



Service Integration Suite

June 2016

An industrial offer

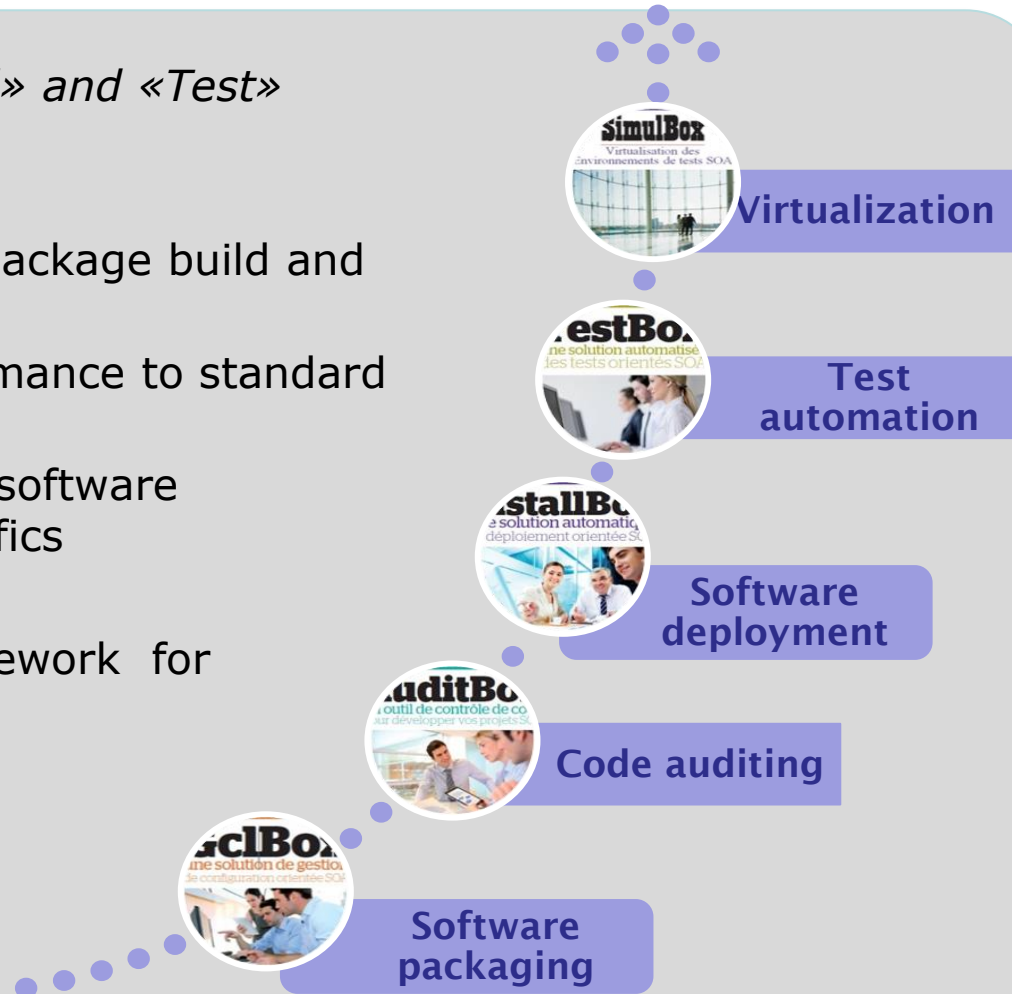
A tool set to sustain the quality of the delivery



A catalog of tool used to secure «Build» and «Test» phases of exchange projects.

- ▶ **ReleaseBox** : Automate software package build and repository management operations
- ▶ **AnalyzeBox** : Verify source conformance to standard and best practices
- ▶ **InstallBox** : Automate application software deployment including solution specifics
- ▶ **TestBox** : Automate test execution
- ▶ **SimulateBox** : Virtualization framework for Information System interfaces

