Modelling Microservices in Email-marketing
Concepts, Implementation and Experiences

Martin E. Brüggemann, Raoul Vallon, Aykut Parlak and Thomas Grechenig
Research Group for Industrial Software, Vienna University of Technology, Wiedner Hauptstr. 76, 1040 Vienna, Austria

Keywords: MDE, Model-Driven Engineering, Enterprise Modelling, Microservices, Email-Marketing, BPMN.

Abstract: Our experience with email-based marketing campaigns (or short: "Microservices") showed that they are an intersecting set of a) projects and processes and b) technology and creativity. Their properties of fixed due date, fixed scope and at the same time fixed duration render classical management methodologies unfeasible. The same applies to the supporting enterprise infrastructure architecture, where ad-hoc changes lead to a loss of stability and performance. The remedy was found in Model-Driven Engineering; a choreography for Microservices helped to indicate improvement opportunities for both the architecture and the information flow within, which in turn increased throughput.

1 THE CONCEPTS OF EMAIL MARKETING AND MICROSERVICES

The branch of email-based marketing services, or short, "eMarketing", is characterized by rather low market entry barriers, since the product "email" does not show enough technical complexity to allow for sufficient differentiation. The resulting wide array of competitors led to a steady decline in prices, which is why for several years now, leading eMarketing providers diversify and render ever more customized and ever more complex services to their customers.

To avoid a competition for price leadership with unknown outcome, eMarketing providers currently move away from technically-aligned providers of specialized software for sending email campaigns, towards agencies for full-service marketing campaign creation: from idea to text production to translation and evaluation of feedback.

The challenge in this transformation from software providers towards full-service agencies lies in the fact that constraints are more than ever externally defined, with only a narrow margin to adapt to changes. In a controlled environment (PRINCE2, 2009), a set of requirements is used to develop detailed specifications and subsequently a project plan. The plan is then executed within the constraint triple of time/scope/resources. Every change to one of the parameters can easily be compensated by the other two. The challenge after the aforementioned transformation and the major problem in eMarketing today is that now all three constraints are defined by customers and partners, resulting in an enormous pressure to further shorten lead times. Lead times of individual eMarketing campaigns currently average 4-8 working days – a time span in which the entire workflow is run; from requirements analysis to implementation, to testing and deployment.

In contrast, the typical iteration length in Agile approaches is 10-20 working days, which is too long in the field of eMarketing and the associated campaign management.

One could argue that these are not stand-alone projects, but rather individual artefacts in a parent process. This is not the case, since an eMarketing campaign is "[...] a targeted, one-time venture, which consists of a set of coordinated, controlled activities with start and end dates [...]", which is exactly the definition of a project from PMI (2004). Well-documented Lean approaches (Anderson, 2010; Knieberg, 2010; Ladas, 2008) are based on the assumption that development follows one underlying process, with discrete process steps, all of which repeatable. Thus, by definition, they describe multiple tasks of a single project, contrary to the requirements outlined above.

Program management (Hanford, 2004; Larman, 2009) shares some resemblance with management of eMarketing campaigns – both disciplines coordinate

Brüggemann M., Vallon R., Parlak A. and Grechenig T..
Modelling Microservices in Email-marketing - Concepts, Implementation and Experiences.
DOI: 10.5220/0005000800670071
In *Proceedings of the 9th International Conference on Software Paradigm Trends* (ICSOFT-PT-2014), pages 67-71
ISBN: 978-989-758-037-6

multiple projects. With eMarketing campaigns however, the frequency is much higher than with traditional projects, with their lead time of days, rather than weeks or months.

At the same time, following the definition in (Krafzig, 2004), a Service is a self-contained unit of functionality, built around the principle of separation of concerns.

Since there are no two identical campaigns, each is created individually, and dozens of individual campaigns are required to accomplish a marketing goal (the service), every individual campaign is merely a "Microservice", a term proposed by the authors to reflect the semi-project/semi-service properties.

