Automated Testing

Are you still testing your SAP PRO as you tested SAP XI 3.0, when the testing requirements and strategies have evolved in the last 12 years?
Questions covered

**What** - is automated testing?

**Why** - do we need to do it?

**When** - should we do it?

**How** - to do it?

**Who** - should be doing it?
International Focus Group for SAP Integration

Global User Group working to on making the SAP integration better. Some of the activity we are working on is:

- **Survey for SAP Integration 10 year edition**

- **International Integration Day in Sweden**
  - Stockholm 5th/ 6th November
  - Link to see more
Daniel Graversen

- Integration Developer
- Started XI 3.0 in 2004
- SAP Mentor
- CEO Figaf that makes testing tools
Michal Krawczyk

- SAP Integration Expert (SAP PO/AIF/IDOC/ALE) - 14 years
- SAP Mentor (over 10 years)
- Partner at int4.com (certified middleware test tool provider)
- 8 times topic leader in SAP PO area - scn.sap.com
- SAP Teched presenter (EMEA/USA)
- book author on SAP Integration (SAP PO and SAP AIF)
Survey result from IFG

Q10 – Further Investment

In which area should SAP invest further to address your company-specific requirements and pain points in regards to SAP Process Integration/Orchestration? (----- lowest priority, ++++ highest priority, N/A= not applicable)

- Monitoring remains a challenge
- Testing is an increasing opportunity for optimization
- All areas are quite close together (within +/- 2 points)

SAP will have a Customer Engagements on testing of SAP PI/PO
Testing
Regression testing is a type of software testing that ensures that previously developed and tested software still performs the same way after it is changed or interfaced with other software.


- A low cost test that tests that all performs unchanged and as expected when any changes in the infrastructure have been applied.
- The more automated the better because it need to be run every time a change is invoked.

Regression testing = Retesting ?
Why automate testing?

- Save developers’ and business users’ time
- Same tests performed each time
- More precise tests and validations
- Speed perspective - FAST
- Cost perspective - less money
- Auto documented testing results
- Trust that you can make changes to the code
When to do automated testing?

Question: how long does it take to test an application interface compared to its development?

- During the project lifetime (rollouts) updates to the existing functionality are be introduced
- Supporting middleware components migrations (Seeburger to SAP B2B add on, dual stack to single stack)
- Allow to install support packs and patches with less effort
  - Save developer and business time
- Ensure that mappings and any other code will not break the existing application interfaces
- Ensure that transports work by testing all objects delivered
Current State of automated testing

Stop Testing your application interfaces - you're not doing it anyway: in contrast to the UI testing, application interfaces (SAP PO, SAP CPI, SAP AIF, SAP IDOC, ALE, BAPI, RFC, etc.) are being tested automatically only in exceptional cases, not as a general rule:

- Who is doing any automated application interface testing?
- Why do you think UIs are more important than A2A/B2B scenarios?
- Who has the “one magic test message” approach that covers every scenario and business rule in the interface?
What happens without proper testing?

- Creates failure in your business
- Impact relationship with partners / customers
- Affect reporting to authorities
- Expensive to your business
  - Extra support time
  - Change management
  - Lost time with business
Myth: Regression testing can be done after all other testing is done (project view)

If you do it at the end what happens when you need to fix something in the development?

Testing should be an integral track in the AD lifecycle to enable as early as possible robust testing -> Speed-up defect detection - Shift Left
Testing in waterfall approach

- **Developer test** - initial run after the development

- **Unit test** - official test after development

- **String testing** - first connectivity test

- **Integrated testing** - all communication is established

- **User acceptance test** - Testing with business users
DevOps/Agile

- Developers (Dev) and Operations (Ops) in one group, probably the case at most SAP PI teams.
- Used in many “internet” companies to ensure and improve their service.
- Key components it to monitor the system, and fix issues.
- All user should be able test the code before releasing it. Not everybody know all scenarios so it can be difficult to figure out how to test it, so there should be a standard for testing.
- Better deployment process to release easier.
DevOps challenges

- Legal requirements for documentation to FDA, Sarbox
- Existing tools focused on waterfall development
- Testing is poorly (too weak or take too long time)
- Logging the changes to the infrastructure
Different types of automated tools

General Workflow tools like HP ALM, IBM Rational

+ Flexible that allow test of the changed objects
+ Covers good the full end to end flow
- Long time to setup tests and variations
- New language to setup tests
- To Test UI and API, to cover B2B and A2A scenario’s programming is required
Different types of automated tools

**Communication protocol specific tools - SOAP UI**

+ Easy and quick to set up test cases for **synchronous** scenarios
- Works only with specific communication protocols (SOAP)
Different types of automated tools

Specific Record Replay designed for SAP middleware (FIGAF IRT and Int4 IFTT)

+ Simple and quick to setup test cases
+ Covers most of the application integration logic
- Not end-to-end testing

Record replay approach

- Select interface to collect test data from
- Automatic collection of test data based on existing data
- Create test case fast based on the collected data
- Run and compare the result

Best Suited for SAP landscapes but may need to be combined in case of non SAP environments only
Which one to choose?

- Why not combine?
- Workflow covers the end to end test
- Record/Replay covers the deep part with many test cases focused on business and or process components
- Different Record/Replay solutions can be combined to test for an SAP business components both the UI interfaces and the B2B or A2A interfaces related to this component
Conclusion

- SAP middleware testing is often wrongly understood - regression testing
- Automated testing needs to happen as soon as possible and not after all other testing is done
- Great middleware developers don’t exist without proper testing tools
- There are many tools on the market and you may need to use a combination of them so your development work is also supported
Q & A

Daniel Graversen - Figaf
Michal Krawczyk - Int4