Under product license



An industrial offer

Continuous integration



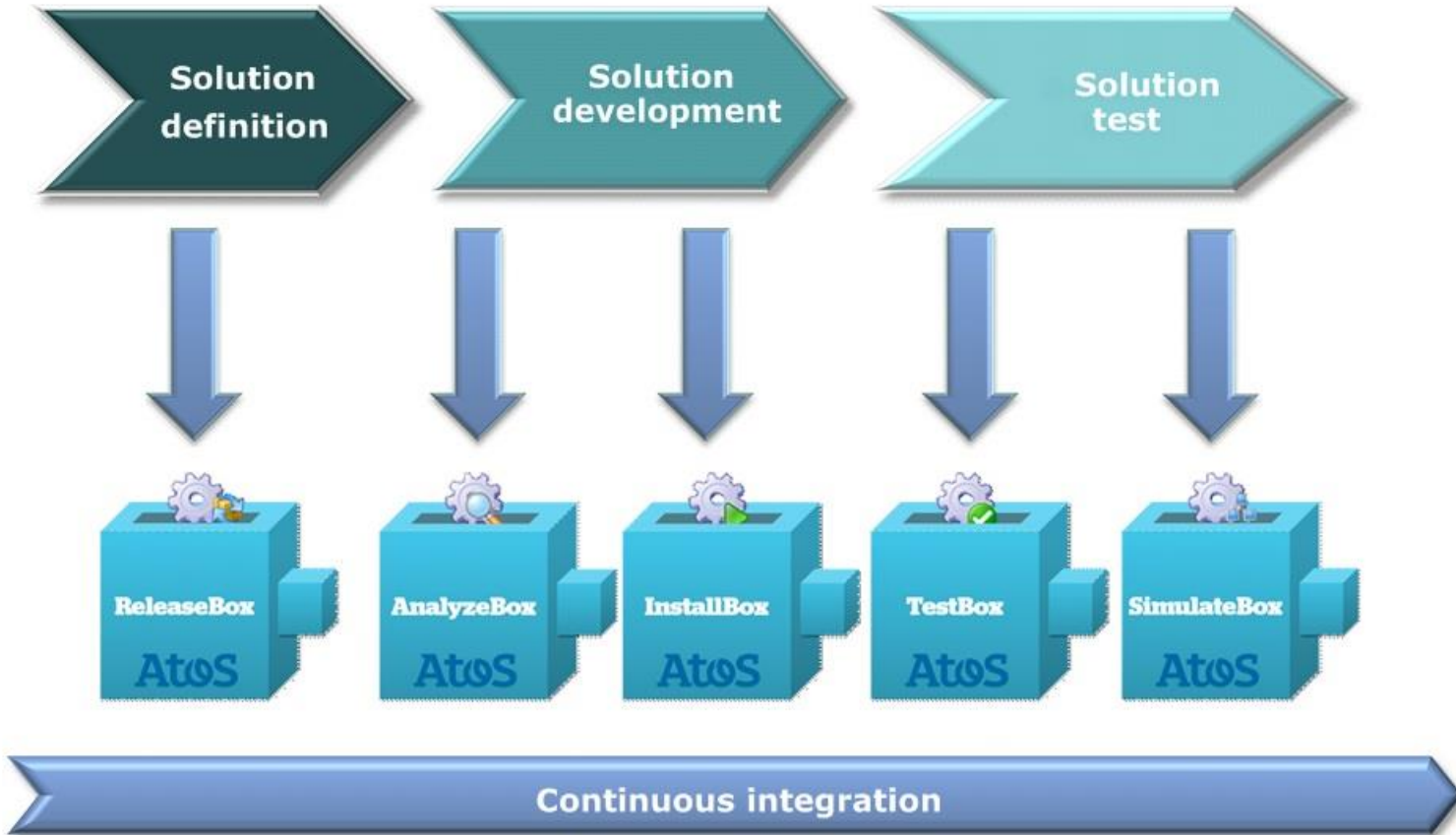
Ensuring quality of our services



- ▶ What is « continuous integration » ?
 - ▶ A frequent internal delivery (daily)
 - ▶ A deliverable automatically created from software configuration management system
 - ▶ Making periodical code review
 - ▶ Running testing campaigns
- ▶ How do this approach should help our customers to improve the quality of their SOA projects ?
 - ▶ Ensure consistency of the version
 - ▶ Identify regressions earlier in the project lifecycle
 - ▶ Measure project improvement
- ▶ Positive feedback on this approach from customers on several projects.
- ▶ Atos has developed a set of specific tools for SOA technologies to automate the test cycle of software components

An industrial offer

Complete cover of a project lifecycle



Service Integration Suite

Configuration management ReleaseBox



► Why this tool ?

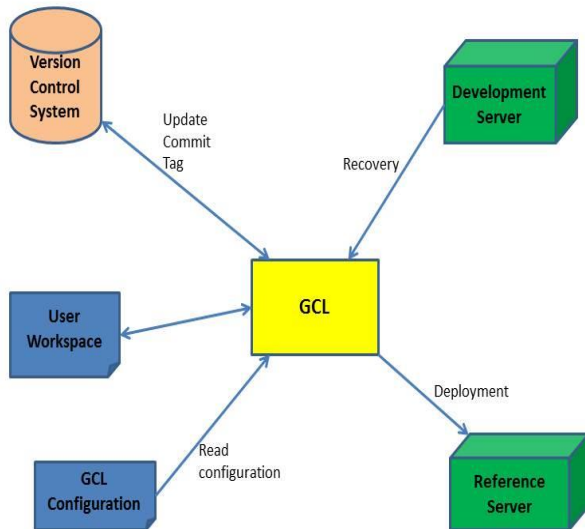
- To get a portable, maintainable and scalable solution

How ReleaseBox matches this goal ?

- Unique frontal interface to support every actions for software versioning and package management
- Overlay on versioning solutions (SVN, CVS, ...)
- Native management of the webMethods and Informatica components
- Automatic deliverable generation (e.g. webMethods, ...)
- Modular conception facilitating new interface developments (versioning solution, components, ...)

