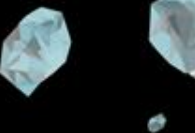
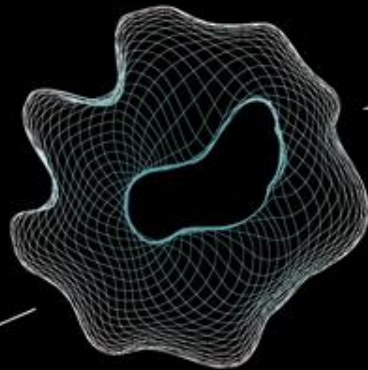


UNIVERSITY OF TWENTE.


**SOFTWARE RESEARCH:
IMPACT AND CHALLENGES**

MARIEKE HUISMAN

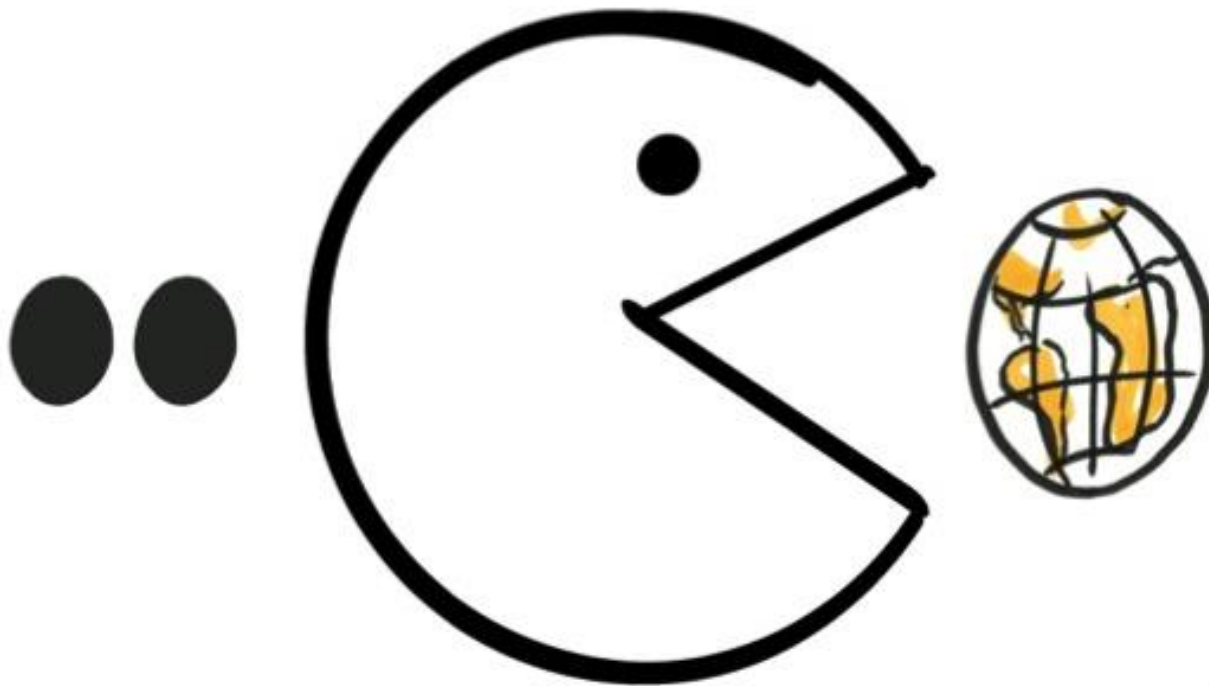
UNIVERSITY OF TWENTE, NETHERLANDS



WHY AM I HERE?

- Professor Software Reliability
Formal Methods and Tools group, University of Twente
- Research on verification of concurrent software
VerCors program verifier 
- Chair of **VERSEN** (2018 – 2021)
Dutch Association of Software Researchers
European Software Manifesto

SOFTWARE IS EATING THE WORLD (2011)



* Marc Andreessen
in Wall Street Journal

SOFTWARE IS EVERYWHERE



Software failures
can have
enormous impact



**Gedateerd computersysteem
veroorzaakte chaos op Britse
luchthavens**

All software
has errors!

Software is
not static

SOFTWARE INDUSTRY

- Software is a growth sector: in 2018, it grew in volume by 4.5 %
- 5,5 % of the total workforce in the Netherlands work in ICT
- Manufacturers, banks and high-tech companies see enormous growths in their software development: software-intensive industry