The authors described an environment of intense competition and cost pressure, resulting in the need to reduce lead times and the desire to offer a multitude of products, features and services. This balancing act of "citius altius fortius" can only be solved by massive component reuse and by uncompromising automation of processes – but how can this best be accomplished?

Due to the strong parallels between software and Microservices, namely intangible output and high degree of creativity, we intend to apply software design approaches in this context. Therefore, the aim of this paper is to use Model-Driven Engineering (MDE) (DSouza, 2001) to indicate improvement opportunities for both the architecture and the information flow within. Therefore, we aim to develop a process model which enables fitting the entire value stream of a Microservice from planning, design, implementation to testing into an average lead time of 4-8 working days. This process model would have to allow for multiple Microservices to be processed in parallel, in a continuous flow of work packages, each of which with minimal scope. The benefit of applying MDE in this context is to be verified by an increase of throughput (measured in Microservices completed per period of time).

This leads us to formulating the research questions below:

RQ1. What is an optimal approach to model Microservices and their interaction?

RQ2. (based on RQ1): Applying MDE, would modelling of Microservices lead to identification of improvement opportunities?

2 METHODOLOGICAL APPROACH

The data for this exploratory case study was provided by an eMarketing provider based in Vienna, Austria, with about 400 employees worldwide. Data was analyzed for only the department that was scope of the modelling approach described in the course of this paper. That department is located in Berlin, Germany, ensuring that the authors were neutral observers. Since data for the baseline (Section 2.1) could only be provided for 9 subsequent months, this was selected as the observation period.

In order to tackle the above research questions, the following methodological approach is pursued:

After defining a baseline and measuring the status quo, we are approaching the first research question in Chapter 3.

The identified modelling approach is then applied to the workflows within the department selected for this case study (Chapter 4); the intended outcome being a full business process choreography. In order to be fully usable, this choreography is to be embedded into the existing enterprise architecture. In order to validate or falsify RQ2, again data from nine months is collected and compared to the prior baseline (Chapter 5).

2.1 Establishing a Baseline

In order to allow for resource planning and – allocation, a constant load would be ideal. Figure 1 however shows the oscillating resource load or, in other words, shows how resources are alternating between being overloaded and idling, which in turn might lead to employee burnout (Maslach, 2001).

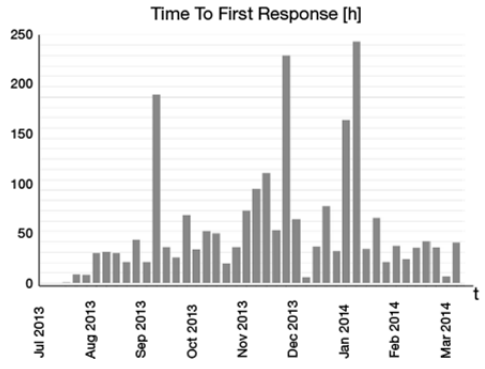

Figure 1: Oscillating resource load, plotted over 270 days.

The impact of such a lack of resources is shown in Figure 2, which plots backlog growth over a period of nine months.

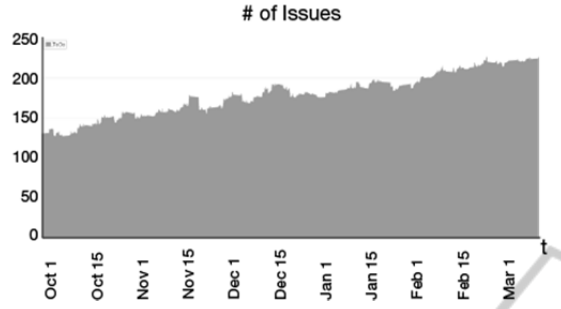

Figure 2: High resource load leads to growing backlog.

2.2 Intended Outcome

To a marketer it comes rather natural to strive for a multitude of offers, with the upper limit set only by the number of customers, which leads to further complexity of the existing infrastructure.

