Specifying ‘Wow!’ at Elektrobit

Case-2014-03-Elektrobit-Specifying-Wow

Thomas Fleischmann, a product manager at Elektrobit Corporation—an engineering company focusing on wireless and automotive technologies—was facing a difficult situation. He was in charge of the development of one of the company’s main products, *EB GUIDE*, which serves as a toolkit for generating and operating car infotainment user interfaces.

1 Elektrobit’s EB GUIDE

1.1 The Need for ‘Wow!’

It was April 2012 and Fleischmann had just received emails from sales staff from all over the world. Elektrobit had been demoing the latest release of EB GUIDE to customers.

In addition, he had just received a Request for Quotation (RFQ), which provided new product feature requests. He had to realize that customers requested features that were well beyond the original specification. No specific shortcomings were mentioned but the general feeling was that the demo showcasing the features of EB GUIDE did not meet expectations. It seemed that EB GUIDE in its current form failed to convince customers. In a questionnaire labeled “Needs”, Fleischmann’s eyes repeatedly caught a phrase ranked at the top of the list:

> *“Need Wow Demo!”* - Japan, *“More Wow demo”* - USA, *“Demo stuff”* - Korea.

The demo lacked a ‘Wow!’ factor and Fleischmann realized that they had to work on a fresh approach. At this point, EB GUIDE had been a part of the market for ten years successfully. It was crucial for Fleischmann to ensure that the product maintained its place in the competitive market. Thus, new features had to be created.

> Various ideas ran through his head: “What can we do? What will amaze people? What will customers want in the future?”

> More practical questions were also on his mind: “When can we implement new features and where do they fit into an already packed release schedule? Can I even justify pulling staff off their current projects for this?”

Fleischmann recognized that the solution to these issues was closer than he had first thought. His mind turned to integrating “multi-touch” into EB GUIDE, a feature he had been thinking about for some time. Now seemed like the ideal opportunity to get it done.

1.2 Elektrobit Overview

1.2.1 Founding and Early Days

The Elektrobit Group traces its history back to the merger of JOT Automation Group Oyj and Elektrobit Oyj in 2002. The actual history goes back further to the foundation of JOT Automation, Finland, in 1985. JOT Automation had been one of the leading provider of services in developing telecommunication electronics. Prior to the merger, Elektrobit Oyj had been working towards establishing new grounds in high-frequency engineering.

Over time, there had been several important additions to the group. An example is 3SOFT in 2004—a company based in Erlangen, Germany—which specialized in designing controller module software. 3SOFT had been founded in 1988 and had just gone through a transformation away from offering services in the medical engineering sector towards providing services the automotive sector.

After 3SOFT, the Austrian company DECOMSYS and the French corporation NCS had been added to the automotive part of Elektrobit Corporation. Acquisitions like these were deliberate. Elektrobit’s goal was to enter the automotive software market and they succeeded well by way of the described take-overs.

Elektrobit’s mission, in their own words, is to enrich people’s lives through innovative technologies, products and solutions as well as offering value creating solutions in the automotive and wireless environment (Elektrobit, 2013).

1.2.2 Elektrobit in Numbers

In 2013, Elektrobit and its subsidiaries were employing 1,648 employees. The annual turnover of Elektrobit amounted to €199.3 million. This reflected a growth of 14.6 per cent when compared with sales in 2012 (Elektrobit, 2013). The company has gotten rid of most business areas unrelated to automotive sector like location tracking or network testing while keeping the automotive and wireless business segments active.

According to the annual report of Elektrobit, demand for automotive products grew well in 2013. This resulted in the majority of the corporation’s net sales being rendered in the automotive sector. Figure 1 presents the net sales over the years from 2011 to 2013 with about €110.6 million gained in the automotive segment in 2012 and €138.3 million in 2013. In its early years, Elektrobit’s automotive department’s revenue amounted to €27.1 million in 2005 (Elektrobit, 2005).

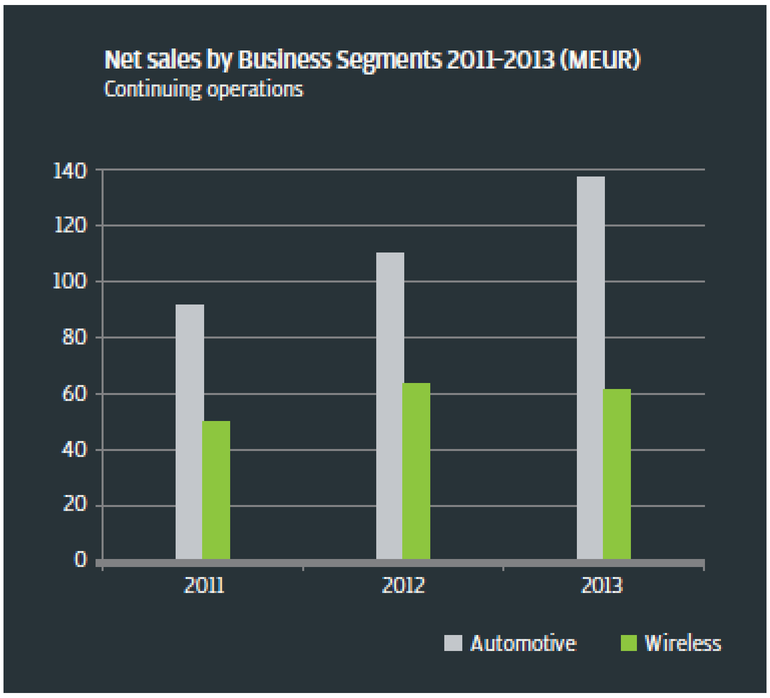


Figure 1: Elektrobit's net sales from 2011 to 2013 (Elektrobit, 2013)

1.2.3 Products and services

The products of the automotive department of Elektrobit consist of services and solutions for several application areas. While some are self-explanatory, others need brief outlining. These include *Navigation*, *Assisted-driving*, ***AUT****omotive* ***O****pen* ***S****ystem* ***AR****chitecture* (AUTOSAR), *FlexRay, Infotainment* and *Human-Machine Interfaces* (HMI).