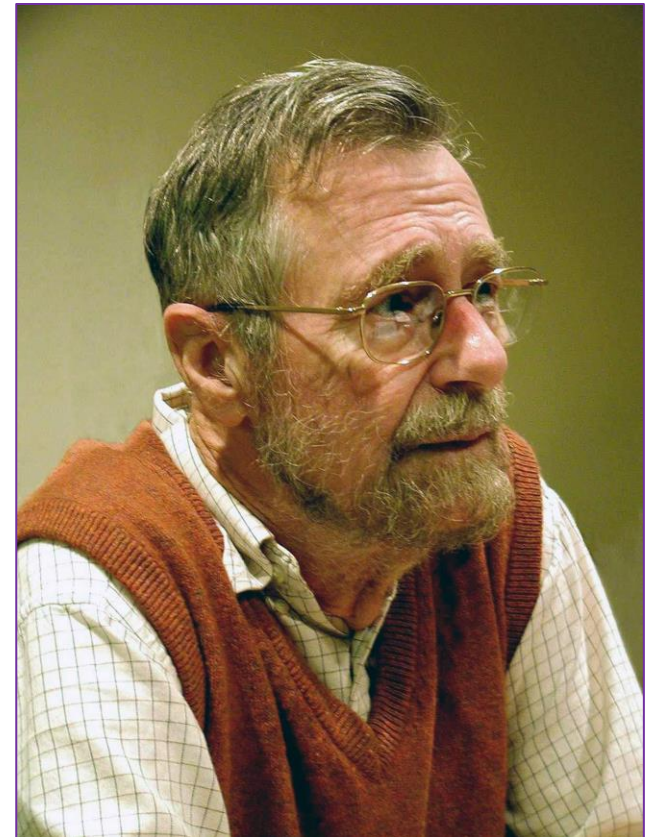




SOFTWARE RESEARCH HISTORY

Software crisis: the difficulty of writing useful and efficient computer programs in the required time [1968, NATO software engineering conference]

Software construction is an intellectual challenge without precedent in the cultural history of mankind [1972, Turing acceptance speech Edsger Dijkstra]



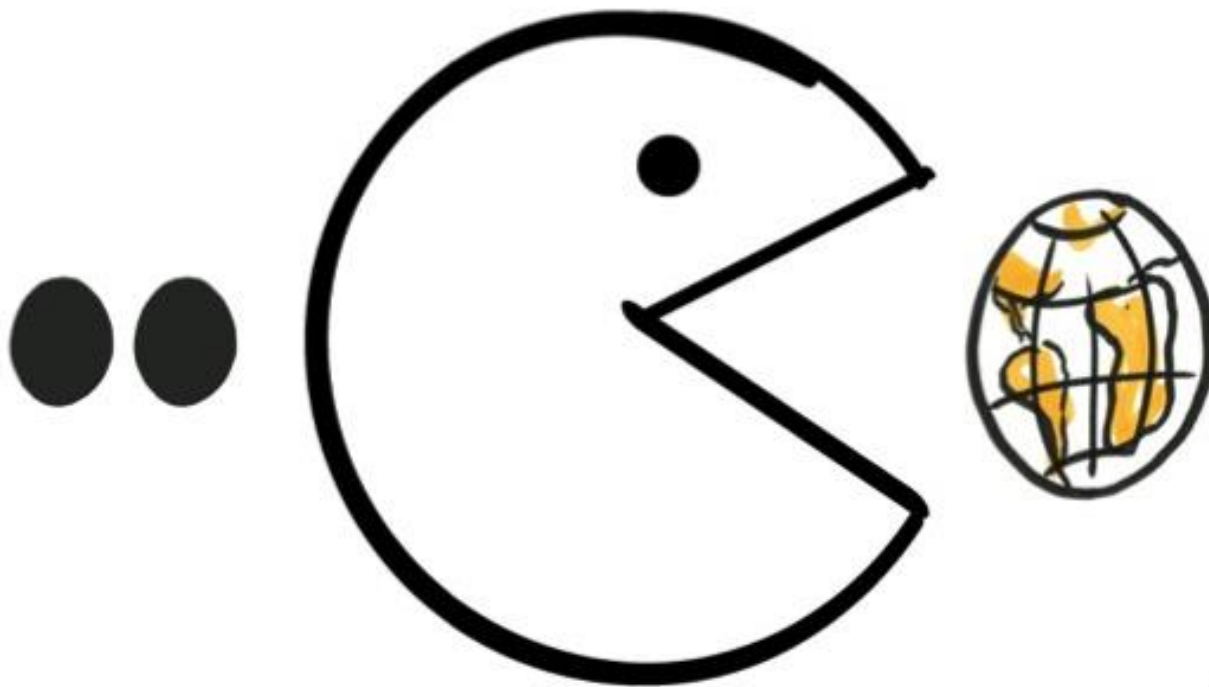
Edsger Dijkstra

ACHIEVEMENTS SINCE THEN

- Powerful methods, techniques, theories, and tools for building software systems that are correct, secure and maintainable
- A deep understanding of how people build and evolve software systems
- Large body of literature
- Research results used in practice
- But software develops quickly