Such an approach is almost impossible for a system architect to work with, as without proper planning the resulting "universal" platform will simply offer low stability and mediocre performance at best.

To tackle the described challenges, a team of system engineers and system architects was soon able to isolate two solutions as the ones to be most promising:

(i) A stop was to be put to ad-hoc customizations and changes, which would be a culture change in itself. All too often, these little tools and scripts are not only poorly documented, but are also implemented under severe time pressure. The customer has already been billed, but the costs of side effects and symptoms are occurring elsewhere in the infrastructure and can rarely be measured - so the ROI calculation is distorted. The ad-hoc approach of engineering was to be replaced by MDE, using high-level models to find improvement opportunities in both productivity and quality.

(ii) Secondly, in order to deliver solutions faster in the future, it was obvious that the ability to reuse whole system components (not just pieces of software) had to be improved. This way, marketing and sales teams can both have their will, and every customer indeed receives a different product, but always one made of existing building blocks, never made from scratch.

The combination of the aforementioned solutions would allow for advancement of the existing architecture towards a true Enterprise Service Bus (ESB) (Menge, 2007). From the beginning, this architecture was designed to be applicable not only to the software aspects of the business, but end-to-end, from project controlling to procurement, to feature deployment and billing. The initiative was built around an idea from Arsanjani (2004), which was ideally suited to combining computational and human resources in one infrastructure.

The described set of solutions gives the impression to be straightforward and easy to implement. However, the envisioned goal comes with its own challenges:

Doing away with ad-hoc customizations reduces revenue in the short term. It requires a great deal of management support to say “no” to a customer’s request due to the infrastructure currently being refactored.

3 MODELLING FOR MICROSERVICES

In order to approach RQ1, a modelling approach for the context of Microservices had to be identified. Methods for modelling large software systems are well described and widespread (Arsanjani, 2004; Benguria, 2007; Dsouza, 2001). The first impulse was to treat products, features etc. as discrete objects, and consequently use Object Oriented Modelling (OOM) approaches. However, OOM approaches have been developed for the modelling and construction of individual large software systems, not for modelling the communication between and the interaction of multiple system components. See (Lee, 2012) for an overview of OOM shortcomings.

The same holds true for the modelling of business processes:

Using BPMN (ISO/IEC, 2013), it is possible to model a business process or a choreography of multiple processes. BPMN models can be checked for validity, but just as was mentioned in the previous section, there are no means for “process discovery”. The great advantage of BPMN, however, is that it is easy to learn and is comprehensible to both technical and non-technical staff, to both domain experts and laypersons.

For all its advantages it was decided to use BPMN and to overcome the shortcomings mentioned by building upon work of Benguria (2007) and Gietl (2010). The result was that we found an optimal modelling approach, which allowed for modelling the individual business units, departments and also hardware and network