AUTOSAR was created by car manufacturers and suppliers as an open software architecture standard to facilitate access to control devices. FlexRay applications use the FlexRay automotive communication protocol to provide a network with fast and consistent data rates throughout the vehicle as a basis for advanced equipment to operate reliably. Infotainment provides information and entertainment services whereas HMI acts as a digital user interface in modern car systems.

These technologies resulted in the following product line offerings:

- *EB street director* is a versatile software platform for navigation with customization features for entry level up to premium systems.
- *EB tresos* provides a scalable basic software, operating system and a tailor-made tool environment to develop and configure the Electronic Control Unit (ECU).
- *EB Assist ADTF* (Automotive Data and Time-triggered Framework) includes an extensive product line with flexible and extensible tools for the design of assisted-driving solutions as well as an independent software development kit.
- *EB GUIDE* is an all-in-one HMI creation and speech dialog platform with capabilities for specification, modeling, code generation and production.
- *Engineering services* are comprehensive services concerning software development for infotainment, driver assistance and ECU in the automotive industry.

1.3 The Product EB GUIDE

1.3.1 Involved People

Thomas Fleischmann had been involved in the EB GUIDE project from its inception. After finishing his studies in computer science in 2004, he had started working at 3SOFT. At that time, the firm was engaged in product development services, embedded software systems for automotive, automation and medical electronics. 3SOFT had its headquarters in Erlangen and was later acquited by and integrated into the Elektrobit Group (Elektrobit, 2005). Fleischmann spent two years in Elektrobit's software development and went on to collaborate in creating and maintaining the initial business plan for EB GUIDE. As a product manager, he mediated product development, distribution and customer relations until January 2014.

1.3.2 The Product

EB GUIDE's two main components are *EB GUIDE Studio* and *EB GUIDE Graphics Target Framework* (GTF). While the former handles the construction, implementation, and customization of user interfaces, the latter acts as a virtual machine in the target system that executes the HMI. EB GUIDE Studio contains a runtime environment to simulate the HMI without having to install it into the GTF beforehand.

> Fleischman explained it this way: "Imagine cooking dinner for some friends. In this case, the kitchen resembles EB GUIDE Studio, the dining room mirrors the GTF and the dinner is your HMI. Ideally you will proceed to cook your meal in the kitchen. It comes with the option to check the taste of the food before serving it, just like Studio enables developers to test HMIs without having to run it on hardware with the GTF. Furthermore you might not be able to fit all your friends inside your kitchen. The dining room on the other hand, suits this purpose optimally. Studio is a powerful tool with high performance demands and is therefore not meant to be executed on embedded hardware, but the specifications of GTF are tailor-made to work with embedded hardware."

The componentization shown in Figure 2. The idea is that Elektrobit provides a program so that Original Equipment Manufacturers (OEM) can design interfaces conveniently. This includes drafting the work flow and the appearance with full control and freedom.

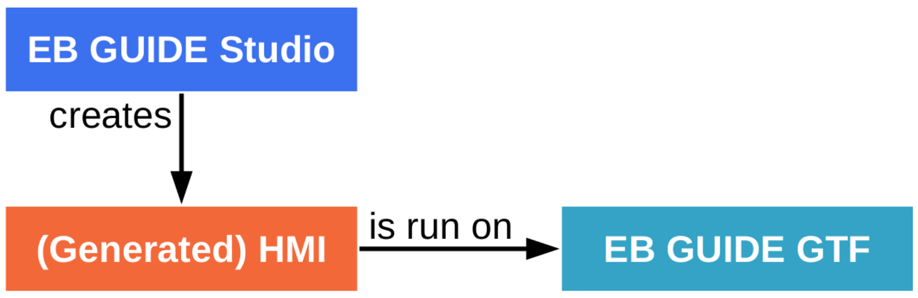


Figure 2: EB GUIDE's basic structure.

EB GUIDE addresses two types of automotive infotainment interfaces. As shown in Figure 3, one interface is the *cluster instrument* which is located right behind the steering wheel where most instruments have been present since the commercialization of the automobile. The other interface is the *head unit* that is placed in the central console, replacing the DIN slot of old car radio systems. Both types come with unique preconditions and technical characteristics due to their requirements and utilization and run on a diverse number of platforms including Linux, QNX (a Unix-like operating system for embedded systems), Android, Windows 7 and CE 7.

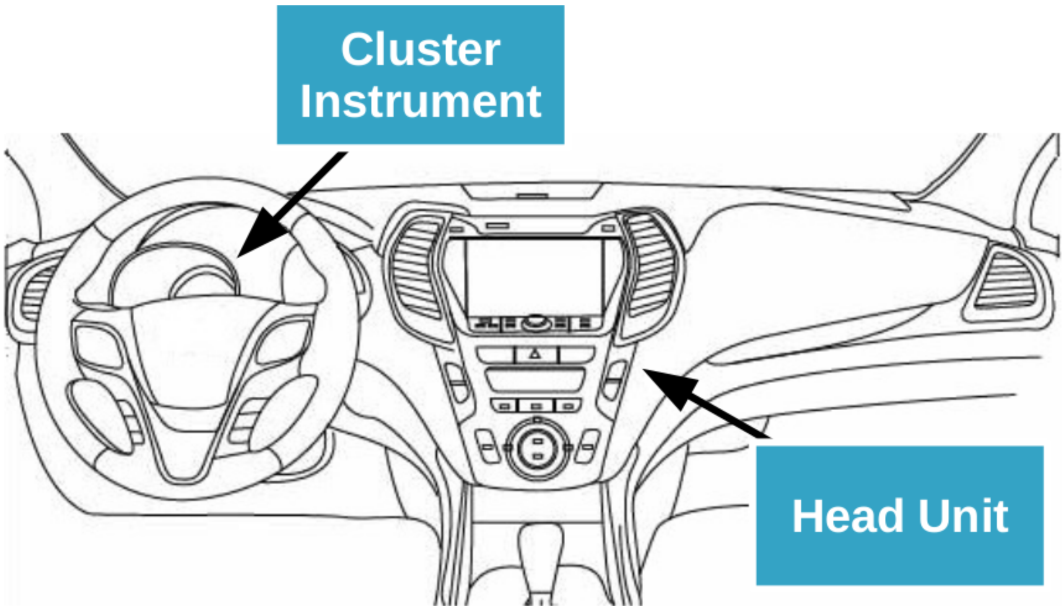


Figure 3: Location of cluster instrument and head unit in modern cars (Freedesignfile.com)

After a car manufacturer designed and approved an interface, the next step is to get in contact with a supplier to work out the transfer of the runtime environment to a suitable hardware to ensure smooth performance for the end-user.

