WHITEPAPER

Ensure Right IT Solutions for Enterprise by Simulating Business Processes

sqs.com

Authors: Oliver Stel
Principal Consultant

Heinz Bons
Principal Consultant

SQS Software Quality Systems Germany

Published: September 2016

Transforming the World Through Quality

OLIVER STEL

Principal Consultant

oliver.stel@sqs.com

Oliver Stel is principal consultant at SQS AG. Oliver is an expert in industrialisation of IT programs. He is focused on project and program acceleration. In different roles in line and program management he has proved that simulation is the key factor in gaining control of a program, even if different implementation trials have failed before.

HEINZ BONS

Principal Consultant

heinz.bons@sqs.com

Heinz Bons is a member of the global service management team supporting services, methods and tools. He is the founder and former Managing Director of SQS GmbH and former COO of SQS AG (1982-2007). Currently he is also member of the supervisory board of SQS AG.

Contents

Management summary

Better requirements and functional specifications as well as early detection of errors and quality issues (shift-left) within IT projects have been vital topics in literature and in practice for decades.

The approach presented in this paper will put a business analyst into a position to identify logical gaps in specifications, fill them with assumptions, indicate the assumptions closing the gaps and discuss them with the process owners. The designed processes will give a clear picture of requirements and projected solutions and the implementation effort. We call this technique process modelling & simulation.

Simulation means we are working with a model of a solution, not with its implementation. Simulations will create a virtual environment to verify whether designed business and system processes contribute to the company's business success. A successful simulation bears comparison with a successful execution of a digital process. This digital process is also the base to fully automated test execution.

The evaluation of commercial benefits is part of portfolio and project management on a very high level (see demand level). It should be part of business analysis and design (see requirements level focused on in this paper) and is finally evident in user acceptance testing and in production itself (see system level). Evaluation in the acceptance phase of IT development is important for creating confidence; with poor quality there are losses of benefits planned or high costs for reworking and loss of time-to-market. Simulation will accelerate time-to-market with up to 90% effort reductions in maintenance in the worst case.

The approach will be illustrated on the basis of two project types – implementation of packaged software and Agile & DevOps software development.

Keywords

REQUIREMENTS MANAGEMENT BUSINESS PROCESS MODELLING

APPLICATION & SYSTEM PROCESS MODELLING

SIMULATION OF BUSINESS SHIFT-LEFT

Introduction

Having good requirements and - based on those - ensuring the right solutions within IT development has been a key challenge for decades. Current solutions discussed and implemented include, for example, Agile, shift-left and DevOps. Waterfall models are out, but they are still the most applied process methodologies in IT development even if a company is "claiming to follow" Agile and DevOps methodologies.

This paper will describe a methodology which "follows" shift-left and which will improve waterfall based development processes as well as Agile & DevOps based project approaches.

Current experience within SQS comes from several projects implementing packaged software, but also from development of individual applications based on waterfall methodology as well as Agile approaches.

What are the possible challenges when implementing **packaged software?**

- We have to understand the processes implemented by the standard product;
- We have to evaluate whether these processes fit our business needs and maybe fit the technical requirements defined by given infrastructure and end users;
- We have to specify requirements for changing processes or adding functionality to the standard product if these changes are really needed (delta requirements to given requirements by the standard product);
- We have to compare different standard products and decide on one of those which fits our business needs best.

What are the challenges when developing and integrating a **new component** within an existing complex IT portfolio based on an Agile & DevOps approach?

- We have to specify good requirements for that new component which will normally be developed within a sprint of 3-6 weeks by an Agile team;
- We have to incorporate existing systems or reusable functional services of these systems to implement the new component;
- We have to incorporate the technical and functional integration of that new component into the existing IT portfolio (see functional and technical processes below);
- Inside the sprint we will define & implement solution requirements as part of the planned development work.

Although the two groups of challenges seem to be quite different, we will describe a solution which is suitable for both project types. Our solution is a shift-left solution that supports both project types and we call it "process modelling & simulation".

