

**Increasing the Efficiency of your  
Automotive Software Development**

Trend Study  
2005

**Arthur D Little**

## Contents

---

<b>1</b>	<b>About this study</b>
2	Executive Summary
3	Trends in Automotive Software Development
4	Levers to Increase Efficiency
5	Arthur D. Little

## During the second quarter of 2004 Arthur D. Little conducted research into how automotive companies can improve the efficiency of their embedded software development processes

### Background and objectives of this study

- This study is a follow up to our 2002 automotive electronics trend study with 55 companies which demonstrated that both E/E and Software are going to dramatically reshape the automotive landscape until 2010
- With software taking on a more important role in the performance and differentiation of electronics systems, our OEM and Tier 1 clients have increasingly asked us to focus on helping them improve the efficiency of their embedded software development processes
- The objectives of this study where to :
  1. Understand the key challenges in embedded software development faced by both automotive manufacturers and suppliers
  2. Identify the most important efficiency improvement levers at their disposal and the ways to operate them

## In developing our understanding of improving embedded SW development efficiency, we have benefited from a number of different sources

Study Sources	Participating Companies
<ul style="list-style-type: none"><li>• In-depth interviews conducted with 18 companies representing different parts of the automotive E-E value chain<ul style="list-style-type: none"><li>– Study of improving internal development processes (participation from 13 companies)</li><li>– Study of delocalization and off-shoring issues (participation from 7 suppliers)</li></ul></li><li>• Desk top research of relevant industry and technology data, and related studies by Arthur D. Little (e.g. recent SW flashing study conducted with both car and truck OEMs)</li><li>• Arthur D. Little internal experts on automotive software development</li></ul>	<ul style="list-style-type: none"><li>■ AUDI</li><li>■ BMW</li><li>■ DaimlerChrysler</li><li>■ Volvo Bus</li><li>■ Volvo Car</li><li>■ Volvo Trucks</li><li>■ Volkswagen</li><li>■ Brose</li><li>■ Continental</li><li>■ Delphi</li><li>■ Freescale</li><li>■ Hella</li><li>■ Johnson Control</li><li>■ Mecel</li><li>■ Siemens VDO</li><li>■ Visteon</li><li>■ Webasto</li><li>■ ZF</li></ul>

## Contents

---

1	About this study
<b>2</b>	<b>Executive Summary</b>
3	Trends in Automotive Software Development
4	Levers to Increase Efficiency
5	Arthur D. Little

## Automotive companies are under severe pressure to increase the efficiency of their embedded software development as part of their electronic systems

### Current Situation

- Software and electronics play an increasingly important role to realize innovations in the automotive industry. During the last years, the number of ECUs and the complexity of electronic architectures has constantly risen, reaching a peak of more than 70 ECUs per vehicle with the current upper-premium class models such as Mercedes S-class, BMW 7-series and Volkswagen Phaeton and a share of electronics and software on automobile value added of up to 25 – 30% and more.
- At the same time, in a race to introduce new features and technologies – not always recognized from a customer perspective as providing real added customer value – the industry had tough times to maintain the vehicle quality and sometimes failed in doing so. Not only did quality problems increase, but also development and validation costs of electronic systems.
- Reduced complexity – simpler architectures with less ECUs (we talk approx. 20 within specific vehicle domains), a standardized software architecture (mainly driven through the AUTOSAR initiative) and a more focused approach towards customer requirements seem to be the industry’s answers to tackle these problems.
- On the other hand, while software-based functionality will further increase, cost pressure is also increasing. This will not only make some estimates on future automotive software market figures obsolete, but also put additional pressure on engineering to increase their development efficiency. While OEMs need to focus on standardization, automation and frontloading of their development processes, suppliers need to increase software/module re-use, further optimize and support their processes and de-localize development activities where appropriate.