To get an idea of what the resulting HMI of EB GUIDE looks like, Exhibit 2 shows a human-machine interface created for a cluster instrument. All elements displayed are digital, including the speedometer needle and can be designed as desired. For instance, the compass in the upper half is optional and could be exchanged for other indicators. The car model in the center can be animated to hint at any problems the vehicle might have (e.g. a broken light, an empty battery or the trunk not being closed properly).

EB GUIDE generates revenue for Elektrobit in the form of licenses for the toolkit and for the systems that run in the cars built by the OEMs. The revenue varies depending on the type of solution chosen. Cluster instruments use less expensive systems and as a result use simpler interfaces. As a consequence, a license for a head unit typically cost 5-10 times more than a license for a cluster element, depending on the complexity of the interface. Elektrobit calls this quota licensing.

Elektrobit is not the only vendor to provide this kind of software. A major competitor is the Qt Project. Initially released in 1995, it provides libraries to Graphical User Interface (GUI) and comes with an open source license. This enables users not only to use the program, but also to modify it freely (Qt Project, 2013).

1.3.3 Automotive technologies in 2012

By 2012, automotive infotainment systems had long reached a solid position with a total estimated revenue of US$34.6 billion (IHS, 2013). Most of these earnings were generated by the Top 10 suppliers in the business, accounting for 65 per cent of the total revenue. Panasonic was said to have gained US$4 billion solely from their automotive infotainment division that year, ensuring them a top spot at 12.1 per cent of the market share.

The global market painted a picture of steady growth with an increment of 3 per cent from 2010 to 2011 and 6.4 per cent from 2011 to 2012. Market research studies predicted sales to reach US$41.2 billion by 2016 as a result of advances in information technology and system replacements in older cars (IHS, 2013).

Automotive infotainment is a crucial pillar of modern car design. Ten years ago, car builders and suppliers had still been referring to the driver seat as the “driver work station”. By 2012, the association to work had long faded. The term “driving experience” had emerged to capture a new, exciting feeling of driving and associated infotainment technologies.

A closer look at the background of the situation reveals that the development of touchscreens was important. The first of their kind were built in the 1970s. The first phone with a touchscreen was produced in 1993 (Holzinger, 2003). Even though this technology has been around for quite some time, it has been making its way into modern day innovation only slowly.

Technology companies began to develop applications with multi-touch capabilities around the year 2000. Multi-touch gained a lot of attention with the release of Apple’s iPhone in 2007. Attention grew even further with the entry of Android into the mobile phone market.

Brands need to differentiate themselves from their competitors all the time. Back in the day, car manufacturers had little means to do so. One way was through car radio systems that could be designed to suit the quality and style of the brand’s identity. These radios could be replaced rather easily since the standardization of the slot as DIN ISO 7736 and its wide adoption across the industry.

However, over time, even lower priced automobiles migrated from DIN-Slots to infotainment systems. These systems, along with the underlying software and interface offer a unique experience to strengthen corporate and branding identities. Not only is the specific design of the digital interface constructed as desired, the diversity of specific technologies helps to provide a unique feel when driving a car.

> As Thomas Fleischmann said: “Audi builds HMIs specifically so that the A4-cockpit resembles the one in the A6-series. A Mini Cockpit, however, feels different from a BMW’s 5-series’ one, although both interfaces are designed by BMW.”

Thus, car manufacturers try to brand not only the individual car but the whole product line in the hopes of making customers return in the future when they are buying their next car.

As an example of infotainment system integration, Audi offered its A3 model with an optional Multi-Media Interface (MMI) touch terminal system in 2012. This is not a multi-touch display but it allowed touch controls on capacitive buttons (Figure 4, left picture) to be relayed to the head unit (Figure 4, right picture). Until then, these kinds of controls had been available only in top-end vehicles. They made it easy to control the interface since less time was spent on reaching the head unit and hence reduced the driver distraction.

Even without an additional upgrade, it is common to have some sort of infotainment system installed in newly bought cars, even in the lower price segments of more basic car manufacturers. Every digital user interface has a purposely built HMI underneath.

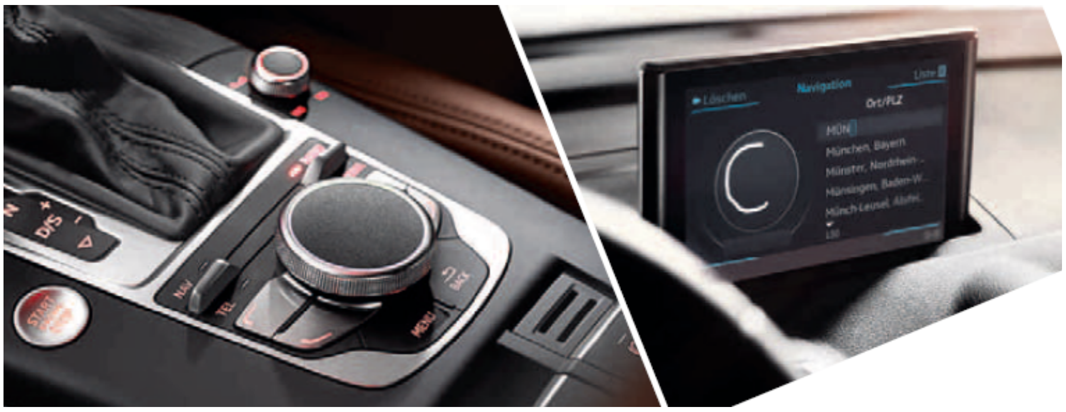


Figure 4: The cockpit of the Audi A3, product catalog of 2012 (Audi, 2012)

The desire to differentiate drove manufacturers to go for more connectivity in their cars. The research firm ABI predicted that 60 per cent of all cars will have Internet and mobile phone integration by 2017. Ford had been shipping its *AppLink* feature since October 2012. It is capable of making the car's buttons on the steering wheel as well as the radio and voice commands available to all popular mobile phones' operating systems (Schneiderman, 2013).

