# Die endliche Geschichte – es war einmal ein Featurerequest

REconf 2009

Carsten Sensler, T-Mobile Deutschland, Andre Karalus

## Table of Contents.

# Table of Contents.

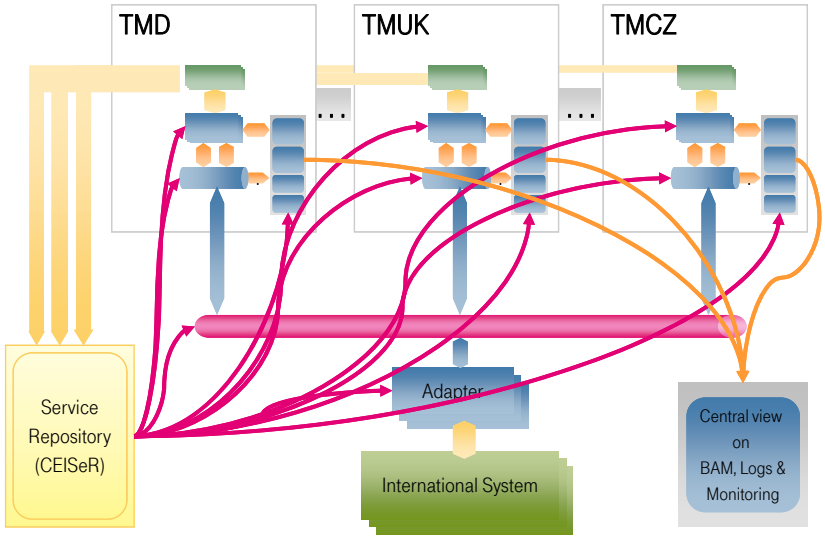




# Technical environment –SOA Backplane Program.

- SOA Backplane will deliver a number of software systems and standards, namely
  - a service bus which is the SOA communication infrastructure
    - Service repository
    - Access layer framework
    - Basic messaging infrastructure (JMS)
  - additional value adding components and functionality including
    - logging, monitoring
    - service contract management
    - business activity monitoring
    - transport components for B2B communication
  - the Backplane Guide and SOA Governance as a set of guidelines and rules as to how SOA will be implemented within T-Mobile.

## SOA Backplane – overview (international view).






## Organizational environment.

- The SOA Backplane is an international program for all national companies of T-Mobile
- One of the main goals is to deliver an enterprise integration infrastructure for all countries
- There are several stakeholders from each national companies with different interests
- Each national company has got his own focus and challenges. It is not unusual, that some of these challenges are in a diametrical opposition. It is caused by the different local situations

## The applied software development process.

- The context is technical, thus „business demands“ are well understood by IT experts
- No need to overcome the gap between business requirements and a formal system analysis
- The requirements can be grasped easily and documented in the appropriate granularity
- TestCases regarding functional requirements can be found easily, for non functional requirements it‘s always hard!
- The team is rather small – an agile process is established to reduce project management overhead



## The applied software development process (2).

- The development of new functionalty is test driven, i.e. before adding new code a test case is written
- Iterations are rather small (usally one week)
- Each day a SCRUM like meeting is established (short standup meeting, where everyone briefly describes progress and problems)
- Continous integration builds are done with running automatic regression test suites (TeamCity)

## Table of Contents.

- Prologue – πρόλογος
  - Description of the technical environment
  - Description of the organizational environment
- Peripety – Περιπέτεια
  - The breakpoints during the journey from a feature request to the implemented feature
  - Overview about the tools of the fairy
- Epilogue – επίλογος
  - Improvements
  - Lessons Learned
  - Summary

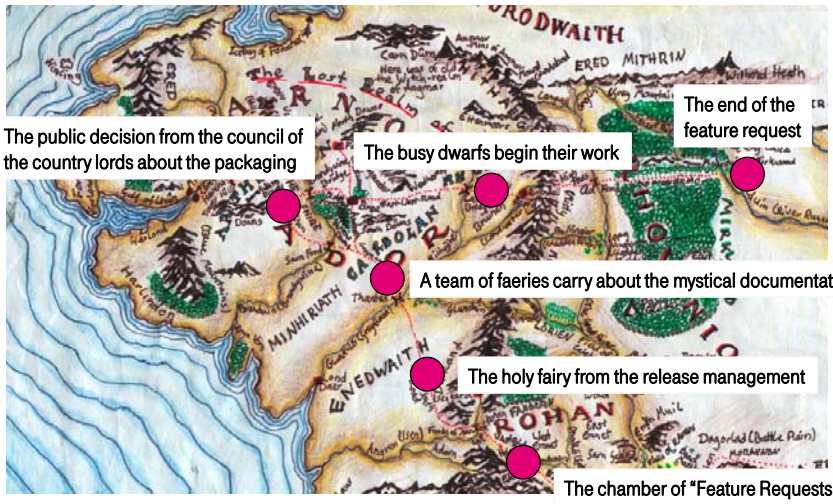


## Question before we start the journey – we will get the answer at the end of the journey.

- Do we need a heavy and complicate software engineering process or is it also with a lightweight and agile process feasible (with the background of the technical and organizational situation)?
- Well known simple and efficient tools and straight forward processes?
- Or better heavy tool chain with a rough complex process?