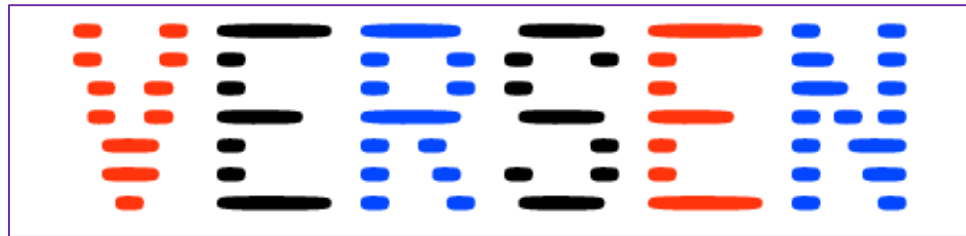
SOFTWARE IS A MOVING TARGET



* Marc Andreessen
in Wall Street Journal

SOFTWARE CHALLENGES

- Software Reliability
- Efficient Engineering of Software
- Software Maintainability and Evolution
- (Software Education)



Dutch Association of Software Researchers

SOFTWARE RELIABILITY

How to ensure that the software behaves as expected?

- Does software execute its intended task?
 - Including security, performance, energy-efficiency, usability, etc.
- All software contains errors
 - The earlier they are detected (or prevented), the better
- Various **verification technologies** exist
 - testing and monitoring systems
 - model checking
 - theorem proving

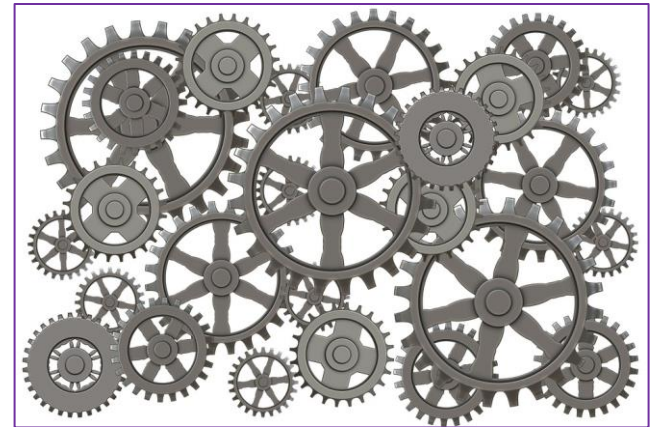
CHALLENGES IN SOFTWARE RELIABILITY

- Reliability by construction
 - Can we construct software automatically?
- Scaling of verification techniques
 - Guaranteeing properties on highly complex software
- Reliable software on modern hardware
 - Effective and reliable use of parallelism
- Robustness against unexpected uses
 - The human factor
- Tools to create and measure software
 - Software should fulfil social, economic, and environmental sustainability needs

EFFICIENT ENGINEERING OF SOFTWARE

How to create all the necessary software in the limited time we have available?

- Demands for software development and change are higher than we can humanly deliver
 - Velocity
 - Volume
 - Variety
- We need to work smarter, not harder



CHALLENGES IN EFFICIENT ENGINEERING OF SOFTWARE

- Shorter development feedback loops
 - Analyze requirements, software artifacts and contextual data
- Tool-supported software development
 - Dedicated new software languages
 - Dedicated development environments
- Empirical software engineering
 - Learn from successful and robust existing software systems
- Automated software engineering
 - Leveraging the advances in AI

SOFTWARE MAINTAINABILITY AND EVOLUTION

How to create long-living software that can be cost-efficiently evolved and migrated to new technologies?

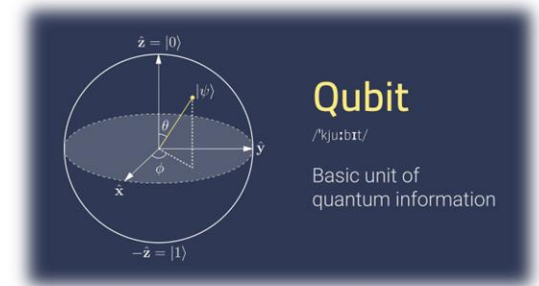
- Software has to be maintained and adapted over time
- Changes can make software too complex
 - Maintenance and evolution become too expensive
 - Or even unfeasible



CHALLENGES IN SOFTWARE MAINTAINABILITY AND EVOLUTION

- Organisations lose control over software
 - Which part of the software is worth maintaining
 - Which parts need to be re-developed from scratch
 - We are unable to predict
 - when software quality is degrading
 - the costs to repair it
- Modern software cannot cope with continuous and unpredictable change
 - Sustainability of the software is often an afterthought
- Software sustainability is not a purely technical challenge
 - Socio-technical aspects, i.e. development teams

THE FUTURE OF SOFTWARE



What happens with the quantum computer?

- How to program the quantum computer?
- Quantum programs are also software
- With all the usual software challenges...

TO CONCLUDE...

- Software is the invisible driving force of our modern society
- Effort to make reliable, high-quality software **routinely underestimated**
- Many open research questions that need to be addressed
- Software research is an important and active research area



Dutch Association of Software Researchers