Advanced touchscreens even allow for multiple, simultaneous inputs (referred to as "multi-touch") to be interpreted by the software as complex gestures. These inputs function as shortcuts to avoid inconvenient multi-step input procedures. Although multi-touch functionality had been in wide use by 2012, especially in mobile phones and tablet computers, no car manufacturer or infotainment supplier offered path-gesture recognition capabilities as part of their infotainment software.

2 The Multi-touch Feature

2.1 Getting Started

Thomas Fleischmann sat in his office and began to sort out his thoughts. It was clear to him that the buzz missing in the demo could be achieved by a feature he had held back to focus on other requirements. Multi-touch had been entering the industry and Elektrobit had to include it eventually in its products.

There were other features that could help getting the customers excited, like visual effects in the HMI. Splash effects on touch and 3D views fell into that category. But Fleischmann decided that at this point multi-touch made more sense. Visual effects were already available in competitors' products. "Me too" would not be enough. He had to come up with "Me too plus X" where 'X' would be gestures.

The need for a bigger 'Wow!' could be satisfied by introducing multi-touch and gestures to EB GUIDE's capabilities in HMI design. This would include widely known input gestures like pinch to zoom pictures or maps as well as other multi-touch instructions to access functions. This would make EB GUIDE more attractive.

To get an idea of what technologies were available and what kind of choices were possible, he instructed his team to come up with concepts that would make the feature become a reality.

2.2 Realization Challenges

Fleischmann and his team had their work cut out for them. With a vague idea of what the feature should be, they had to come up with a solid description to ensure smooth implementation.

> “Just what to consider?” Fleischmann asked himself.

First off, it was clear to him that multi-touch could only be realized on head units. Inputs with multi-touch and gestures cannot be realized on a cluster instrument located behind the steering wheel. The cluster instrument would also be out of reach for a co-driver.

Multi-touch and path-gestures required non-trivial hardware. Not all kinds of displays allow for multiple inputs. Resistive touchscreens only detected up to two touches simultaneously. They are less precise and wear out quicker because they usually need to be pressed harder.

Capacitive displays work on a different principle. They detect alterations in conductivity (the ability to transmit electrical current) on the screen. Hence, they are much more precise and allow for multiple inputs at the same time with only a slight increase in cost.

To properly run HMI with multi-touch and gesture support, the hardware has to provide the necessary performance. In their implementation, state machines, a concept from computer programming, were needed. These state machines consumed more memory and more processing power, so that the hardware would be difficult to place in the limited space provided by the cluster instrument. This is another reason to use the head unit since they it already was able to render navigation information at high definitions, handle Internet streams and multimedia playback concurrently. The cluster instruments were not capable of any of that.

The fact that the cluster instruments would not be able to include the new feature was important at Elektrobit, because sales of cluster instruments were significantly higher than those of the much more expensive head units. Head units accounted for only a tenth of the number of cluster instruments sold. This difference can be attributed to the significantly higher cost of the head unit which cannot be justified for a new car with a small budget.

Additionally, many end-users who rarely use navigation simply do not need this kind of high-end application to get directions and may just use an equivalent software on their phones. Only high-end cars that are equipped with premium infotainment systems.

	OEM Display Growth Opportunity for 2011-2019		
	Estimated revenue in million dollars (USD)		CAGR in %
	2011	2019	
Head unit	23.3	81.7	17
Cluster instrument	4.4	56	37.4

Table 1: Estimated revenue growth for the infotainment opportunity

The numbers in Table 1 reassured Fleischmann in his endeavor to go for multi-touch. The Compound Annual Growth Rates (CAGR) projected for the years 2011-2019 predicted cluster instruments to grow at 37.4 per cent a year. Although the growth rates for head units pale in comparison, they were still promising and a safe way of approaching the product's future.

Another important decision was the method of path-gesture generation. Since Elektobit did not construct the HMIs themselves, car manufacturers would need additional tools to design and handle these gestures. Either EB GUIDE needed another tool where gestures could be determined from scratch or it had to incorporate a collection of preset gestures. Fleischmann identified that offering another tool came along with unpredictable incremental costs in product support and maintenance. On the other hand, there might be demand by some customers for more gestures than the preset ones.

Elektrobit was already building software for touchscreens and hence Fleischmann and his team had to keep in mind that implementing multiple inputs and path-gestures would have to be convenient to customers. Inputs had to be recognized with minor failure rates but had to be processed in due time so that the controls still felt pleasant and responsive. Additionally, these gestures should not interfere with single touch operations.

In human-machine interaction, comfort and satisfaction of usage are important to end-user happiness. Capacitive touchscreens allow for response times of less than 20 milliseconds which is lower than the receptiveness of the human brain and thus does not enhance the user experience (Holzinger, 2003). But there is a measurable correlation between user happiness and the response time of touchscreen events. A factor to consider is the potential distraction of the driver. 750 milliseconds is regarded as the acceptable time window for a driver to disregard traffic in order to control a touchscreen (Reisinger, 2014). However, the response times should not exceed 200-250 milliseconds in order to avoid negative user experiences. Path-gesture recognition should have a rather high success rate, otherwise the customers might neglect using gestures altogether. Good recognition software can achieve success rates as high as 93 per cent (Lee, 1999).

Finally, any solution would only make sense if it was sufficiently cheap and hence could be expected to be profitable. This constrained the design space for multi-touch further.

