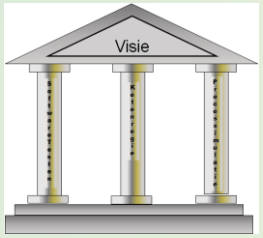
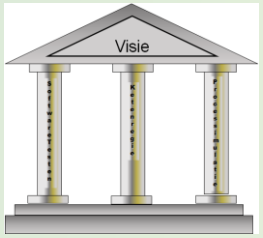


Validation and Verification of packaged based software

Valid 2022

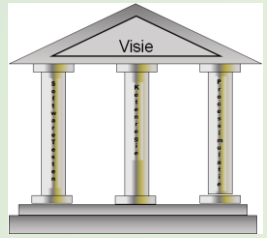
Jos van Rooijen

Agenda



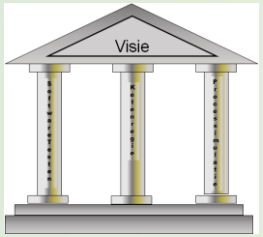
- Introduction
- Context
- Some challenges
- Defining the (test) approach
- Future work

Introduction



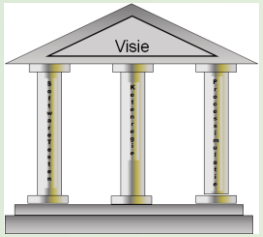
- Consultant at Huis voor Software kwaliteit
- 30 years in software testing & quality management
- Co-author several quality related books
- Test expert online magazine Computable
- Publication areas; Testing, Education and quality monitoring
- Graduation supervisor Avans university of applied science
- Visiting lecturer Universities of Applied Science
- Member of the steering committee Valid conference
- Member of the board Dutch Testing Society

Context



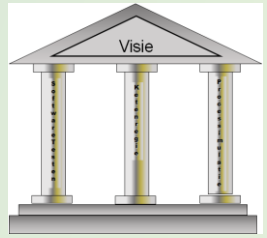
- Companies are implementing more & more packaged based software to support their core business processes
- All these packages must cooperate with each other
- A lot of these systems must be available for almost 24/7
- The question is how to implement such new or updated systems without risks?
- The risk of harm can be tremendous if the package is unavailable

Reasons to implement package based software



- Less development effort required
- Develop faster
- Shorter lead time
- Package itself is stable and in use by many customers!
- On regular base updates available

Package based software

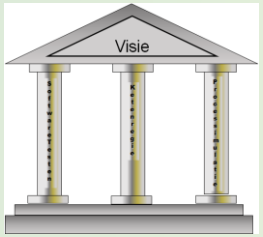


An application program developed for sale to the general public. Packaged software is designed to appeal to a large audience of users, and although the programs may be tailored to a user's taste by setting various preferences, it can never be as individualized as custom-programmed software

Scope of the definition in this presentation is:

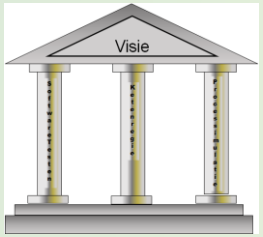
- Business related software

Testing off Package based software



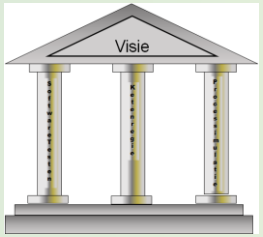
- Why is testing of a package required?
- It is already tested by the vendor?
- Some main challenges:
  - The quality of the package itself
  - Is there a 100% match between the package and the business processes
  - Embedding in the existing IT-landscape
  - Unknown customizations
  - The quality of the data

Challenges from a customer point of view



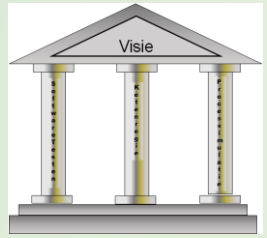
- The package:
  - The quality of the package itself
  - The support of the vendor
  - Adaptability in the It-landscape of the customer
- Knowledge of related business processes
- Knowledge of the current IT-landscape
- The maturity of the vendor
- No. of customisations / custom code
- Frequency of system updates
- The maturity of their own organization

Challenges from a vendor point of view



- Maturity of the customer
- Blueprint available for future based design
- Known / unknown interfaces
- Other projects with possible impact on the package
- Integration of the package in the current IT-landscape
- IT driven or business driven project
- The unexpected! How well are the circumstances known to involved parties?

How to deal with these challenges



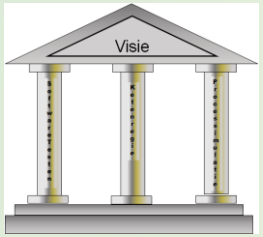
The main question is:

“How to gain trust in the new situation”

3-way approach:

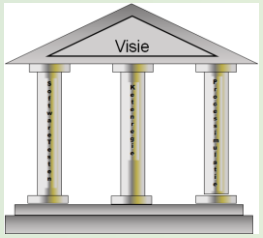
- Apply quality monitoring
- (opt.) define the data migration strategy
- Define the test approach by hand of applying the PRICEPS model:
  - Business perspective
  - IT perspective

The PRICEPS model



- Risk based approach
- Characteristics of implementing a package
  - P = Parameters
  - R = Requirements
  - I = Interfaces
  - C = Conversion (data migration)
  - E = Enhancements
  - P = Performance
  - S = Security
- Base for defining the test strategy