## Our study identifies three efficiency improvement levers : Standardization, Automation and Delocalization

### Levers to increase automotive software development efficiency

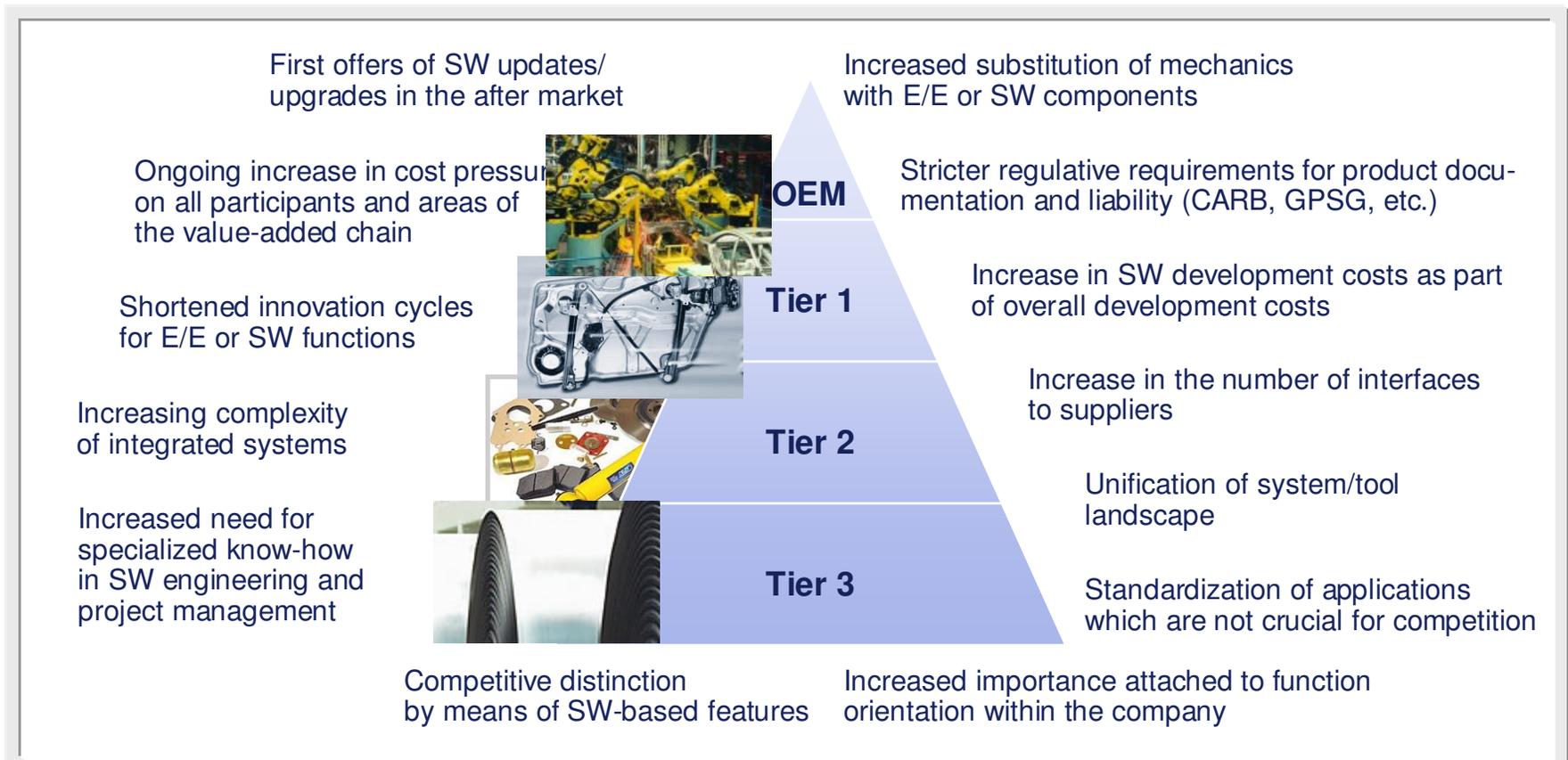
- Standardization of architecture, system and module re-use, and a higher automation and tool support are the first two important levers to reduce software development costs.
  - Our analysis among OEMs and suppliers demonstrates that huge savings potentials in the development process are still not realized – overall, participants to this study estimate a potential reduction of total man-hours per project of around 30% to be feasible.
  - However, our own project experience shows that, by applying state-of-the-art methods and tools, e.g. model-based specification, automatic code generation, and a strict requirements management, savings of up to 50% can be realized within a 2 years time-frame. While complexity (Lines of Code) doubled at the same time, the overall productivity in automotive software development therefore increased by 300% and the number of bugs in integration testing on ECU level has been reduced by up to 90%.
- Delocalization of embedded systems development to low cost engineering countries is another big lever to improve development efficiency. It has already been executed successfully by a number of automotive companies and will most likely further increase.
  - In this study, we have analyzed the approach of seven leading suppliers (total number of software developers > 6,000) and found that software development labor costs can be reduced in a range of 30-50%.
  - However, as delocalization of embedded systems development also bears certain risks, it has to be planned and executed carefully. The preferred and most successful way here is not to outsource to third parties, but to set-up and seamlessly integrate own subsidiaries located abroad.