Process models

A process model describes all activities executed by humans and/or by machine. A trigger (= Actor + Event) causes an end2end chain of activities, which will be covered by the simulated process design. All inputs and all outputs are predefined in these sequences.

Scenario: The actor (e.g. a bank customer) requires a function (e.g. transfer money from an account) which is covered by defined activities interacting between the customer and the bank teller. The company itself also adds some activities to manage the events at the back end or to be compliant with external and internal banking regulations.

The company has to identify all external and internal events which are necessary to run a business and provide implemented instructions to execute or process these events.

To structure the process portfolio of a company and to reduce complexity in modelling, implementation and testing, we differentiate (see Figure 1)

- the business layer of processes and
- the application layer of processes

Interfaces between the two layers are the business process step (BPS) (i.e. abstract project view), which is equivalent to an application process (i.e. product view) which implements the BPS (except at manual/organisational BPSs).

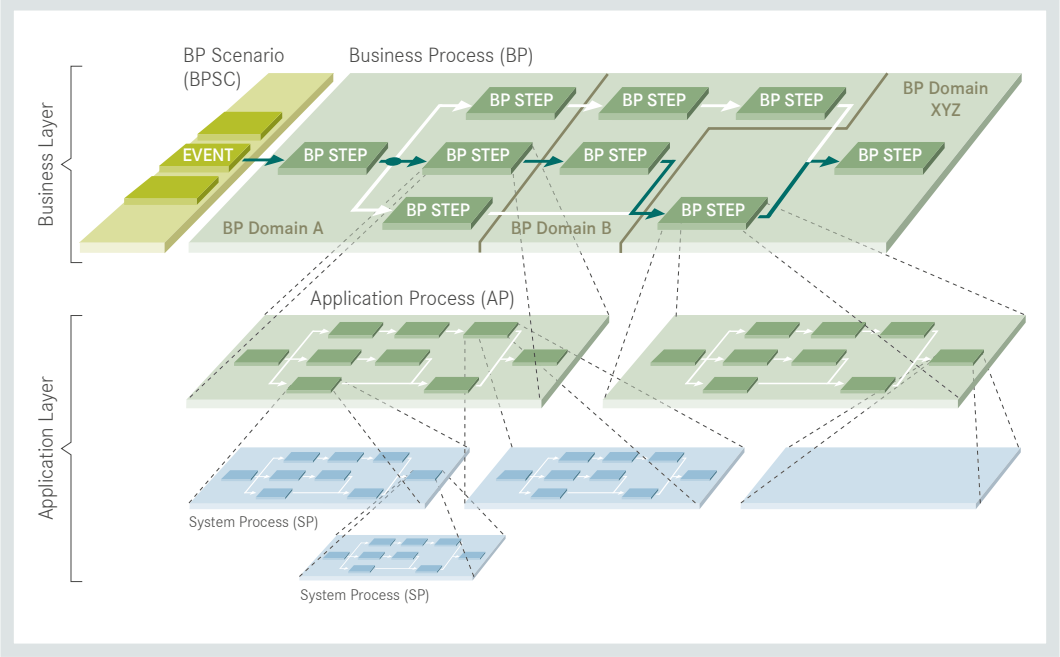

Figure 1: Structure of a process portfolio

The business layer

The business layer describes business events and their processes to work from; see business processes (BP). BPs are specified at end-to-end (e2e) company level. Because of internal division of labour we may differentiate

- BPs at company level
- BPs at domain level and
- BPs possibly at location level (see processes at Frankfurt & New York branch).

A domain has its own defined domain events which have to be processed. A BP has its specific event which initiates the first activity; activities are business process steps (BPS). Key rules for BPSs are

- change of (people's) role to execute that BPS,
- change of application to process that BPS and
- possible re-use of a group of activities.

BP or event scenarios can be specified by a specific model of single business objects (BO) and their processes to handle or manage them (e.g. BO = bank account; processes = create BO, transfer money from BO, get cash from BO, block BO). Activities of a BO model are theoretically BPs, practically often BPSs or application processes.