2.3 Engineering Process

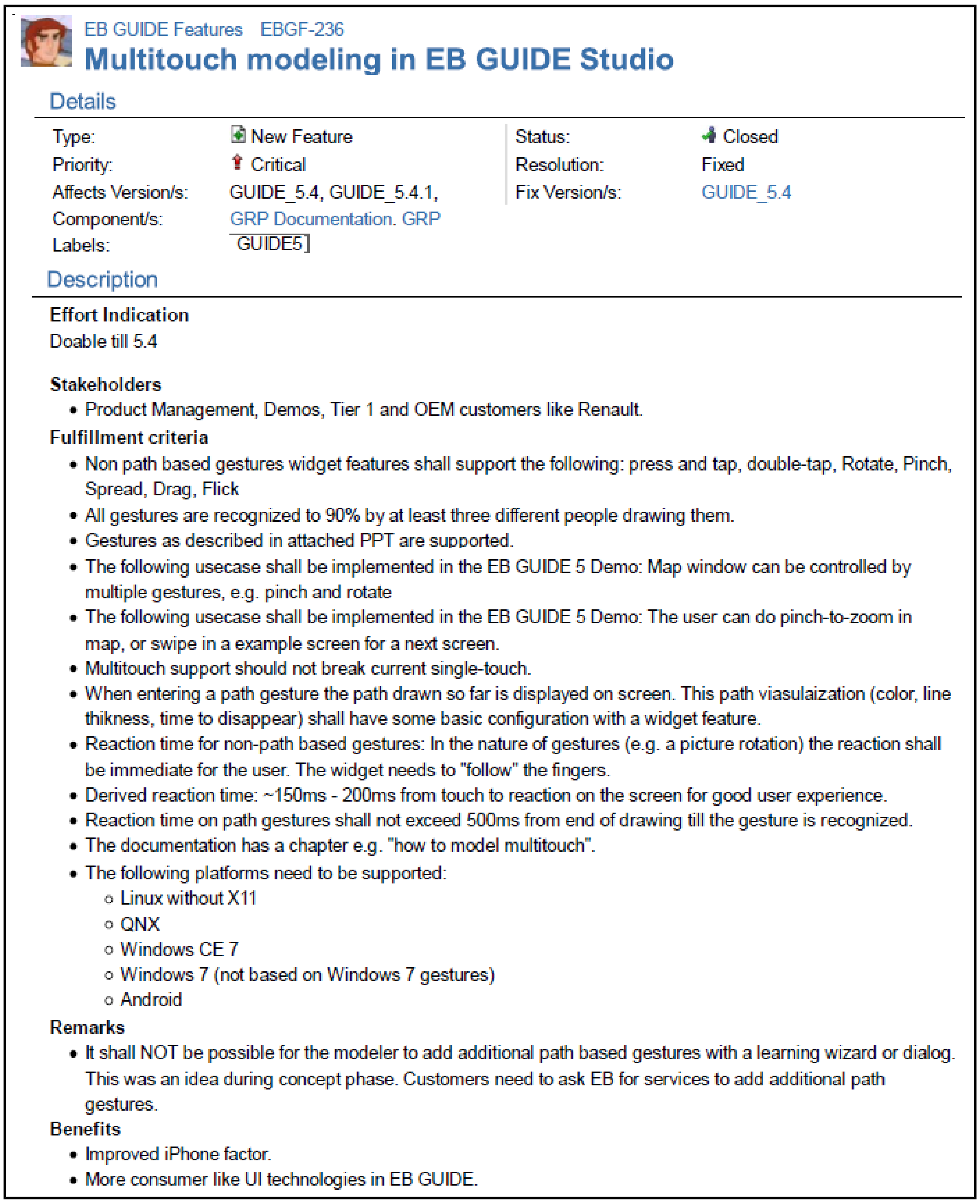
Development of EB GUIDE was performed using Scrum. Scrum is an agile software development methodology with an iterative and incremental development strategy. Work is split into separate packages to be processed in a sequence of defined time-boxes of a few weeks called sprints. At the end of each sprint, there was (ideally) a functional release of the software under development. Fleischmann, fulfilling the role of the product owner, defined the functionality of multi-touch and gesture recognition. He captured this functionality as a Scrum “epic”, that is, a high-level feature.

The multi-touch epic outlined all key facts, including the following:

- All basic multi-touch inputs known from mobile phones shall be available to control the interface screen elements.
- At least 90 per cent of all gestures given by three different people shall be correctly recognized.
- There shall be a preset collection of gestures available without a tool to create them.
- The feature shall support a set of platforms (defined in the epic).
- Multi-touch inputs shall not interfere with single touch operations.

Figure 5 provides a screenshot of the epic as Fleischmann had entered it into Jira, a web based issue and bug tracking product with project management features.

Exhibit 3 shows some gestures that had been included in the epic created by Thomas Fleischmann.


EB GUIDE Features EBGF-236

Multitouch modeling in EB GUIDE Studio

Details

Type:	New Feature	Status:	Closed
Priority:	Critical	Resolution:	Fixed
Affects Version/s:	GUIDE_5.4, GUIDE_5.4.1,	Fix Version/s:	GUIDE_5.4
Component/s:	GRP Documentation, GRP		
Labels:	GUIDE5		

Description

Effort Indication
Doable till 5.4

Stakeholders
- Product Management, Demos, Tier 1 and OEM customers like Renault.

Fulfillment criteria
- Non path based gestures widget features shall support the following: press and tap, double-tap, Rotate, Pinch, Spread, Drag, Flick
- All gestures are recognized to 90% by at least three different people drawing them.
- Gestures as described in attached PPT are supported.
- The following usecase shall be implemented in the EB GUIDE 5 Demo: Map window can be controlled by multiple gestures, e.g. pinch and rotate
- The following usecase shall be implemented in the EB GUIDE 5 Demo: The user can do pinch-to-zoom in map, or swipe in a example screen for a next screen.
- Multitouch support should not break current single-touch.
- When entering a path gesture the path drawn so far is displayed on screen. This path viasulaization (color, line thikness, time to disappear) shall have some basic configuration with a widget feature.
- Reaction time for non-path based gestures: In the nature of gestures (e.g. a picture rotation) the reaction shall be immediate for the user. The widget needs to "follow" the fingers.
- Derived reaction time: ~150ms - 200ms from touch to reaction on the screen for good user experience.
- Reaction time on path gestures shall not exceed 500ms from end of drawing till the gesture is recognized.
- The documentation has a chapter e.g. "how to model multitouch".
- The following platforms need to be supported:
  - Linux without X11
  - QNX
  - Windows CE 7
  - Windows 7 (not based on Windows 7 gestures)
  - Android

Remarks
- It shall NOT be possible for the modeler to add additional path based gestures with a learning wizard or dialog. This was an idea during concept phase. Customers need to ask EB for services to add additional path gestures.