## Contents

---

1	About this study
2	Executive Summary
<b>3</b>	<b>Trends in Automotive Software Development</b>
4	Levers to Increase Efficiency
5	Arthur D. Little

## Software is a key driver in the race to develop automotive innovations faster, cheaper and better



Source: Arthur D. Little Research and Benchmarking Study on Software Download 2004

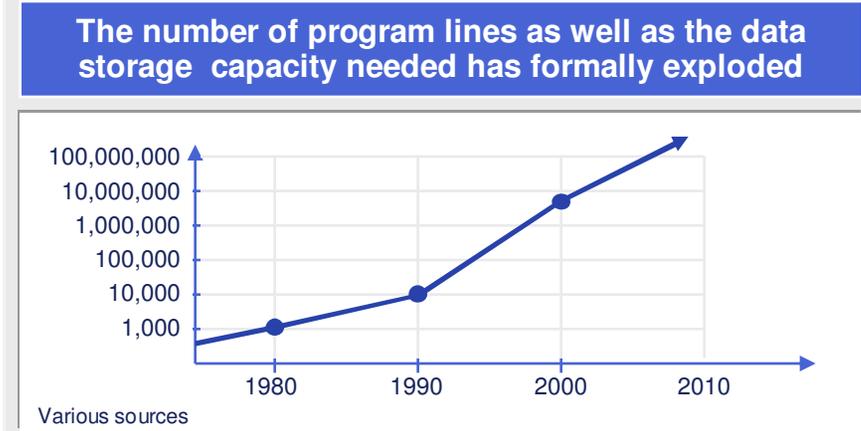
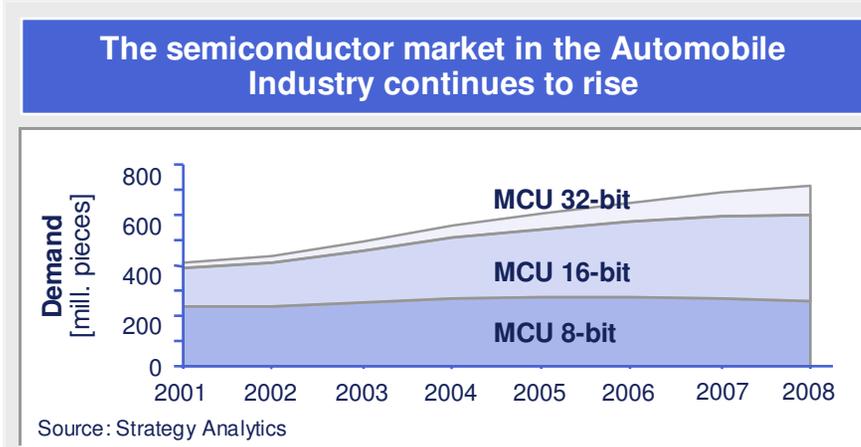
CARB: California Air Resources Board; GPSG: Equipment and Product Safety Act

**Some studies forecast a share of software on automobile value for the year 2010 of € 100 bn and estimate the volume in 2000 by € 25 bn – these numbers seem to be far too high**

Estimated volume	Plausibility check
<p><b>€ 25 bn from software (ca. 5% vehicle value) in 2000<sup>1)</sup></b></p>	<p>⌢ 100,000 software (and function) developers (at a maximum cost of € 250,000 per annum/developer) working in 2000 for the automotive industry is not a plausible figure</p> <p>A comparison:</p> <ul style="list-style-type: none"> <li>■ Total number of software developers at Siemens VDO, Delphi, Continental, Visteon, Johnson Control, ZF: about 6,000 people. Share of software development costs on revenues: &lt;&lt;5%<sup>2)</sup></li> <li>■ Microsoft turnover in 2000: ~\$23 bn</li> </ul>
<p><b>€100 bn from software in 2010<sup>1)</sup></b></p>	<p>This figure seems also not plausible, since it represents:</p> <ul style="list-style-type: none"> <li>⌢ 400,000 software developers in 2010</li> <li>⌢ 50% of the world market for software in 2003<sup>3)</sup></li> </ul>

