Sheer Deploying Pleasure
Shifting gear towards Agile delivery of mobility services

Patrick Verpaalen, Start-up Manager
Rens Voogt, IT Lead
# Sheer Deploying Pleasure

## Agenda

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>About Alphabet</td>
</tr>
<tr>
<td>2</td>
<td>New horizons</td>
</tr>
<tr>
<td>3</td>
<td>The road to Agility</td>
</tr>
<tr>
<td>4</td>
<td>New technology</td>
</tr>
<tr>
<td>5</td>
<td>Questions</td>
</tr>
</tbody>
</table>
Alphabet is BMW Group’s Business Mobility branch since 1997
It is part of the Financial Services Segment

Future-proof solutions for changing needs

Car financing & Fleet management

Car Sharing / Car-on-demand

AlphaCity
AlphaRent

eMobility

AlphaElectric
Alphabet is the No. 4 leasing provider in Europe
Global footprint is continuously enhanced

We serve 80,000 customers in 17 European countries, Australia and China with a team of 2,650 people.

We hold top 4 positions in 7 European countries.

We provide customised services, innovative solutions and manage multi-make fleets consisting of passenger cars and light commercial vehicles.
One Alphabet IT with two hubs: Breda and Munich
Looking for new talent

Breda Office, Netherlands
- 97 employees
- Internal Development Hub
- Dev and Ops together

Munich Office, Germany
- 69 employees
- Group IT Network
- Services & Steering

Solution Delivery incl. Application Operations

166 employees

Inhouse Development

> 56 active projects

CI/CD & DevOps

> 170 applications

Microservices Platform
Sheer Deploying Pleasure

Agenda

1. About Alphabet
2. New horizons
3. The road to Agility
4. New technology
5. Questions
A new international platform, built for change
Investing in the future

We break down our business in consistent, modular and flexible configurable business services.

Heterogeneous products, processes and monolithic systems with a defined set of functionalities

Comparable to a feature phone

Easy to use, any device

Easy to access, single identity

Easy to assemble, quickly to change

Easy to connect standard tooling

Easy to generate, consistent data

Easy to cooperate, market flexibility

Comparable to a smart phone
# Sheer Deploying Pleasure

**Agenda**

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>1</strong></td>
<td><strong>About Alphabet</strong></td>
</tr>
<tr>
<td><strong>2</strong></td>
<td><strong>New horizons</strong></td>
</tr>
<tr>
<td><strong>3</strong></td>
<td><strong>The road to Agility</strong></td>
</tr>
<tr>
<td><strong>4</strong></td>
<td><strong>New technology</strong></td>
</tr>
<tr>
<td><strong>5</strong></td>
<td><strong>Questions</strong></td>
</tr>
</tbody>
</table>
Our Challenge: Deliver in fixed time & budget frame

Our Answer: Transformation to 100% Agile
Why Agile
Embracing Change and Uncertainty

**Known project issues**

- Strong separation of Business and IT
- Create everything at one moment in time, try to foresee all future changes, in one big block
- Create the results sequentially, apply proofed approaches, failures are later expensive
- Ensure alignments by big all-deciding reviews

**How Agile can help**

- Bring business and IT closely together
- Create smaller chunks of the total solution to deliver quickly visible results
- Learn on the way by short term sprints and retrospective on process and results
- Ensure constant touch points between stakeholders to ensure quality deliverables, early

**Our Agile approach**

- Refine Parkinson's law: Work expands to fill the time available for its completion
- Change of requirements on the way difficult to inherit
- Constant review and adaption improves quality of results
We have adopted Large Scale Scrum
Using LeSS Framework: Autonomy, Mastery and Purpose

Key principles

- Deliver one common product
- Decrease number of handovers
- Reduce steering to a minimum
- Reduce barriers
- One Scope Management
- Strive for simplicity & easiness
- Ensure velocity
- Centralised planning

LeSS Principles
Current program setup
Line following in 2018

Start 2015
100% SDM / Waterfall
Three releases per year
Oracle PL/SQL
A handful of monolithic systems
Fixed desks
Outsourced

Target 2020
100% Agile
Multiple releases per day
Java / JavaScript
A swarm of microservices
Flex desks
In-house

Development vs Operations
Traditional infrastructure
• One Program Backlog
• Four Area Product Owners
• Five feature teams, one enabling team
• All knowledge and disciplines part of the team
• Strong discipline on Agile ceremonies
• Co-location of developers (and other stakeholders as much as possible)

=> Ready for go-live Nov 2017 – 18 months BEFORE planning
# Sheer Deploying Pleasure

## Agenda

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>About Alphabet</td>
</tr>
<tr>
<td>2</td>
<td>New horizons</td>
</tr>
<tr>
<td>3</td>
<td>The road to Agility</td>
</tr>
<tr>
<td>4</td>
<td>New technology</td>
</tr>
<tr>
<td>5</td>
<td>Questions</td>
</tr>
</tbody>
</table>
History of BMW Group IT & Alphabet IT
Plan-Build-Run-Silos

