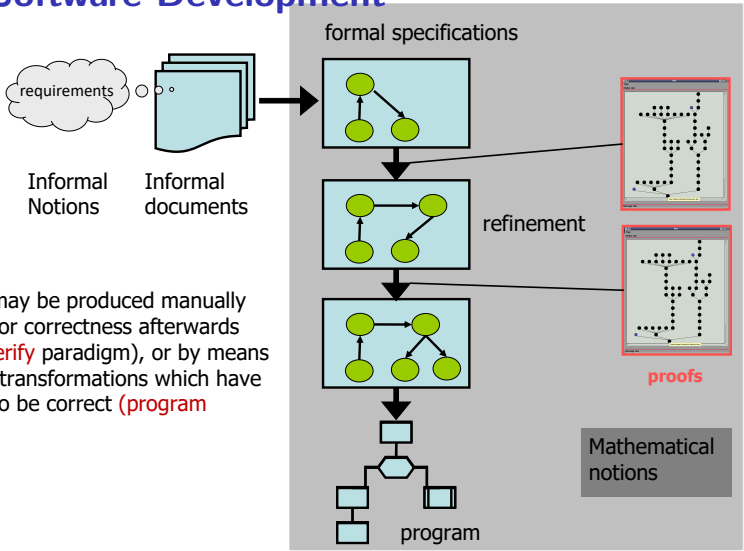
Deutsch: Wirksamkeit

V&V Methods

- **Testing**
  - Test case generation, black- vs. white box
  - Hardware-in-the-loop (HiL) testing: integrated HW/SW system is tested
  - Software-in-the-loop (SiL) testing: only software is tested
  - Program runs using symbolic values (symbolic execution, concolic test)
- **Simulation**
  - An executable model is tested with respect to specific properties
  - This is also called Model-in-the-Loop (MiL) testing
- Static/dynamic **program analysis**
  - Dependency graphs, flow analysis
  - Symbolic evaluation, abstract interpretation
- **Model checking**
  - Automatic proof by reduction to finite state problem
- **Formal Verification**
  - Symbolic proof of program properties



Formal Software Development



Refinements may be produced manually and checked for correctness afterwards (invent-and-verify paradigm), or by means of automated transformations which have been proven to be correct (program synthesis)

Concepts of Quality

What is Quality?

- Quality is the collection of its characteristic properties.
- Quality model: decomposes the high-level definition by associating attributes (also called characteristics, factors, or **criteria**) to the quality conception.
  - See Wikipedia for a long list of quality attributes.
- Quality **indicators** associate **metric values** with **quality criteria** , expressing "how well" the criteria have been fulfilled by the process or product.
  - The idea is to **measure** quality, with the aim of continuously **improving** it.
  - Leads to **quality management**
    - TQM = total quality management
    - Kaizen = continuous incremental quality improvement
  - ... but note Goodhart's law:
    "*When a measure becomes a target, it ceases to be a good measure.*"



Quality Criteria: Different Dimensions of Quality

- For the development of artifacts quality criteria can be measured with respect to the
  - development process (**process quality**), or
  - final product (**product quality**).
- Another dimension for structuring quality conceptions is
  - **Correctness** (*Korrektheit*): the consistency with the product and its associated requirements specifications, and
  - **Effectiveness** (*Wirksamkeit*): the suitability of the product for its intended purpose.
- A third dimension structures quality according to product properties:
  - **Functional properties** : the specified services to be delivered to the users
  - **Structural properties** : architecture, interfaces, deployment, control structures
  - **Non-functional properties** : usability, reliability, availability, security, maintainability, guaranteed worst-case execution time (WCET), costs, absence of run-time errors, ...

Other Norms and Standards

- ISO 9001 (DIN ISO 9000-4):
  - Standardizes definition and supporting principles necessary for a quality system to ensure products meet requirements
  - 'Meta-Standard'
- CMM (Capability Maturity Model), Spice (ISO 15504)
  - Standardizes maturity of development process
  - Level 1 (initial): Ad-hoc
  - Level 2 (repeatable): process dependent on individuals
  - Level 3 (defined): process defined & institutionalized
  - Level 4 (managed): measured process
  - Level 5 (optimizing): improvement feed back into process

Summary

- **Safety** vs. **Security**
- **Quality**
  - collection of characteristic properties
  - quality indicators measuring quality criteria
- Relevant **aspects of quality** here
  - Functional **suitability** and **safety** (functional correctness)
  - **Dependability** (availability, reliability, security — non-functional correctness)
- Next week:
  - Concepts of safety, legal requirements, certification