1) Source: HVB/Mercer    2) Source: ADL Benchmarking Study 2004    3) Source: EITO 2004: Global Power of Software: € 200 bn

**However, with the proportion of electronics in a vehicle continuing to rise, the share contributed by software on the value added will double by 2010**



- Number of software developers in the Automotive industry in 2004 about 20,000 – 25,000 world-wide (without OEM)
- Value added about € 4 – 6 bn.
- Statement of electronics suppliers: number of software developers will double by 2010, development costs will decrease by 30% – 50%

▼

**"Non-captive" market volumes  
"Software in the Automobile<sup>2)</sup>"**

Year	Value Added	Software Volume
2004	4 – 6	6
2010	8	10

Source: Arthur D. Little

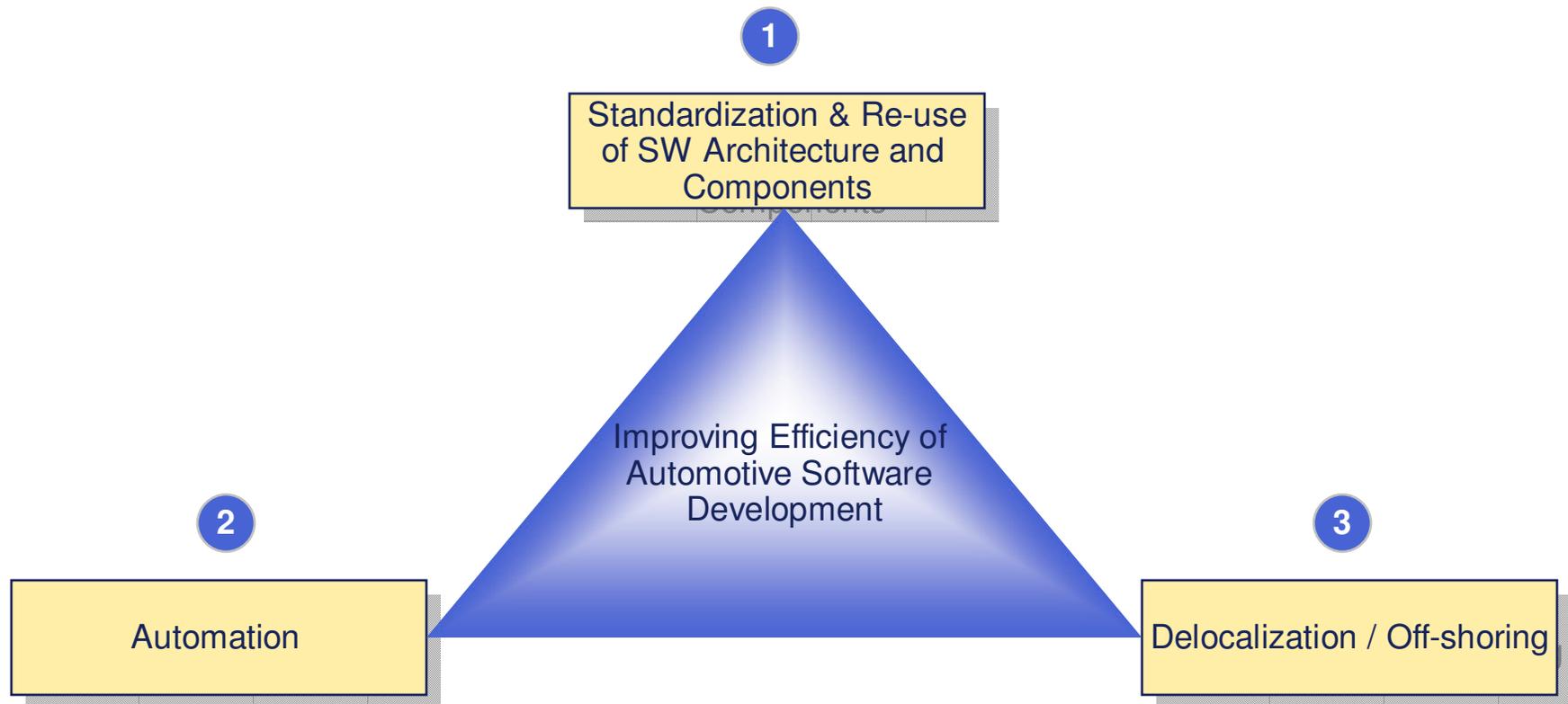
1) Source: ADL-Benchmarking Study 2004  
 2) without services outside the vehicle, for example vehicle fleet management, Infotainment, etc.)