The business layer goes down in detail to the level of application processes (a BPS is represented by an application process). In the business layer the activities to execute an application process are black box.

The application layer

The application layer specifies application processes and their activities. An activity itself comprises concrete user-dialogs, calculations, comparisons, read and write actions etc. or can be implemented by another system (i.e. system process). With that view we get a vertical integration of system processes which again can be under the responsibility of different domains. The application layer simulates the usage of a concrete system including complete variants management. Moreover, this layer contains all usage knowhow of products like e.g. SAP or PLM.

To answer the question "when to apply process models in these two layers", discussion of another paper is required. We will limit the discussion to the two project types above to give a general idea.

1. Process models for implementing packaged software (waterfall style)

For introducing standard packages the most relevant process layer will be the business layer.

Models in this layer show the business processes using functions of the new IT package. They show the abstract data flows from a business view which are implemented or have to be implemented by the standard package. The business layer consolidates the use cases of business and transfers the processing flows to processes in the application layer. The standard/old application layer will be changed "into" the client specific application layer to cover individual business needs; a design loop will balance business needs and available functionality of the standard package.

The business layer will show the horizontal integration of the new business processes with existing systems which are not substituted and used further in future.

The application layer "implements" the business layer. It furthermore becomes relevant when existing old systems have to be integrated vertically with the packaged systems or when standard processes are heavily changed. Process models help to understand the workflow using the new packaged solution and are supportive in training end users.

Both types of process models can be provided as basic assets from external to support the project.

2. Process models for implementing new software components (Agile style)

Modelling and simulation of processes in the business layer help in portfolio management of Agile projects and in identifying and ensuring good black box requirements (from a functional and a technical view of integration).

Modelling and simulation of processes in the application and system layer help with new development, especially changes within a sprint or in maintenance projects within the lifecycle. Process models easily illustrate dependencies between processes, e.g. of owners within the internal Agile team or even between external teams; process models allow a quick change because of design changes, check dependencies and possible side effects and again automatically generate appropriate test cases to check the implementation of these changes. The time from "specification the change" to "test the change" will be reduced to the optimum.

Currently we are running a pilot to generate automated test cases from application models.

Simulation of process models

Simulation means "playing" our business (business process) by following the different scenarios through the process structure and adapting the models to our experiences and needs. The scenario is represented by a path through a model. By analysing a path, the process owner projects from the experience with a single path to the change of the model. All changes are consolidated within processes and logically verified. This also includes the analysis and information about necessary changes which might impact others. Requirements are directly linked to assumed corresponding activities.

In process modelling every time the business analyst identifies a gap, he takes an assumption and marks it. The maturity of the process is reduced accordingly. If the process owner confirms the assumption, the maturity rises again.

Let us illustrate the analogy with industrial production – designing a vehicle. At first, crash simulations will verify the passenger protection in a vehicle. If the simulations show sufficient results, the casting forges are produced to form the chassis of the vehicle. By analogy with waiting in the production of casting forges until the experience with the simulation is sufficient, in IT projects you verify the overall effect of your business process chain first. In this comparison coding is the casting forge. It is clear; designing a model takes more time in the requirement phase, but the go live is achieved much faster.

By analogy with the sustainable approach in the automotive industry, we apply the same rules to IT projects and their integration in Business Operations. In a nutshell, you find here the features of a simulation based approach. The simulation is contained in a model. From an asset perspective, a traditional IT project is just generating cost. Almost no quality or business operations asset is generated. Normally only the code remains from a project. The documentation is obsolete with the next changes or releases, as it is not updated. With a simulation approach the asset generation is almost 100% in the functional area. Since we are simulating the solution at model level and not the implementation, at this point the knowledge and experience curve using the solution increases.