What is the gain ?

- **Coverage increasing** : of the assets required to ensure the versioning and the packaging of the software delivery
- **Tasks simplification** : simplify, with its single interface, the manipulation of complex versioning and packaging mechanisms
- **Secure deliverables generation** : improving the delivery package consistency



Service Integration Suite

Code review with AnalyzeBox



- ▶ **Why this tool ?**
 - ▶ Ensure consistency of the code and the quality of deliverable
- ▶ **How AnalyzeBox matches this goal ?**
 - ▶ Looking for unsafe source code that can generate incident at run-time
 - ▶ Detection of prohibited use of services or features at deliverables (eg. backup or restoration of webMethods Pipeline, ...)
 - ▶ Identification of « dead » source code
 - ▶ Preventing service locks
 - ▶ Checking of basic naming rules that ensure the comprehension of the and facilitate the maintenance.
- ▶ **What is the gain ?**
 - ▶ **Reducing the risks of bugs :** controlling in advance the code helps to reduce the risks of defect detected in the following phases of the application build project
 - ▶ **Generating a full report :** helping developers to detect and solve problems
 - ▶ **Improving the developer's productivity :** the periodic control helps the developer in the respect of rules that gives homogeneity and clarity to the source code simplifying bug fixing and evolution.

Service Integration Suite

Deployments automation with InstallBox



▶ **Why this tool ?**

- ▶ To reduce at maximum manual tasks and unavailability periods due to application deployment in critical environments

▶ **How InstallBox matches this goal ?**

- ▶ Full automated: the actions requested to realize a software deployment are completely automated
- ▶ Secure: The rollback management enable retrieving a consistent state in case of failure during deployment process
- ▶ Multi-technology: the solution targets integration software suites (editors and open-source) with an adapted requirement coverage
- ▶ Modular: the architecture of the solution simplifies the adaptation of the tool to meet the regular needs of upgrade.

▶ **What is the gain ?**

- ▶ **Decreasing documentation effort**
- ▶ **Human errors** : by minimizing manual operation, operation errors and misunderstanding are avoided
- ▶ **Reducing delays and unavailability** to operational deployment
- ▶ **Improving security** by including roll-back procedure

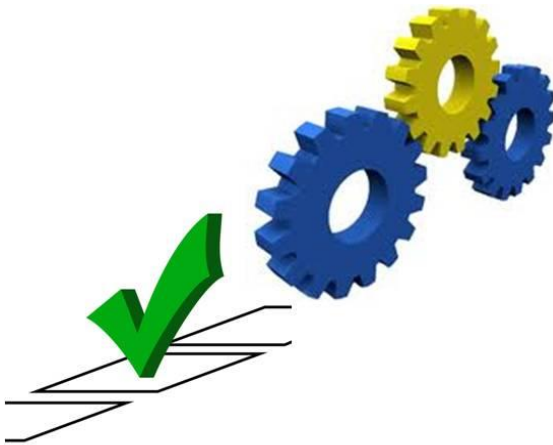


Service Integration Suite

Test automation with TestBox



- ▶ **Why this tool ?**
 - ▶ To easily define automated test cases that suits the technology requirements and/or functional logic of each application
- ▶ **How TestBox matches this goal?**
 - ▶ Portability – TestBox can be set-up on all platforms
 - ▶ Standardization of writing test cases with the XML declarative language
 - ▶ Modularity - organizing within test campaigns, test case sequences according functional areas
 - ▶ Multi-technologies - TestBox manages native webMethods and Informatica components as well as many technologies (DBMS, JMS, email, FTP, HTTP ...)
 - ▶ Integrating – coupling to HP ALM is enabled using Import/Export functionalities
- ▶ **What is the gain ?**
 - ▶ **Simplifying the definition of test cases** : by providing an intuitive language and a modular structure
 - ▶ **Facilitating the evaluation** of tests generating automatically a report of the test session through a comparison with referenced results
 - ▶ **Scalable to technological evolutions**



Service Integration Suite

Virtualization framework with SimulateBox



- ▶ **Why this tool ?**
 - ▶ To test applications independently with a portable solution
- ▶ **How SimulateBox matches this goal ?**
 - ▶ Native management of many technologies (SGBD, Web service, mail, FTP, HTTP, ...)
 - ▶ Simulation of a client : interactions performed by a "Customer" service partner.
 - ▶ Service simulation : simulates the service provided by a partner
 - ▶ « Proxy » mode to record dataset from real exchanges.
 - ▶ Automatic dataset generation from Excel document
- ▶ **What is the gain ?**
 - ▶ **Simplifying the injection flow in the application**
 - ▶ **Complex orchestrations simulation** : allows to test high complexity applications
 - ▶ **Improve agility of Information System** : facilitating tests between applications (decreasing dependencies : availability, planning constraints, security requirements, ...)
 - ▶ **Adaptable to technological evolutions**
 - ▶ **Complementarity with TestBox solution**

