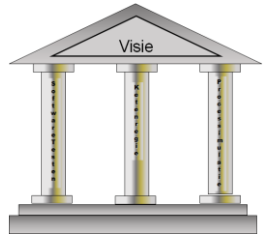
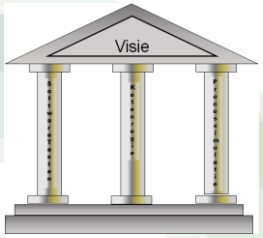


The future of software testing

General approach

Jos van Rooijen



Agenda

- Introduction
- Challenges per perspective
- How to proceed? The first ideas
- Summary

Introduction

Consultant at Huis voor Software kwaliteit

35 years in software testing & quality management

Co-author several quality related books

Test expert online magazine Computable

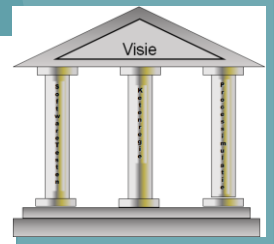
Publication areas; Testing, Education, test automation and quality supervision

Graduation supervisor Avans university of applied science

Member of the steering committee Valid conference and IARIA fellow

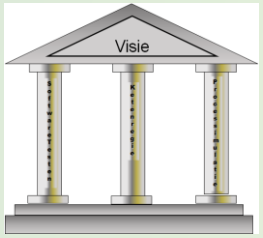
Member of the board ‘historical society’ city of Lingewaard