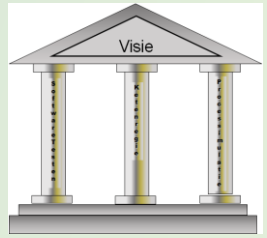
The practicality of the PRICEPS model



Some examples of questions for defining the test strategy based on the PRICEPS model

PRICEPS	Points of attention
P	Blueprint available for parameter settings to tune business processes
R	What are the requirements? What are the priorities?
I	Which interfaces are relevant. First tier and second tier
C	Is data migration necessary
E	Which kind of enhancements are available and/or required
P	Requirements related to performance/stress/load Check also other possible relevant non functional requirements
S	Requirements regarding Attack & Penetration Requirements regarding

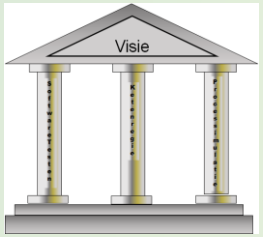
Validation / test approach (1)



- The final result must be:
  - A guideline for what must be tested (functional & non functional)
  - Several elements can be worked out

risico nummer	risico	indiener	kwaliteitsattribuut	prioriteit	Testvorm	Testtechniek	Toets	UT	UIT	ST	SIT	FAT	GAT	PAT	risico in testbasis?
1	VIA sluit niet goed aan op / werkt niet goed samen met WAIB	B&E, BL	connectiviteit	M	Functionaliteit	Programmainterfacetest* Procescyclustest Dataflowtest*					xxx	xxx			
2	DigiD sluit niet goed aan op / werkt niet goed samen met WAIB	B&E, BL	connectiviteit	M	Functionaliteit	Programmainterfacetest* Procescyclustest Dataflowtest*					xxx	xxx			
3	Als VIA niet beschikbaar is, dan werkt WAIB en/of de keten niet goed	TA	connectiviteit	M	??	??									

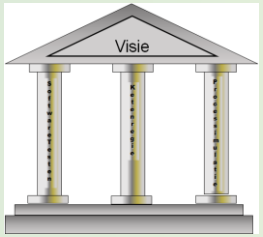
Validation / test approach (2)



General guidelines:

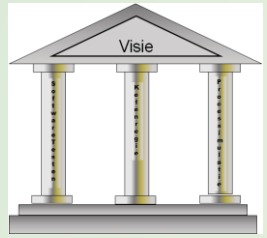
- Test techniques:
  - Path testing among subsystems
  - Process cycle test
- Coverage degree:
  - Path coverage
- Test environments required
- Type of resources required
- Test types required (vendor & customer point of view)
- Stakeholders to be involved

Validation / test approach (3)



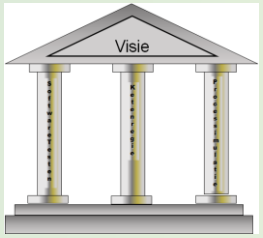
- Tooling:
  - Test automation tooling
  - Test management tooling
  - Tailor made queries
- Data:
  - Conditioned test data
  - (real) test data
  - Production data

Quality monitoring



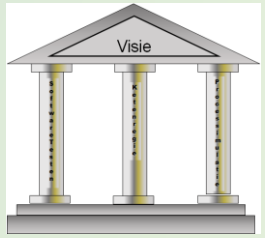
- The aim is to get insight in the quality and completeness of the package delivered by the vendor
- Questions are:
  - Are requirements met?
  - Risks mitigated?
  - Documentation sufficient?
  - No. of bugs insightful?
- Tools to be used:
  - Validation of testware
  - Join system demo’s
  - Cooperate in test execution

Data migration (1)



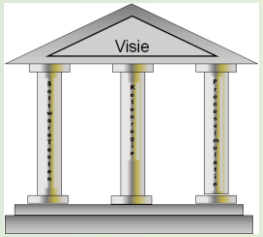
- How to get the data in shape for the new / updated package
- Steps to be taken to migrate data from the current system to the new system?
  1. Define the acceptance criteria
  2. Insight in the quality of the data
  3. Data washing
  4. Defining the migration rules
  5. Test the migration rules with conditioned data
  6. Test the data migration by hand of existing test data
  7. Test the data migration by hand of production data

Data migration (2)



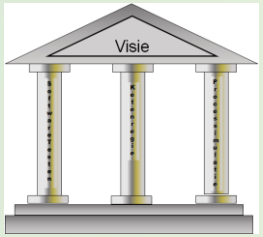
1. Execute some trial runs
    1. Identify failures
    2. Identify the performance
    3. Measure the results
    4. Solve the failures
    5. Repeat this step till no failures are present anymore
2. Combine the new package and the migrated data into a business process test
3. Solve issues
4. Repeat till acceptance criteria are met
5. Use the collected measures as a base for implementation scenario

What are the results?



- Testing:
  - Clear insight what must be tested
  - No surprises
  - Package is incorporated in the IT landscape
  - Increasing confidence in the new package
- Implementation:
  - Business is able to work with the new package
  - Coherence between several programs
  - Business have a better understanding of their own processes and creates a base for further business enhancements
- Applying methods as PRICEPS and quality monitoring delivers the necessary focus
- Data migration:
  - No loss of data
  - From a financial point of view no loss of earning

Future work



- Apply the approach in new package based pension systems
- Apply the approach in new compliancy rules regarding pensions

Questions?

Thanks for your attention.

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