Benefits
- Improved iPhone factor.
- More consumer like UI technologies in EB GUIDE.

Figure 5: Jira document of the resulting feature definition as a Scrum epic (Elektrobit, 2013)

2.4 Scheduling the Feature

The freshly defined feature had to be added to the product roadmap of EB GUIDE. Fleischmann had to decide on how to incorporate multi-touch to the release plan for 2012. Figure 6 shows that the release plan already contains complex and time-intensive tasks.

By the time the feedback to the dissatisfying demo had reached Fleischmann, the program version 5.2.1 of EB GUIDE had already been released. He now had to evaluate the importance of prioritizing the release of a feature that might excite customers over implementation of pending tasks like integrating OpenVG (a framework to enhance graphic processing performance on embedded systems), a Simulink extension (a data flow programming language used for digital signal processing) or the new version of HyperText Markup Language (HTML5) with improved multimedia support.

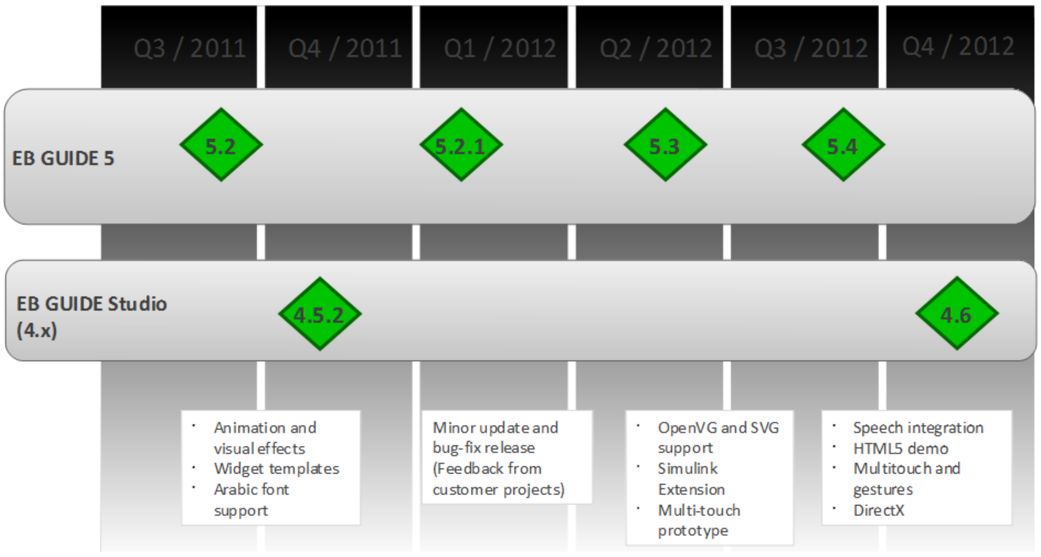


Figure 6: Product roadmap for EB GUIDE 2012 (Elektrobit, 2012)

Fleischmann was convinced that exciting customers had to be prioritized over incremental features. Improvements to the technological factors in EB GUIDE would have to wait. Losing customers due to an underwhelming demo could not be tolerated and the fact that multi-touch and specifically path-gestures were a new factor in automotive infotainment systems seemed to support this decision. Therefore, the first functionality of multi-touch was scheduled for EB GUIDE's 5.3 release and to be fully developed by and incorporated into version 5.4.

Fleischmann added the multi-touch epic to the product backlog. The product backlog contained all the epic feature descriptions that were still missing (were "backlogged") in the product. The backlog sorted these descriptions in order of descending priority. At the beginning of the next sprint session, the planning meeting was supposed to define which functionality was to be implemented next.

2.5 Finding time for 'Wow!'

At Elektrobit's development department, the usual sprint length was two weeks. After Fleischmann decided that the multi-touch feature would be a part of the next product release, his team began to break down the multi-touch epic into smaller user stories, each of which addressed a particular aspect of the epic.

EB GUIDE is a tool with customers which are different from end-users. Thus, the engineering team had to pay significant attention to the usability of the pending multi-touch feature implementation.

> An example of a user story is: "As a designer, I want path-gestures to be identified and referenced by a unique attribute so that scripting the gesture input events can be processed easily."

This example user story would be relevant to customers (designers) but not end-users (drivers). However, there were also user stories that affected both roles:

> "As a driver I want a multi-touch pinch to zoom the navigation map so that the behavior of multi-touch does not differ from my mobile phone."

After getting the feature planned and into development, Fleischmann did not participate further in the creation or adjustment of user stories. At Elektrobit, his job as a product manager was to prioritize epics and user stories and oversee the entire process. Therefore, sprint planning meetings were used only to tell the team which goals should be achieved within the next sprint. Fleischmann did not have to personally attend the meeting as the product backlog already contained those goals in the form of user stories he wished to be realized.

The other important event was the change control board meeting. Unlike the sprint planning meeting, which discussed user stories that had yet to be implemented, the change control board meeting discussed only user stories that had already been implemented. In this meeting, Fleischmann and other product managers balanced the importance of features not yet implemented with requests to improve features that had already been implemented. This event was scheduled once every sprint.

2.5.1 Elements of a feature

As soon as Fleischmann and the team broke down the multi-touch epic into smaller user stories, they had to get them into a reasonable order. Since the whole implementation would be partitioned into multiple sprint increments, Fleischmann had to match all related stories to one sprint backlog. He also had to keep an eye on the correct functional order of the feature realization, the dependencies of the stories and had to be cautious to neither overload the capacity of the team nor demand too little from them. Exhibit 4 showed the list of initial user stories. They still had to be broken down further and prioritized properly.

