3. Inspection of the requirements, such as a system requirements specification to determine whether the requirements are complete, clear, consistent, testable and maintainable. We will require clarifications of the requirements are ambiguous.
4. Inspection of design documents such as the system design specification to determine whether they are traceable to requirements, complete, clear, consistent and feasible.
5. Inspection of source code using programming languages experts, inspection tools, static and dynamic code analyzers. So we could provide a report on whether we believe it is coherent, traceable to the design, consistent, maintainable and does not contain KPI bottlenecks.
6. Preparing the source code for the semi-automatic model checkers, theorem provers and other tools that perform verification. This stage will add annotations to some parts of the source code, such as pre/post conditions and invariants, to ensure safety, predictability and mathematical correct behavior.
7. Independent testing where a test plan will be developed and an indepedent test will be executed based on the requirements.
8. Deliver ISV² inspection reports, annotation in the source code, generated code, ISV² test summary reports and Periodic reports.

Contacts

VisionSpace Technologies Lda
Rua Alfredo Cunha nº37
Matosinhos, Portugal

VisionSpace Technologies GmbH
Rheinstr. 41, 64283
Darmstadt, Germany

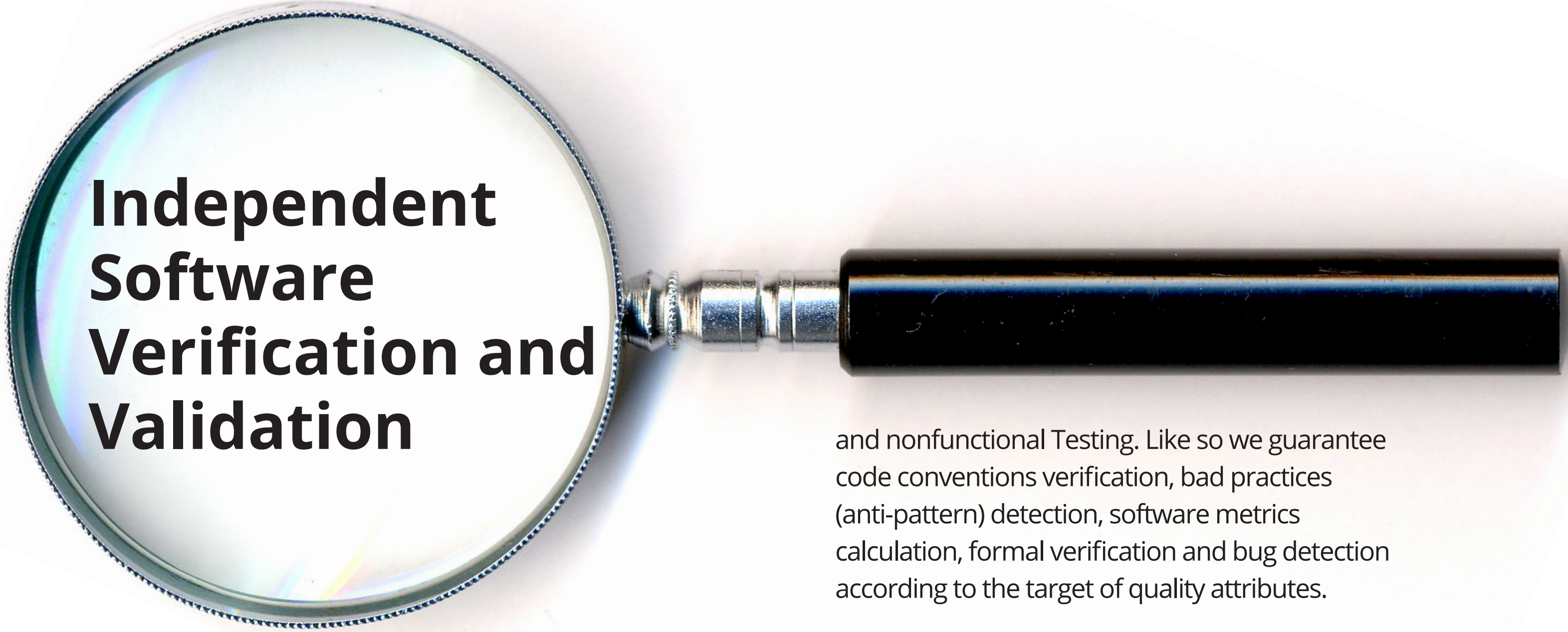
+351220931881
info@visionspace.com
www.visionspace.com

ON.2 O NOVO NORTE
PROGRAMA OPERACIONAL
REGIONAL DO NORTE

QREN QUADRO
DE REFERÊNCIA
ESTRATÉGICO
NACIONAL
PORTUGAL2007.2013

UNIÃO EUROPEIA
Fundo Europeu de
Desenvolvimento Regional

Independent Software Verification and Validation



Independent Software Verification and Validation (ISV^2) aims to improve quality, raise confidence and reduce costs in the software product subject to the ISV2 process. It is also intended to reduce development risks by having an independent organisation performing verification and validation of the specifications and source code.
Raising the confidence is particularly important for critical software, whose failure may lead to hazardous events, loss of life and exceptional costs, damage to health, environmental damage, grave economic losses, or loss of reputation. ISV^2 is therefore targeted to find faults in critical and/or safety or dependability related components.

Verification: This product is right

We assure that the software fully satisfies all the expected requirements, specifications and regulations and we are able to do so by performing Analysis and functional and nonfunctional Testing. Like so we guarantee code conventions verification, bad practices (anti-pattern) detection, software metrics calculation, formal verification and bug detection according to the target of quality attributes.

Validation: This is the right product

Our validation phase is a process of establishing evidence that provides a high degree of assurance that a software accomplishes its intended requirements.
We ensure that development and verification procedures for a software result in the software that meets initial requirements, specifications, and regulations.
We produce software models and use simulations to predict faults or gaps that might lead to invalid or incomplete verification or development of the software. A set of validation requirements, specifications, and regulations may then be used as a basis for qualifying a development flow or verification flow for a software.
We also ensure that modifications made to an existing qualified development flow or verification flow will have the effect of producing a software that meets the initial design requirements, specifications, and regulations.

Our process

Our ISV^2 process is derived from well known aerospace standards and is tailored to span the entire lifecycle of the software project or enter at a later stage and can be adapted to several software development methodologies.
A typical engagement will have a focused effort at key goals such as Determine whether the requirements are complete, clear, consistent, testable and maintainable; Determine whether the design specifications are traceable to requirements, complete, clear, consistent and feasible; Determine whether the source code is coherent, traceable to the design, consistent, maintainable and does not contain KPI bottlenecks and Ensure that the source code is safe, predictable and is mathematically correct.
The emphasis of these goals may vary, depending on the maturity of the software, budget, time and the target of the quality attributes (safety, security, reusability or usability).

1. Kickoff meeting to identify the individuals who will coordinate and participate in the engagement, determine all the assessment requirements of the client, identify resources, establish a timeline and deliverables, brief out the process and the schedule to the participants)
2. Project tracking in our Redmine tool and provide a forecast to complete. Our report will include control panels that show actual vs. planned for milestone completion, functionality produced, cost, effort, and defect rates.