Agile working in a waterfall environment! A key challenge!
New way of working at BMW and Alphabet IT
You build it, you run it

- From waterfall projects and outsourcing
- To agile approach and insourcing
- Acknowledge competence gap: invest in new competencies
- Invest in technology
- No silos in the organization (go DevOps)
## Alphabet IT

**How can we plan IT as our plants**

- Make work visible
- Make work plannable (time and budget)
- Limit the amount of work in progress
- Identify events
- Automate as much as possible
- Eliminate waste
- DevOps collaboration
- Management commitment to change
## Delivery Pipeline in action

### Average stage times:

<table>
<thead>
<tr>
<th>Stage</th>
<th>Checkout</th>
<th>Build and Unit Test</th>
<th>SonarQube</th>
<th>Publish</th>
<th>Component Test</th>
<th>Create Docker image</th>
<th>Publish Docker image</th>
<th>Deploy DEV</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10s</td>
<td>2min 17s</td>
<td>24s</td>
<td>31s</td>
<td>5min 1s</td>
<td>14s</td>
<td>3s</td>
<td>5s</td>
</tr>
<tr>
<td>2</td>
<td>8s</td>
<td>2min 28s</td>
<td>25s</td>
<td>31s</td>
<td>3min 52s</td>
<td>18s</td>
<td>3s</td>
<td>5s</td>
</tr>
<tr>
<td>2</td>
<td>867ms</td>
<td>1min 30s</td>
<td>23s</td>
<td>34s</td>
<td>8min 23s</td>
<td>40s</td>
<td>3s</td>
<td>5s</td>
</tr>
<tr>
<td>3</td>
<td>1s</td>
<td>1min 26s</td>
<td>22s</td>
<td>33s</td>
<td><strong>9min 23s</strong></td>
<td><strong>Failed</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>8s</td>
<td>2min 59s</td>
<td>22s</td>
<td>36s</td>
<td><strong>8min 55s</strong></td>
<td><strong>Failed</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>8s</td>
<td>2min 42s</td>
<td>25s</td>
<td>32s</td>
<td><strong>8min 18s</strong></td>
<td>11s</td>
<td>3s</td>
<td>4s</td>
</tr>
</tbody>
</table>

*Notes:*
- Red indicates failed stages.
- Green indicates successful stages.
Need for OpenShift
Business case

Old infrastructure

• 220 web rooms with multiple instances needed
• Complex to manage (operate)
• Guarantee that each instance is the same is difficult
• Lead time within BMW (72h; if available)

Why OpenShift?

• Automated service discovery
• Fully flexible to scale out
• Fully flexible to create environments on the fly
• Same deployments to different environments
• Ability to rollback to a previous version of your application
Scaling a software landscape

**Scale up**

```
[ ] → [ ] → [ ]
```

**Scale out**

```
[ ] + [ ] + [ ] + [ ]
```

![Graph showing scale up and scale out strategies](image)

- **Capacity**
- **Performance**

- **Physical Limit**
How are we dealing with parallel development?

Code Branching Strategy

feature-y

feature-x

master

1.0.0  

1.1.0  

1.2.0  

Tag indicates version that successfully passed pipeline.

Catch up with master

Red Hat Forum Breda 2017

16/10/17
Setup of OpenShift
Cluster and environments

Current setup of OpenShift

- Enterprise version v3.5.5.31.24
- 2 OpenShift clusters (Dev and PROD)
- 280 GB memory
- 168 CPU cores
- 6 development teams
- 22 environments

Outside of OpenShift

- External PostgreSQL databases for persistent data (per service)
- Elastic Stack
  - FluentD
  - ElasticSearch
  - Kibana
- Prometheus for environment availability and endpoint monitoring
Monitoring and Logging of OpenShift environments
Lessons learned
Go full throttle

How we started
• First provided OpenShift solution (3.2)
• Not flexible in allocating resources
• Not flexible in creating projects
• Provided by external company
• Operated by external company
• Minimal training
• No logging in place

Current state
• OpenShift Version 3.5.5
• Logging in place (external ElasticStack instance)
• Full flexible in creating OpenShift projects (environments)
• OpenShift developers followed the Red Hat developers training
• Full flexible for scaling
• Persistent data

Key take aways
• 100% commitment from all involved
• Steep learning curve of OpenShift
• Command lines are extensive, but very useful
• Red Hat OpenShift Enterprise training will provide Return on Investment
Sheer Deploying Pleasure

Agenda

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>About Alphabet</td>
</tr>
<tr>
<td>2</td>
<td>New horizons</td>
</tr>
<tr>
<td>3</td>
<td>The road to Agility</td>
</tr>
<tr>
<td>4</td>
<td>New technology</td>
</tr>
<tr>
<td>5</td>
<td>Questions</td>
</tr>
</tbody>
</table>

Red Hat Forum Breda 2017
Questions?

https://www.alphabet.com/nl-nl/vacancies/corporate-division/it
Alphabet Business Mobility Services
Keep Moving