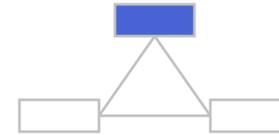
## Contents

---

1	About this study
2	Executive Summary
3	Trends in Automotive Software Development
<b>4</b>	<b>Levers to Increase Efficiency</b>
5	Arthur D. Little

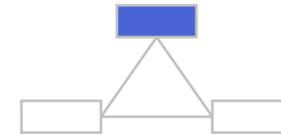
Automotive companies typically use three main levers to improve the efficiency of their software development processes





**Our study demonstrates that standardization of SW architectures and re-use of components bear huge potential**

Value Driver	Design requirements	Approaches
<ul style="list-style-type: none"> <li>■ Premium price</li> <li>■ Residual value</li> <li>■ Extra equipment strategies</li> <li>■ Development costs</li> </ul>	<ul style="list-style-type: none"> <li>■ Brand-specific functionality</li> <li>■ High service ability also late in product life cycles</li> <li>■ Flexibility of system concept</li> <li>■ Economics of scale/re-use</li> </ul>	<p><b>Standard Software Architecture</b></p> <ul style="list-style-type: none"> <li>■ Basic platform with standard applications</li> <li>■ Brand-specific module offerings</li> <li>■ Clearly defined interfaces (Make-or-Buy)</li> <li>■ Downsizing concept – comprehensive application concept for different segments</li> </ul>
<ul style="list-style-type: none"> <li>■ Purchasing costs</li> </ul>	<ul style="list-style-type: none"> <li>■ Hardware-independent design</li> <li>■ No monopolies, increased competition</li> </ul>	<p><b>HW Abstraction</b></p> <ul style="list-style-type: none"> <li>■ SW versions compatible with different hardware and software configurations</li> </ul>
<ul style="list-style-type: none"> <li>■ Warranty costs</li> <li>■ Update costs</li> </ul>	<ul style="list-style-type: none"> <li>■ Quality proof - systems</li> <li>■ Fault tolerance in the system</li> <li>■ Software download</li> </ul>	<p><b>Standard processes &amp; Re-use</b></p> <ul style="list-style-type: none"> <li>■ Disentangle development &amp; diagnosis</li> <li>■ Re-use (e. g. SW libraries)</li> <li>■ Process standards (testing for error cases, coding, flashing, diagnosis, etc.)</li> </ul>
<p>➔ <b>Study participants stated an average reduction potential of 15 to 20% (man-hours per project) by increasing the degree of standardization compared to today</b></p>		



**However, OEMs need to adapt their development processes to be able to take full advantage of “plug&play” architectures allowing to flash and re-use SW in different ECUs/network environments**

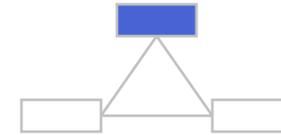
**OEM development processes not yet designed to realize full potential of standard architectures**

- Application development is not yet performed independent from development of communication layer
- A "statically" defined communication matrix (K-Matrix) is used by major German OEMs to allow ECU re-use in different network environments but also requires validation of communication
- A more holistic and dynamic approach, such as offered by Volcano, would allow re-use of application software in different network environments without need for communication validation. Signal and frame mapping can be completely automated and requires only changes in specific ECUs
- OEM development processes usually have to be adapted to realize these potentials, since only a few OEMs have implemented a consistent systems engineering approach