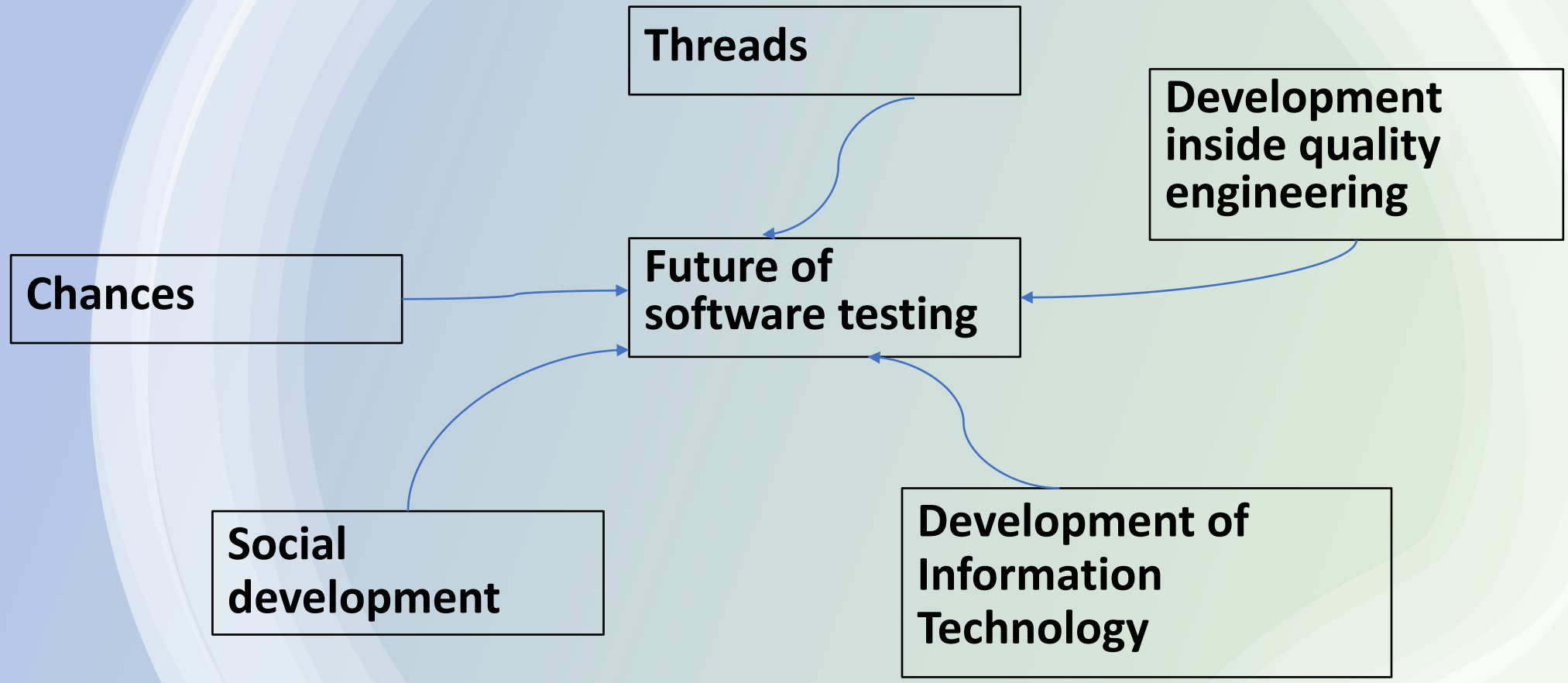
Introduction



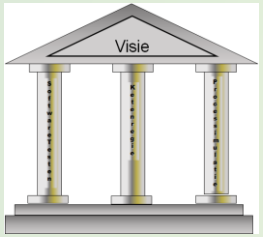
- Software testing is still really traditional
- Innovation and research is below par
- The (so called) innovation concentrates:
  - How to apply test in a new development method
  - Test tooling
- Hardly none testing techniques to beat future challenges we have to face, like:
  - Self driving cars
  - Code development by hand of AI (how to prove the code?)
  - Dependencies of medical devices
  - Smart devices connected to everything and everywhere
  - Etc.

The context


Visie


Threads
Development inside quality engineering
Chances
Future of software testing
Social development
Development of Information Technology

Some challenges per perspective



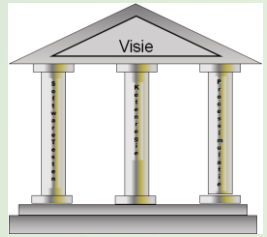
Developments of Information Technology:

- Increasing complexity systems connected with everything and everywhere
- Variability information systems is increasing
- Complexity of the configuration of business processes increases
- Unknown where the system start or stops

Developments Quality Engineering:

- Increasing dynamics. Development of information systems is never finished. So testing is also never finished!
- Increasing complexity
- Old problems are never solved
- Bugs appears on different levels. Configuration, integration or parametrization

Some challenges per perspective



Threads:

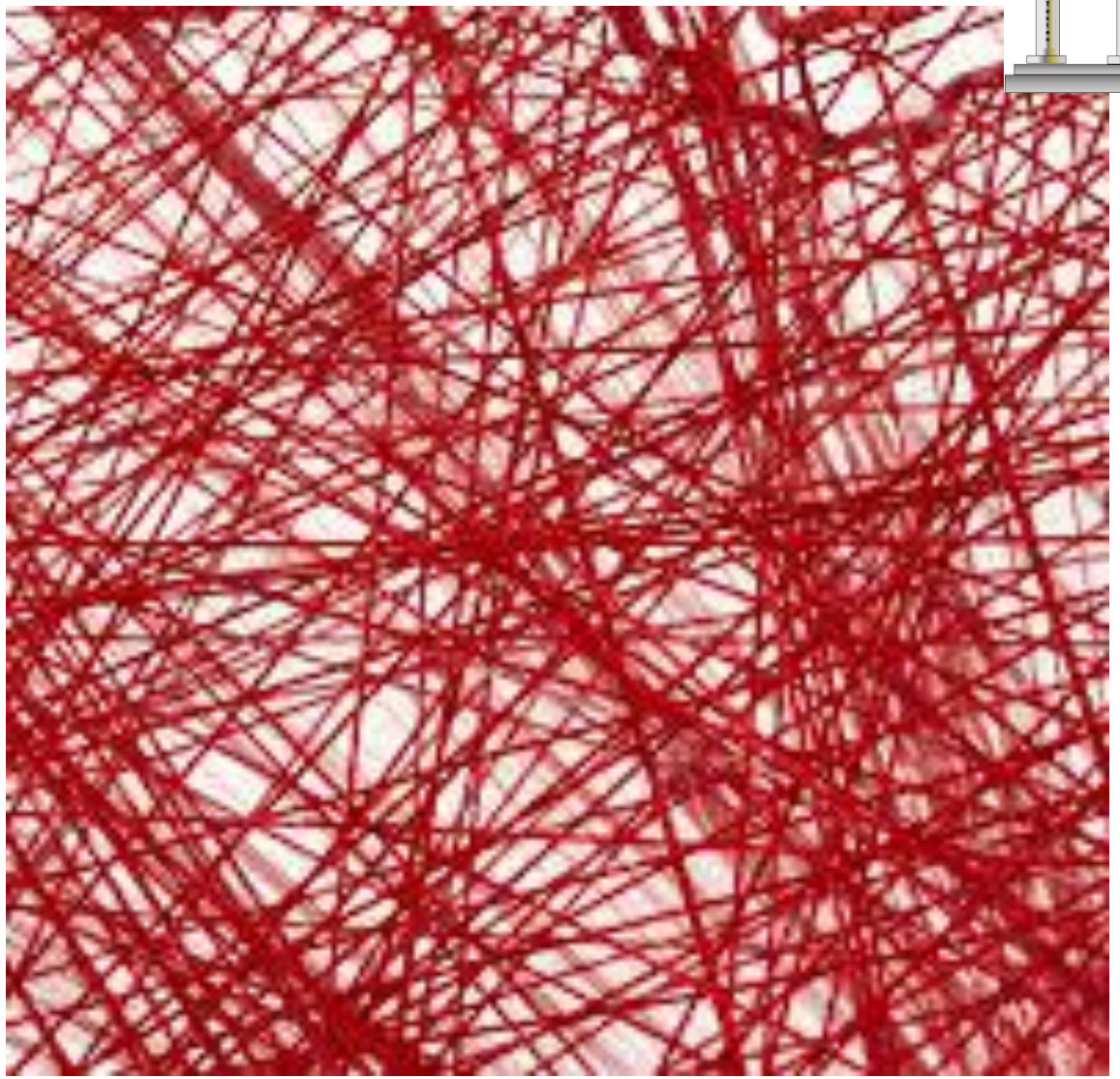
- Low chance, high impact
- Aging
- Complexity
- Self learning information systems; we don’t know any more how the information system works
- Lack of cooperation between the industry and academia
- Availability of labor will become a problem