3 Challenges Ahead

As clean and simple as the process might sound, it was not all a smooth sailing at Elektrobit. After receiving inputs from the concept request phase, Fleischmann was certain that it would not take too much effort to bring multi-touch to EB GUIDE. However, gesture recognition would require a lot more work. To correctly detect a gesture pattern, the software had to determine if the current input was a gesture and detect how and when the gesture started and where it ended. This complex behavior needed its own computation engine and since gestures should be acceptable on every screen, this engine had to be integrated into each one of them.

> Consider the following user story: "As an end user, if I draw a roof gesture, the navigation system should promptly calculate a route to my saved home address."

Thus, regardless of what state the infotainment system was in, the “roof” gesture had to be recognized and processed. At first sight, this user story seemed harmless. In fact, it was initially prioritized as less urgent, because it was considered easy to implement.

There had already been a few sprints to implement the architecture for the pattern recognition engine. To handle the “roof” gesture input on any screen, the team positioned several invisible layers on top of all screens to detect them. The resulting problem was that no commands were able to reach the underlying layers, effectively jamming the whole system.

Therefore, in order to implement this gesture, the software architecture had to be adapted significantly. This required more time, but since EB GUIDE’s 5.3 release was already close and could not be pushed back, the incorporation of this feature into the upcoming release delayed. This forced Fleischmann to greenlight the release of a flawed version of EB GUIDE in which multi-touch had been disabled in order to not have the the invisible (multi-touch) layer block more basic touch events.

As the corresponding change to architecture and processing, the first step of every pattern detection was to determine if the input was just a simple touch. If so, complex detection was immediately terminated and the instruction was passed on to the bottom layer. This fix was implemented in the next release.

List of Abbreviations

AUTOSAR	AUTomotive Open System ARchitecture
ECU	Electronic Control Unit
GTF	Graphics Target Framework
GUI	Graphical User Interface
HMI	Human-Machine Interfaces
OEM	Original Equipment Manufacturers

References

Elektrobit (2005). Annual Report 2005. Available from http://www.elektrobit.com/download/2160/annual_report_2005/pdf (Accessed September 2, 2014)

Elektrobit (2013). Annual Report 2013. Available from https://www.elektrobit.com/file.php?id=4337 (Accessed May 11, 2014)

Holzinger, A. (2003). Finger instead of mouse: touch screens as a means of enhancing universal access. In *Universal access theoretical perspectives, practice, and experience* (pp. 387-397). Springer Berlin Heidelberg.

IHS, Inc. (2014, August 6). Panasonic Tops List of Auto Infotainment Suppliers in 2012. Available from http://press.ihs.com/press-release/design-supply-chain-media/panasonic-tops-list-auto-infotainment-suppliers-2012

Lee, H. K., & Kim, J. H. (1999). An HMM-based threshold model approach for gesture recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on, 21*(10), 961-973. Available from http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=799904 (Accessed April 11, 2014)

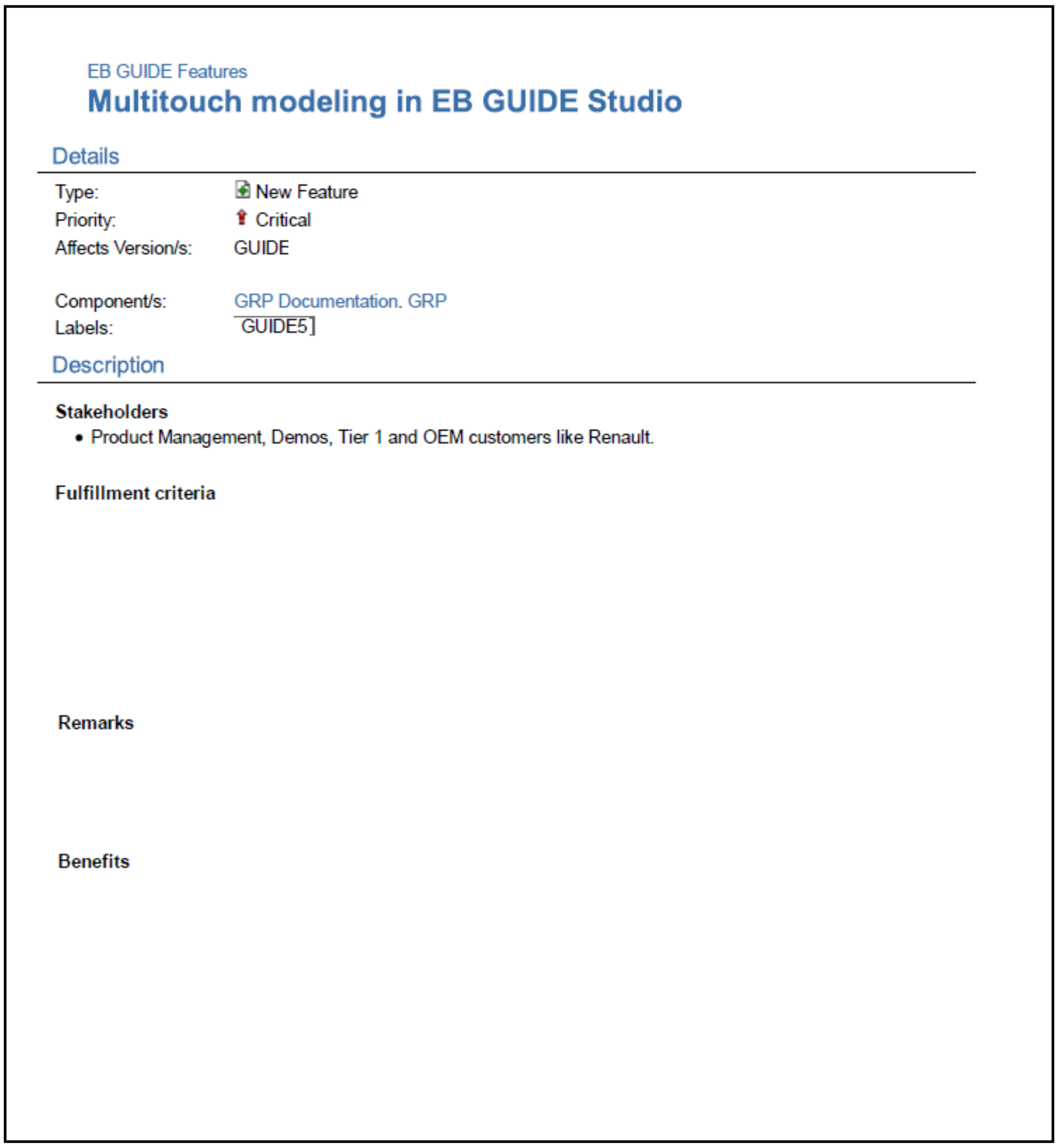
Reisinger, C. (2014). Interaktion mit Touchscreens im Fahrzeug. *Human-Computer Interaction in the Car*. Available from http://www.eislab.net/publications/2014/TR2014-HumanComputerInteractionInTheCar.pdf#page=11