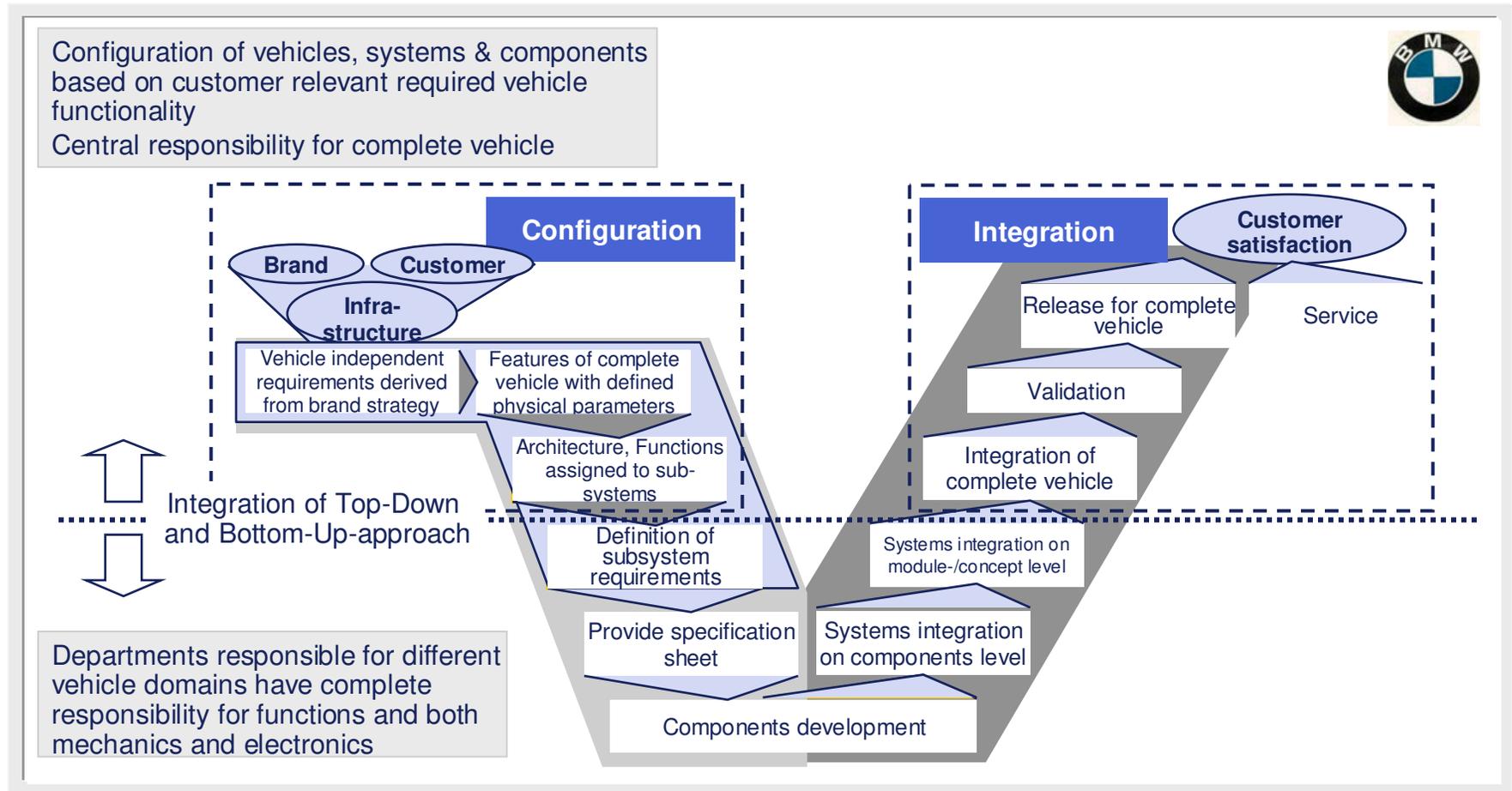
**OEMs will not use flash memories in all vehicle domains (Passenger cars)**

Functional Domains	2004 <sup>1)</sup>					2007 <sup>1)</sup>				
	20%	40%	60%	80%	100%	20%	40%	60%	80%	100%
Body & Comfort	69%					63%				
Infotainment	88%					88%				
Chassis Control	88%					94%				
Powertrain & Transmission	88%					88%				
Active/Passive Safety	78%					84%				
Man Machine Interface	100%					100%				

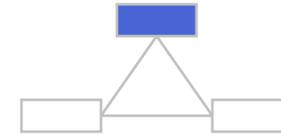
1) Percentage of control units in a functional domain, in which flash memory is used mainly or exclusively  
 Source: Interviews; ADL Study on Software Flashing, 2004 (Truck OEMs show higher rates in some areas)



# BMW has implemented a clear systems engineering approach in its processes and organization to realize distributed functions with high quality

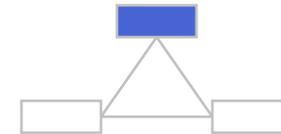


Source: G. Reichart, BMW, Euroforum Jahrestagung Systems Engineering, 1.2.2005; illustration by ADL