Simulations are taking care of

- Complex environments to be integrated. The more complex the design of the system, the higher the impact of errors leading to critical showstoppers. The complexity could create impacts on two layers, impacts between actors or components.
- Binding know-how inside a model. The exact picture of an operational solution no longer depends on head monopolies. The model is public domain inside the barriers of confidentiality. This broad knowledge applies to functions, interfaces, allocations of domains and the integrated solution. With the design of simulations, the project and the operations team can maintain replacement chains much more easily. Moreover, the team can release system knowledge into the model and focus on next issues/requirements in the line. Gaps in process knowledge become transparent and can be closed in a planned manner.
- Specification of requirements and their validation fall into one work step. Testing of implemented systems is pure mechanics. Requirements are interpreted into process activities and integrated into the model right away. The certainty of planning goes in synch with the maturity of the processes. The more mature the processes are, the more certain are the time and budget to realise the implementation. If the model is 80% mature, the planning of implementation can be seen as certain.

Our simulation approach is depicted in Figure 2. Depending on the layer we are interested in, there are a couple of steps to be conducted until the final result is provided.

1. We have our requirements at business and/or application and system level (in Agile we call it requirements and user stories). We can also have reviews to improve formal quality of requirements but also to be sure that needs of different user views and groups are covered.

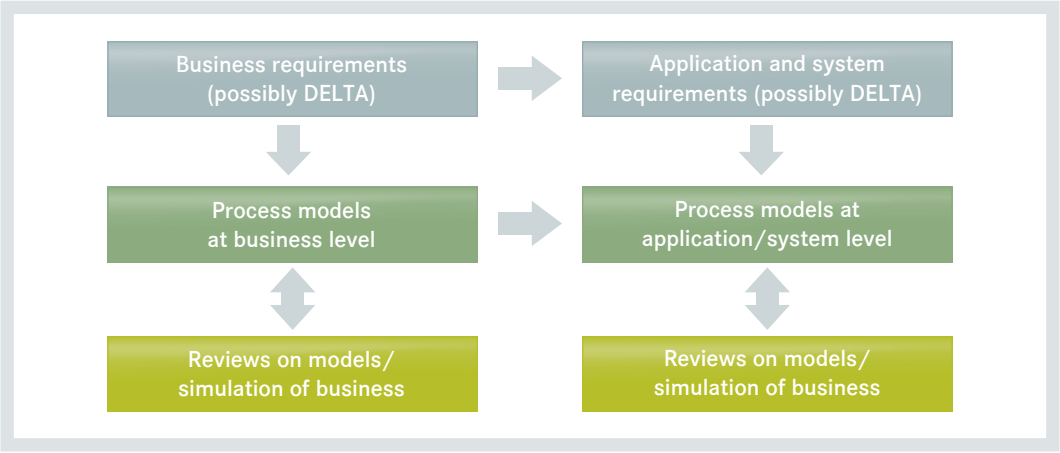

Figure 2: Process modelling and simulation

1. We specify/define process models to make visible how the solutions look in detail according to the requirements specified. We will do that at business level to define business processes for the company end2end or limited to a company's domain; we will do that at application and system level to define application and system processes for implementing required functionality.
2. We run reviews on these process models to simulate the business; executing this we compare processes with business needs and check if requirements are complete and correctly specified. We have a model of the final solutions to make sure that "the solution is what we really require". We can "hand over" the model to developers and testers to improve communication and interpretation of requirements about required solutions. An important part of the UAT against implemented systems ("do we get right solutions") is covered by this kind of simulation. With this effort, the process owner gains already experience with the new solution. The functional UAT will be executed against the model, not the implementation.

Our approach to process modelling and simulation ensures the right IT Solutions for an enterprise, e.g. in projects implementing packaged software it ensures choosing, defining and implementing the right solution; in projects implementing new components – including in Agile style – it ensures that the integrational requirements are well known and understood and that the IT portfolio is ready to take over the work results of defined sprints.

For both project types it is valid to derive test cases from the process models:

- test cases from the business models are applied in function integration, end2end testing and user acceptance testing (for company or domain);
- test cases from the application & system models are applied for function and interface testing and can be extended to end2end testing, e.g. in case of automated execution.

