

VEReniging Software Engineering Nederland

#### 

### VEReniging Software Engineering Nederland

#### About VERSEN

VERSEN is the Dutch National Association for Software Engineering. The mission of the VERSEN association is to bring together researchers, educators and practitioners in the field of software engineering in The Netherlands, who share the common goals of advancing the field of software engineering, raising public awareness of the challenges and opportunities of the field, acquiring funding for groundbreaking research, and transferring academic results to broader society.

www.versen.nl

#### Manifesto on Software Research and Education in the Netherlands

\*\*\* By VERSEN

Software plays a role in almost every aspect of our daily lives. As software becomes increasingly complex, ensuring correctness, safety, security and flexibility is a major challenge. Software research and education in the Netherlands are under siege due to the steeply increasing teaching load and decreasing funding. A serious investment in software research and education is therefore urgently needed.

The Netherlands wants to take the lead in digitization and wants to influence technological developments. To achieve this goal, excellent research and education in software creation are a prerequisite. The industry of software-intensive systems is crucial for our economy; 5.5% of the total workforce in the Netherlands works in ICT. Well-educated software engineers enable the success of innovative companies, but they are at the moment starving for talent and qualified software professionals. Research in computer science and software engineering has produced many powerful methods, techniques, theories, and tools for building correct, secure and maintainable software systems. However, with the lightning-fast technological change and churn, many new challenges for

software have come into existence.

# Challenges

#### Software Reliability

How to ensure that software behaves as expected? This does not only concern the basic intended functionality, but also quality aspects such as security, performance, energy-efficiency, scaleability, sustainability and usability. All software contains errors, and the earlier they are detected (or prevented), the better. Urgent research questions in software reliability are:

- How to create software that is correct and reliable by construction?
- How to create software that respects required quality aspects (see above) by design?
- How to scale verification techniques to check software for bugs or inconsistencies?
- Humans are the weakest link in using softwareintensive systems. How to design software that actively detects, prevents and mitigates the consequences of human error?

## Software Maintainability and Evolution

After software has been created, it has to be maintained and adapted over time. If this is not done effectively, the software becomes more and more complex, and the costs of maintenance and evolution become unsurmountable. Urgent challenges are:

- Organisations lose control over software and they can't gauge which part of their software is worth maintaining and which parts need to be re-developed. How to analyse and prioritise such maintenance tasks?
- Modern software cannot cope with continuous and unpredictable change. How to facilitate software that is changeable by design?
- Socio-technical aspects help, or hinder, achieving maintainability goals. How do we organize software development to embrace and facilitate change?

Sponson CWI



Sponsored by:





### Efficient Engineering of Software

Software engineering is exceeding human capabilities in termcvvenges in efficient engineering are:

- How to make software development more effective and efficient, in order to continuously deliver?
- How to create efficient tools that help us to develop software that is reduced in size and complexity?
- What can we learn from existing software systems?
- Which software engineering practices are effective and productive?
- How to use even more automation in software engineering to bring it to the next level?

#### Software Education

The software industry in the Netherlands employs hundreds of thousands of developers, which all need further professional development and life-long learning.

Scalable practices are needed for:

- Teaching technical skills for software development such as design, programming, programming languages, testing, frameworks, and tooling.
- Teaching soft skills such as collaboration, communication, and giving feedback.
- Attracting and retaining a diverse student population.

To address these challenges, we need new educational practices, new tooling that provides high-quality feedback on software system quality, and new ways to assess software development skills. These require not only pedagogical soundness, but also the ability to take into account the competencies of the user to give feedback and help, and to assess skills.

#### Our Goal

By addressing all these challenges we intend to:

- Increase the number of software engineers graduating at higher education institutes.
- Increase the quality of graduates and employees.
- Provide easier access to development opportunities.
- Increase the diversity of the student population and workforce.