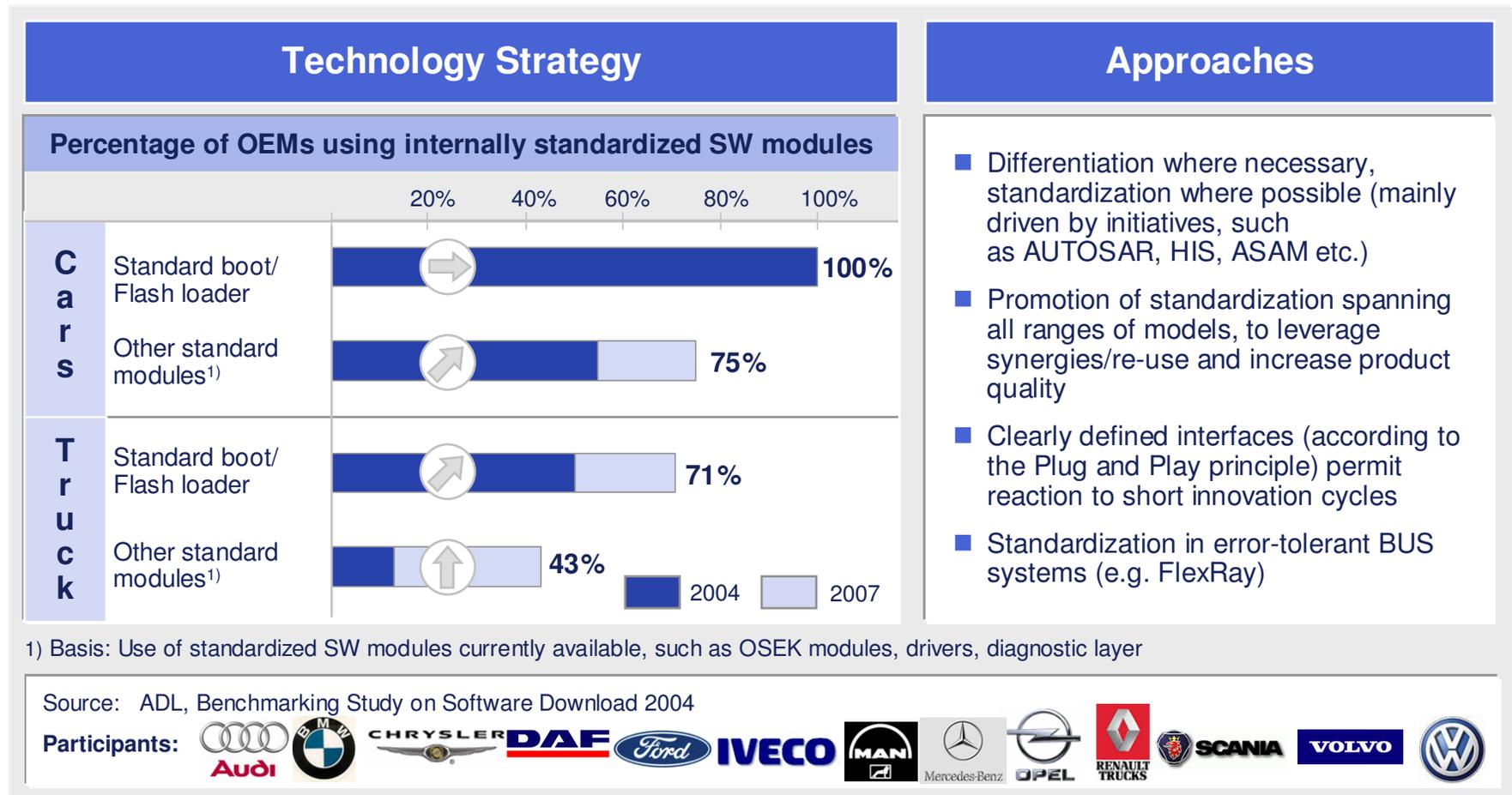


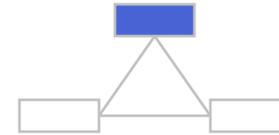
## Using a "holistic" network design approach, Volvo Bus implemented "Plug & Play" to realize customer-specific vehicle functionality