In case of changing systems we first change the process models and secondly generate necessary test cases for testing and regression testing.

Benefits of process modelling & simulation

There are a couple of benefits that have been proven valid, e.g. simulation model as single point of truth, cost savings through simulations, and sustainable investments.

1. Profile of a simulation model

The simulation model represents the single point of truth. It covers the entire configured project scope and release management for ALL functional information of a project/program. It shows all paths through that model, which implies containing all logical and concrete test cases.

The coverage is configurable and execution times can be set to be the basis of planning.

With all effects you will save 30% in the design phase, due to full transparency of productivity. The emphasis is directed to the specification & design phase. The rest of the project phases are just execution.

2. Cost savings using simulations

The model is free of redundant documentation and free of break of media. The implementer codes the model once, since the simulations are comprehensive and with a minimal number of logical errors and incompletenesses. Implementation is reduced by 30% – 60% due to full content commitment of the process owners.

Test case design is reduced by up to 90% effort. It is up to program management to generate test cases from more process layers or just one. Change of requirements will always be integrated into the process models first. Test cases to be executed or regression tested will be generated again and again from these models.

With the next release of a process domain, the costs for specification & design will drop by up to 90%, since only new requirements have to be integrated into running business operations.

Less training costs. As a result of gaining experience while designing the paths in the simulation, the participating users become experts in the tool usage of the target solution. Since the simulation is in fact the detailed description of activities to execute the process variations, the simulation model is a user guide at the same time.

3. Investment in simulation

The creation of a formal process path is more complex than the creation of a logical test case. A challenge is to think in business processes, rather than operational execution activities; there will be an increase of up to +30% costs on the project learning curve, but afterwards it will go down to zero in business operations!

The need for discipline in applying modelling notation vs. benefit from early error detection will increase up to +10% costs in the requirements engineering phase; this will be break-even at the end of the project. It is a one-time investment in process areas with very low or no subsequent costs.

After the setup, the effort is approx. 30 minutes only for a new functional requirement.

Conclusion and outlook

There are four trends in IT operations, which will dominate the next decades:

1. Trend towards standard IT packages, away from proprietary solutions.
2. Products are taking on their own identity (internet of things).
3. The end customer is insisting on individualisation of the product he is using or intending to buy. Hence variances in production, purchases or offers will rise.
4. Industry standards will enforce system interfaces (see Industry 4.0). IT objects, for example, will communicate with each other automatically. Example: Mercedes-Benz is already installing car to car information. If one vehicle detects a danger, it informs other vehicles (currently of the same make) to slow down.

The four trends will lead to a dominant share of standard software in industry. Configurable standard software will contribute to a shorter path in time-to-market. But the decisive step in time-to-market will be achieved by providing process models to simulate the usage of standard packages as well as individual packages. Business will gain experience before the software is implemented. The method is modelling & simulation, which in the automotive industry has proved to be very mature. Companies able to logically integrate their IT solution even before implementing it will be ahead in the race to gain market share.

SQS is offering complete simulation models for ERP implementation into client companies. SQS is building up a product oriented standard software simulation repository (e.g. ERP) initiated with products with high market shares. There is a strong fence between proprietary customer simulation models and product models like SAP or Siemens PLM. Product models are re-used, since they are owned by SQS. Proprietary customer models are strictly confidential.

SQS solutions are based on Visio from Microsoft or BIC from GBTEC (see www.gbtec.de) adding an SQS notation and technical extensions on top with respect to our approach, especially with respect to subsequent testing process requirements. SQS can upload models to any BPMN2.0 compatible BPM solution. SQS extensions are supporting project management, simulation and logical and concrete test case extraction; extraction of automated test cases is under development. Transfer to BIC from GBTEC is under analysis.

SQS Software Quality Systems AG
Phone: +49 2203 9154-0
Fax: +49 2203 9154-55
info@sqs.com | www.sqs.com