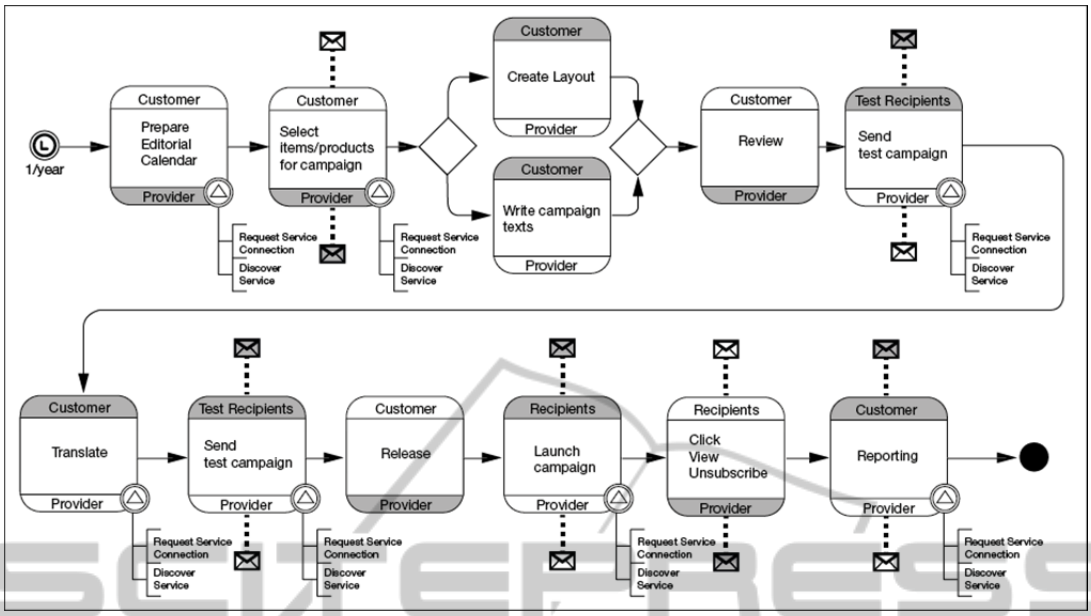

Figure 3: A Choreography for Microservices.

infrastructure within one unified model. All components were to be autonomous independent elements, or short, "Services", that can be described, discovered and orchestrated (Krafzig, 2004).

Therefore, it was necessary to encapsulate (and later utilize) non-computational resources and provide an interface to the existing infrastructure, much in the same fashion as Amazon Inc. demonstrated with their "Mechanical Turk" (Berinsky, 2012; Paolacci, 2010).

Encapsulation of human processing steps and linking them to an ESB would allow for an optimal scalability and also load balancing.

The modelling approach thus identified will now be used to verify our RQ2, namely if the application of MDE in this context of Microservices would lead to the identification of improvement opportunities.

4 TOWARDS A TRUE MICROSERVICES ECOSYSTEM

In a first step, the process chain was modelled from initial customer request to implementation and quality assurance, with strong focus on the message flow and the sequence of messages being passed. In order to do so, a department was selected for modelling whose process steps incorporated the highest level of uncertainty, but at the same time showed the highest costs – resulting in the highest potential for optimization.

The result can be seen in Figure 3. The reader might recognize the strong resemblance to BPMN choreographies, but note the additional elements to allow for service description and –discovery.

Furthermore, the choreography depicted in Figure 3 is embedded into the overall enterprise architecture. Figure 4 shows the architecture as a layer model – note the embedded choreography at the top.

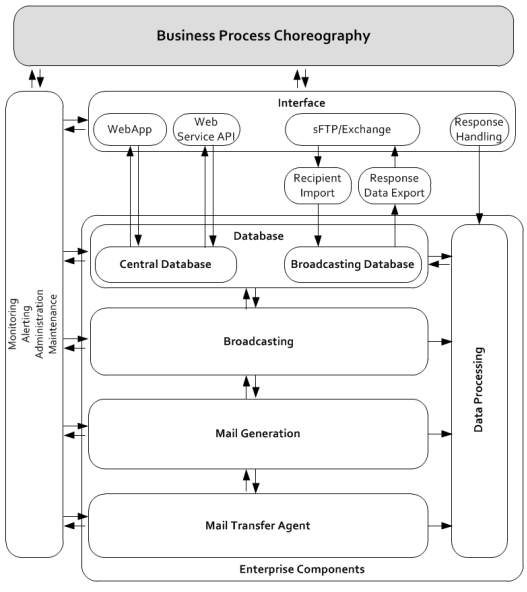

Figure 4: Enterprise architecture (layer model). Microservice choreography to the top, human processing steps shown to the left, all other: computational processing steps.