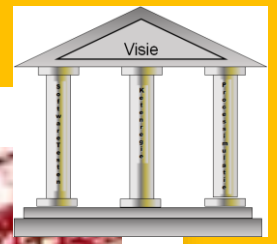
Chances:

- Can we test the situation of tomorrow instead of today? Can we look ahead?

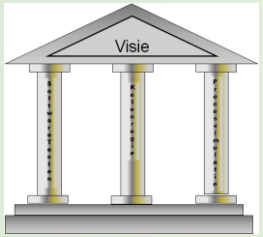
The common thread

- Environment plays an important role
- Chance versus impact
- From reactive to proactive
- Dependency and complexity is increasing
- We don’t know the information system any more
- We are not able to solve old problems



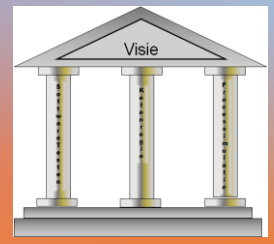


What are the measures we have to take?

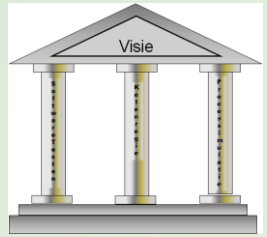


- What kind of testing techniques / approaches we have to develop?
- Is there something available?
- Traditional test approaches are not applicable anymore
- Ideas as:
  - AI of course
  - Formal methods
  - Neuro linguistic Software Testing
  - Model Based testing

What are the measures to take?
(input Valid2024)

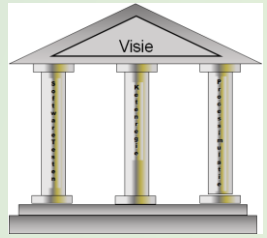


- Design for testing
- Setting up trials
- Scope of innovation; functional or nonfunctional
- Modelling
- Formal correctness
- Statistical / dynamic analysis
- Understanding the context combines with requirements. Smart requirements→ how to validate them
- Prototyping the system itself. Not only the UX
- Looking for the application of AI



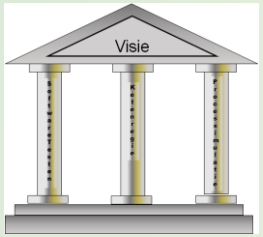
Artificial Intelligence / AI-tooling forms the base to explore further the future of software validation / testing

First draft test approach

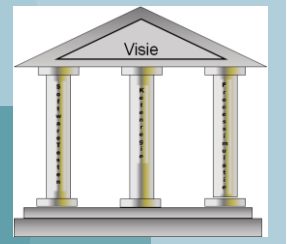


- Test infrastructure
- Skills of a tester
- Test management
- Testing techniques

Required infrastructure



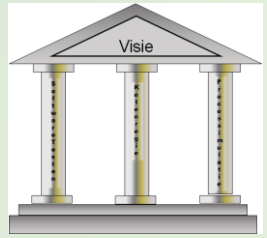
- Tooling;
  - Off the shelf
  - Or tailor-made tools
  - New developed AI-tooling
  - Embedded in existing tooling
- Test environment:
  - Increasing CPU capacity
  - Integration with CI/CD and test automation
  - Version control & reproducibility
  - Management and monitoring