Let's start the journey to find the answers

## The breakpoints during the journey for the quest.






## The chamber of "Feature Requests".

- If a feature request is raised, the requestor needs to provide some information
  - Who am I (role)?
  - Who is the technical contact/expert?
  - Short description about the needed functionality
  - Priority
  - Desired release
- The weapon of the requestor
  - Microsoft Word

## The holy fairy from the release management.

- Each feature requests results in a document called functional specification
  - A fairy from the team will take care about the feature request (it is the feature owner) and he will interview the requestor and the technical expert
    - The requestor for getting all requirements
    - The technical expert for rough technical details
- Within the release management each feature request will be evaluated and ranked
  - A business case will be computed
  - A rough impact analyses of involved components
  - A rough effort estimation to get an imagine of the costs
- The weapons of the holy fairies are:
  - Microsoft Word, Excel and for the documentation management Subversion.
  - Of course enough time for having several meetings with the participating colleagues



## A team of faeries carry about the mystical documentation.

- The functional specification are written within this team with support from the development and requestor
- Afterwards the documents are spreaded around the countries to get review feedback (it is not unusual, that somebody else from a different NatCo as the requestor needs something similar and provides slightly different requirements)
- One week later, the documents will change their state from “proposal” to”reviewed”
- The team rework the documents and when they finished, the status is “agreed”

## A team of faeries carry about the mystical documentation.

- The functional specification documents are the basis for writing design documents. The design documents are more technically and solution oriented.
- The technical expert/ development department estimate the effort for the realization of the feature
- A detailed impact analysis and risk estimation is done in this documents
- The detailed design documents are the basis for the test department to define the test cases and to control the completeness of an implemented feature.



## The public decision from the council of the country lords about the packaging.

- A proposal of the features for the next release is presented by the release management to the EI country managers from the other national companies
- The “council of the country lords” decide about the scope of the next SOA Backplane release
- If the decision is finalized, the implementation/test from “busy dwarfs” can start

## The busy dwarfs begin their work.

- The implementation and test phase starts after the final packaging from the “lords”
- The detailed design document is needed to verify on the one hand the completeness of the feature and on the other hand for the test department to define test cases



## The end of the feature request.

- If the test department approve the feature and the quality, a sign off document is written for the preparation of the launch of the new components
- We schedule three days for the update of a new SOA Backplane release to get all national companies to the latest version
- Four releases per year which cover round about 8 components (design time, runtime and provision time components)

## Table of Contents.





## Improvements.

- At the beginning of our “quest”, we weren’t aware of having a formalized feature request process. We implemented a feature request “on demand”.
- But with a more formalized process it helps to explain and to argument why feature XY is not in release YZ
- The risk is more ratable. Because before you start the implementation, you know exactly which components are affected
- Some feature getting cheaper, because you can collect several features for one component and you do a huge change instead of several small changes

## Lessons Learned.

- Projects dealing with technical infrastructures are easier to handle because the misconceptions in understanding the requirements are less
- A leaner requirements engineering process better matches an agile project



## Question before we start the journey – we will get the answer at the end of the journey.

- Do we need a heavy and complicate software engineering process or is it also with a lightweight and agile process feasible (with the background of the technical and organizational situation)
- Well known simple and efficient tools and straight forward processes?
- Or better heavy tool chain with a rough complex process?

## Answers to the questions...

- Did we need a heavy and complicate software engineering process or is it also with a lightweight and agile process feasible (with the background of the technical and organizational situation)

NO!

- Well known simple and efficient tools and straight forward processes?

YES!

- Or better heavy tool chain with a rough complex process?

NO!

**In a technical and agile environment you don't need a long running requirements process and software engineering process**

The END...



## Additional Information.

- oAW – openArchitectureWare : http://www.openarchitectureware.org
- C. Sensler, A. Karalus, SOA@T-Mobile – Vollautomatische Service Provisionierung auf dem ESB – Teil 1-3 in: JavaMagazin, 10.2008 – 12.2008, http://www.sensler.de/public.html
- C. Sensler, A. Karalus, M. Märtens, Ein Blick hinter die Kulissen - Modellrepository@T-Mobile, in JavaSPEKTRUM 1/2009

# Personal information.

- Dipl.-Ing. Carsten Sensler
  - Employee of T-Mobile Deutschland GmbH since April 2007 (but since December 2005 working in the SOA Backplane program )
  - Department of Enterprise Integration
  - System & Solution Designer
  - Functional leader of the international Service Provisioning Team
  - Contact: publication@sensler.de

- Dipl.-Inf. Andre Karalus
  - Freelancer,
  - Designer and developer of the runtime core component of the ESB in SOA Backplane and of the Core from the Service Repository



# Sources.

- http://de.wikipedia.org/w/index.php?title=Datei:Unicorn_n22_kerry.jpg&filetimestamp=20050220215709

- http://ardapedia.herr-der-ringe-film.de/index.php/Bild:Loth01.jpg