5 EXPERIENCES AND CONCLUSIONS

The benefits of this approach could be seen instantly: not only have areas for automation and streamlining been identified, but entirely new business models, where the resources freed up by the optimization efforts could be put to proper use. A streamlined information flow resulted in an increased output, as Figure 5 depicts. The slope of the backlog growth curve is now negative:

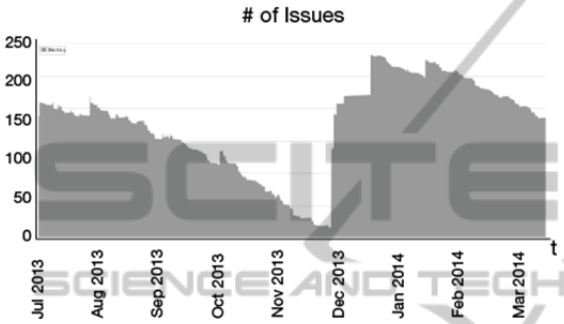

Figure 5: Reduction of backlog request volume due to increased throughput. Note the backlog increase due to new requests being added at the start of new year, resulting in a non-continuous curve.

Encouraged by the positive results showed by our new approach of using MDE in the context of Microservices, the authors are planning to carry out subsequent studies with an increased scope.

We are considering to extend the application of MDE in areas that have similar properties and challenges and incorporate their processes into the existing choreography. This area of application lies on the ingress side of our model and is formed by internal and external suppliers. Examples would be the processes for text creation, localization, graphic creation and combinations thereof.

We consider measuring such a comprehensive modelling and validating any synergy effects from future work.

REFERENCES

Andersen, D.J. 2010. Kanban: Successful Evolutionary Change for Your Technology Business, Blue Hole Press, 2010.

Arsanjani, A. 2004. Service-oriented modeling and architecture. https://www.ibm.com/developerworks/library/ws-soa-design1/

Benguria, G. et al. 2007. A Platform Independent Model for Service Oriented Architectures. *Enterprise Interoperability*, 2007, pp23-32

Berinsky, A. et al. 2012. Evaluating Online Labor Markets for Experimental Research: Amazon.com's Mechanical Turk. *Political Analysis*, 2012

DSouza, D 2001. Model-Driven Architecture and Integration: Opportunities and Challenges. *Model-Driven Architecture and Integration 2001, pp1-32.*

Forrester, J. W., 1992. System Dynamics, Systems Thinking, and Soft OR. *System Dynamics Review*, Vol 10, No 2.

Gietl, F. et al. 2010. Annotationsbasierte Prozess-modellierung in SOA – dargestellt an einem Beispiel aus dem Precision Dairy Farming. *GI Jahrestagung*, 2010.

Hanford, M.F. 2004. Program Management: Different from project management. http://www.ibm.com/developerworks/rational/library/4751-pdf.pdf

ISO/IEC19510 2013. Information technology - Object Management Group Business Process Model and Notation. http://www.iso.org/iso/catalogue_detail.htm?csnumber=62652

Knieberg, H., Skarin, M. 2010. Kanban and Scrum: Making the Most of Both, C4Media, 2010.

Krafzig, D. et al. 2004. Enterprise SOA: Service-Oriented Architecture Best Practices, Prentice Hall, 2004.

Ladas, C. 2008. Scrumban: Essays on Kanban Systems for Lean Software Development, Modus Cooperandi Press, 2008.

Larman, C., Vodde, B. 2009. Scaling Lean&Agile Development. Thinking and Organizational Tools for Large-Scale Scrum, Addison-Wesley, 2009.

Lee, S.-H., Yoo, H. 2012, Requirement Analysis for Aspect-Oriented System Development, *Lecture Notes in Electrical Engineering*, Vol. 215, pp 1201-1209, 2013.

Maslach, C. et al., 2001. Job Burnout. *Annual Review of Psychology*, Vol. 52, pp397-422, 2001.

Menge, F. 2007. Enterprise Service Bus. Proceedings of Free and Open Source Software Conference, 2007, pp1-6.

Paolacci, G et al. 2010. Running Experiments on Amazon Mechanical Turk. *Judgment and Decision Making*, 2010.

PMI Project Management Institute 2004. A Guide to the Project Management Body of Knowledge (PMBOK) 3rd ed., pp5-6.

PRINCE2 2009: Managing Successful Projects with PRINCE2, Axelos, 2009.