Skills of the new born tester

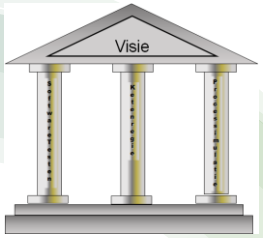
- Subject matter knowledge. Moving to specialists. Different kind of subject matter knowledge (domain knowledge (context), compliance knowledge, system- & conceptual thinking). It must be the next level!
- Able to read the context of the subject for assessing the results of the AI application
- Data science capacities
- The ability of defining the right questions (problem decomposition, prompting)
- The ability of interpreting an answer (critical thinking)
- Assessing the unexpressed requirement related to the oracle
- Statistical analysis
- Being able to think out of the box. Not being afraid for the unknown
- Being able to validate the quality of the model (understanding and valuing the outcome of models)

Test management



Is test management still a necessary role? Maeby it is:

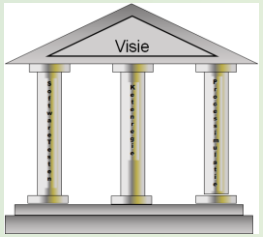
- Organize the shift left. Test as early as possible
- From a resource perspective more business knowledge is required. The context of the system where it must act is increasing in importance. These experts must be selected for the team
- Reporting;
  - Traceability, coverage and analysis potential problems
  - Defining schedules for executing and analyzing the results
- Selection of the necessary LLM's
- Availability of test data will be challenging:
  - The test data itself. AI will support the generation of test data by hand of process models for instance
  - The amount of test data
  - Part of the test data must be used for learning the LLM's



Testing techniques:

- Scope detection system
- Neuro linguistic Software Testing

(part 1) Scope Detection System (SDS)



The idea:

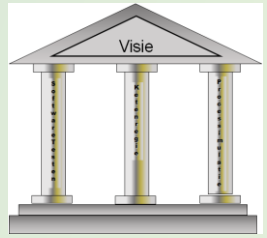
By hand of AI-tooling analysis of the design / user story or release notes on aspects as:

- Validation off the completeness and correctness. Related to possible impact
- Determine the impact on the System Under Test (SUT)
- Defining the test approach (functional & nonfunctional)
- Calculate the required effort based on statistics

The status:

- POC is running for: Package based software for the domain of pension comparison renew of policies
- A short demo; https://regressiontest.mainfact.ai/home

(part 1) Scope Detection System (SDS) one example



mAInfact: Document Regressor | Inrichten nieuwe werkplek (win | E-mail - van Rooijen, Jos - Outl | Hotels in Lisbon - Mercure Lisb | Instellingen: Privacy en beveili

regressiontest.mainfact.ai/docviewer

Google | Huisvoorsoftwarekw... | www.testcommunit... | https://www.cs.ru.nl... | ISO/IEC JTC 1/SC 7 -... | Tele2 | Inloggen ondernem... | Rekeningoverzicht -... | Voordragen | Comp... | Alle bookmarks

mAInfact

Jos van Rooijen deel 2 | Uitloggen

Document A

In the domain of pensions yearly policies must be aangepast. One of the most important functions of the pension systems.

Important from a compliance perspectief.

This test compares the outcome between 2 documents from the last year en the coming year.

When there are differences the tool will detect these differences.

Document B

In the domain of pensions yearly policies must be renewed. One of the most important functions of the pension systems.

Important from a compliance perspective.

This test compares the outcome between 2 documents from the last year en the coming year.

When there are differences the tool will detect these differences.

1

1

Text Changes

A aangepast.

B renewed.

Replace Chars: 50-60 → 50-58

A perspectief.

B perspective.

Replace Chars: 150-162 → 148-160

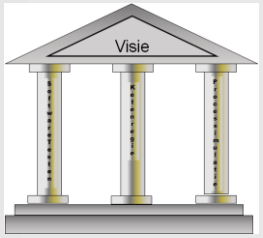
Filter | Tekst | Visueel | Lay-out

1 / 1

22%

JSON downloaden | PDF downloaden | delete

(part 2) Neuro linguistic software testing



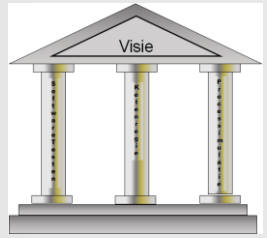
Based on the SDS the following steps are taken:

Basic approach:

- Based on process design/models, use cases, requirements, ?, the AI-bot is looking for all possible flows and decision points
- These models etc. are translated to LLM's
- At each decision point the AI-tooling test all possible functionality (happy flow / worst case). Based on the functionality the required testing technique is automatically selected. Tests are executed based on the available documentation
- Based on the results the AI-tooling decide how to proceed. Probably visible by hand of VR-tooling
- Where necessary the AI-tooling select or generate the required test data. Generated, synthetic data or anonymized
- At the end of the chain or functional requirement the results are reported and can be validated by a human being, a subject matter expert

Neuro linguistic software testing

- Repeat the tests a few times to create a baseline



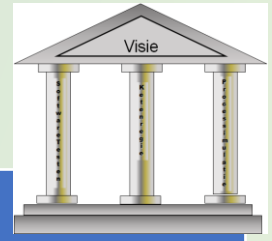
Boundary conditions:

- By hand of parameters definition of the scope, coverage degree, selected testing technique. For instance, system test, condition coverage or decision table testing technique
- Define the required coverage degree, edge cases
- Definition of the quality of the so-called test base by hand of the SDS
- Generation of required test data which is representative for production situation

Open questions;

- Is it possible to validate on a statistical base the quality of the system?
- How to test all the possible interfaces?
- How to deal with the number of possible combinations and / or connections
- How to deal with the predictability of the outcome?
- How to select the proper testing technique?

AI and risks



Reliance in AI

AI testing depends on the data used. What is the quality of the data?

Interpretation of AI results. How to guarantee this?

Unpredictable scope. Scope is expanding by hand of the AI-bot

Much focus on unit testing

Hallucinating can affect test results

AI and Intellectual Property (IP)

Test oracle problem. AI is not always consistent in the results

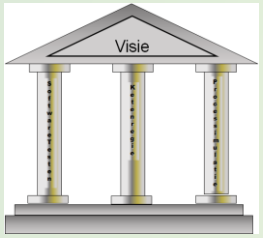
Gathering and challenging specifications and assumptions, communication with stakeholders

The correctness of the LLM. How to prove this? (formal correctness)

No more generalists but just specialists. Knowledge is key for success

Transparancy of algorithm's

Some general thoughts

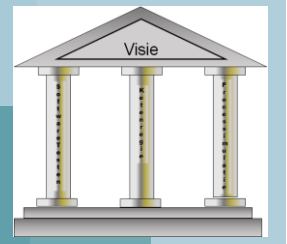


A new kind of education is required. Education 3.0!

Is it necessary to apply always AI? Think about this

Impact of scalability of the solution

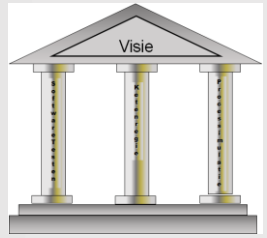
How to deal with joiners? AI request for seniors



Advantages

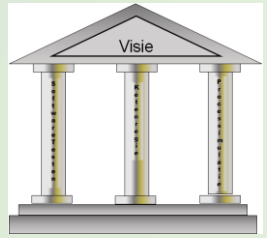
- Less people required
- Increasing quality (increasing coverage degree)
- Increasing speed (time to market)
- Coverage of testing is increasing
- Learning curve is very low
- Decreasing the required testing effort

Summary



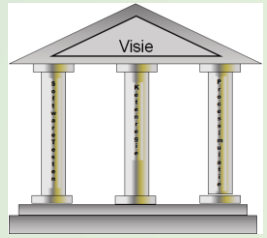
- Software testing is at the moment really traditional
- The question which rise; is the software test community prepared for the future challenges?
- The statement is that this is not really the case
- Some rough ideas are proposed in this presentation
- It requires a lot of effort to reach a mature level
- One thing for sure: nothing is the same anymore!
- New kind of labor is required

References



- Working party ‘Dutch testing society’ – the future of software testing
- https://www.getxray.app/blog/the-top-5-software-testing-trends-of-2023
- https://medium.com/@realtestify/the-road-ahead-8-trends-in-software-testing-for-2024
- https://testsigma.com/blog/software-test-automation-what-to-expect-near-future/
- The future of Software Quality Assurance, Stephan Goericke, 2020
- Artificial intelligence and software testing, Smith et all, 2022
- TestNet; Testen met Artificial Intelligence, 2019

Abbreviations



Abbreviation	Description
LLM	Large Language Model
RNN	recurrent neural network
LSTM	Long / Short Term Memory
IP	Intellectual Property
AI	Artificial intelligence
SDS	Scope Detection System

Questions?

Thanks for your attention.

Jos van Rooijen | 0031 (0)6 -54 90 62 82 | jos@huisvoorsoftwarekwaliteit.com