Schneiderman, R. (2013). Car makers see opportunities in infotainment, driver-assistance systems [special reports]. *Signal Processing Magazine, IEEE,30*(1), 11-15. Available from http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6375942 (Accessed October 17, 2014)

Qt Project (2013). Open Source Licensing of Qt. Available from: http://qt-project.org/doc/qt-5/opensourcelicense.html (Accessed October 27, 2014)

Appendix

Exhibit 1


EB GUIDE Features

Multitouch modeling in EB GUIDE Studio

Details

Type:	New Feature
Priority:	Critical
Affects Version/s:	GUIDE
Component/s:	GRP Documentation. GRP
Labels:	GUIDE5

Description

Stakeholders

- Product Management, Demos, Tier 1 and OEM customers like Renault.

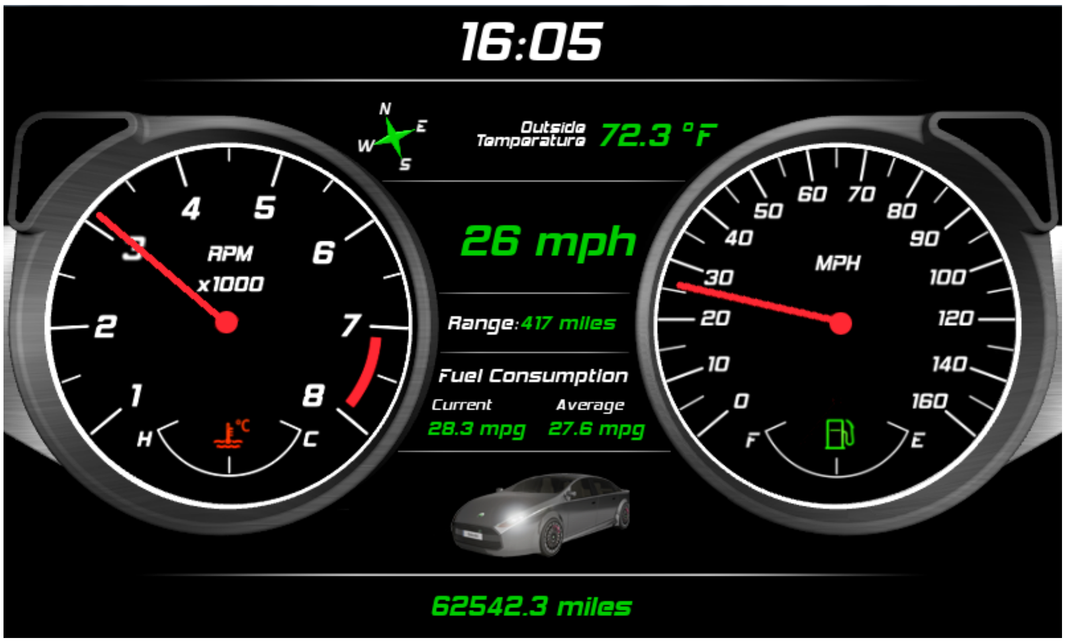
Fulfillment criteria

Remarks

Benefits

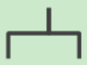
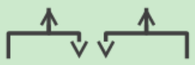
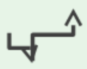
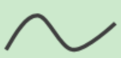
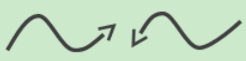
Template for feature description as a Jira document (Elektrobit, 2013)

Exhibit 2



Example of an HMI running on a cluster instrument display (Elektrobit, 2013)

Exhibit 3

Gesture	Drawings	Name	Reaction in GUIDE 5 Demo
		Roof	Show a popup „starting navigation to home“
		Swipe right	TBD
		Swipe left	Delete characters in the arabic input screen
		Check mark	Confirm „Ok“ before the cluster screen
		Wave	Goto media player

Multi-touch gesture suggestions (Elektrobit, 2012)

Exhibit 4

User Story (Backlog Item)	Priority	Complexity
Create architecture overview		
Extend touch engine for multi-touch		
Apply new MT controls to EB GUIDE demo		
Create path-gesture engine		
Create set of gestures		
Build libraries to support usual platforms		
Add chapter “multi-touch” to product documentation		

Overview of a set of early user stories for multi-touch, snapshot of the initial backlog[1]

1 The backlog is fictional but derived from the original data

About this Case

This teaching case was taken from the Product Management by Case collection, a collection of free cases for teaching product management, available at http://pmbycase.com.

Conceptual guidance and teaching notes are available to lecturers. To receive those, please send an email to case-requests@group.riehle.org or dirk@riehle.org.

Case Copyright

Case Credits

Contributions by Ann Barcomb, Hendrik Koch, Dirk Riehle.

Copyediting by Dinakar Geddapu, Dirk Riehle.

Case Data

Short code:	Case-2014-03-Elektrobit-Specifying-Wow
Case license:	CC BY SA 4.0
Case revision:	180807

More Cases

Case 2012-01:	Ensuring innovation at Method Park
Case 2013-02:	Two-sided markets at Netdosis
Case 2014-01:	User experience design at Immowelt
Case 2014-02:	Switching suppliers at Nokia
Case 2014-03:	Specifying ‘wow!’ at Elektrobit
Case 2016-01:	Licensing choices at ownCloud
Case 2016-02:	Stock options at Caldera
Case 2016-03:	Hard software marketing choices at ownCloud
Case 2016-04:	Pricing at Everest SARL
Case 2017-01:	The case of SUSE Manager