Challenge	Solution
<p data-bbox="275 623 489 695"><b>VOLVO</b></p>  <ul data-bbox="285 899 1045 1321" style="list-style-type: none"><li>■ Buses and coaches are characterized by a very high variance in the body – besides "standard" buses offered, body builders around the world build up customer specific bodies based on the platform offered by Volvo bus</li><li>■ Volvo Bus had to build up an infrastructure to allow the easy integration of bodybuilder functions, by avoiding the need for integration tests</li></ul>	<ul data-bbox="1178 654 1961 1300" style="list-style-type: none"><li>■ Using a "holistic" network design approach, Volvo Bus built up a signal data base and a common function library to which all body builders have access via a global website.</li><li>■ The required body functionality is implemented independent from the ECU hardware. Frames and messages are automatically compiled, whether the ECUs correspond to the assigned technical registration is tested automatically and timing and CPU load is verified by a postprocessor.</li><li>■ No integration test is necessary, thus enabling Volvo bus partners to realize basically an "endless" number of customer variants in a very efficient way.</li></ul>



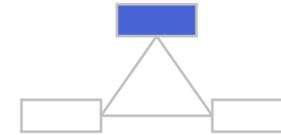
## Most OEMs have implemented standard software cores and promote cross-industry standardization, e.g. in the AUTOSAR initiative





**The AUTOSAR initiative is seen as positive by all study participants ; the first AUTOSAR compliant ECUs will be seen in the 2007/2008 model year**

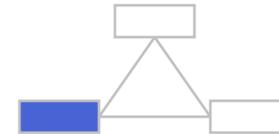
- AUTOSAR has two main objectives:
  - 1<sup>st</sup>: definition of a truly standardized software core with clearly defined interfaces
  - 2<sup>nd</sup>: definition of a standardized way of communication between different functions in order to move functions easily from one ECU to another and integrate software functions from different vendors
- Current development within AUTOSAR is mainly driven by one OEM who will introduce the first AUTOSAR compliant ECUs around 2007/early 2008.
- Standards will be seen first in the comfort and chassis area, infotainment and powertrain will see more "quasi" standards driven by suppliers, since expected gains from overall standardization are much lower (large and dominant suppliers are already using their own standard architectures in this area)
- The realization of the second objective is to be expected in the longer term. There are still intense discussions in the consortium about the "right way" to design the communication layer of the vehicle network and the right tools to support the move of functions from one ECU to another. It is Arthur D. Little's opinion that the foundation for that has been laid with Volcano's "Holistic approach".



## Standardization will put high cost pressure on software suppliers and reduce costs for basic and standard software modules for OEMs

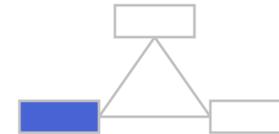
Procurement Behavior of OEMs	
<b>Basic software:</b> <ul style="list-style-type: none"><li>■ Communication layer (CAN, LIN, ...)</li><li>■ Network management</li><li>■ System services (diagnostics protocols, ...)</li></ul>	<ul style="list-style-type: none"><li>■ Standardization spanning all manufacturers</li><li>■ Hardware-oriented functions (ECU) from semi-conductor manufacturers</li><li>■ Pricing: Open book calculation</li></ul>
<b>"Standard" functions:</b> <ul style="list-style-type: none"><li>■ Window lifter</li><li>■ ...</li></ul>	<ul style="list-style-type: none"><li>■ Standardization of the OEM or functions specific to the supplier</li><li>■ Massive cost pressure due to software target costing (25% – 30%)</li><li>■ Hardware-oriented functions from the hardware supplier, allocation over the unit price</li></ul>
<b>Differentiating functions:</b> <ul style="list-style-type: none"><li>■ Gear shifting strategy</li><li>■ ...</li></ul>	<ul style="list-style-type: none"><li>■ Implementation by an OEM or by a "3rd party" commissioned by the OEM</li><li>■ Integration by the hardware supplier</li><li>■ No readiness on the part of the OEM to pay for licenses to use innovative functions / algorithms ("Classic" control systems); allocation over the unit price</li></ul>

➔ "Win-Win" business model for OEMs and software suppliers has not been developed yet



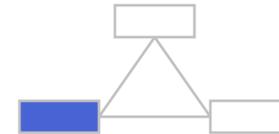
## The automotive industry is following the V-design cycle, with tool support available for all tasks

The V-Design-Cycle <i>illustrative</i>	Tool Support
<p>As a series of "incremental V's" during product development</p>	<ul style="list-style-type: none"><li>■ Various tools are available on the market to support specific development tasks. No tool covers all aspects, thus leaving still room for proprietary developments driven by OEMs.</li><li>■ No tool currently covers all aspects of architecture and embedded systems development, the degree of integration between different development tools is low and will stay low short to medium-term.</li><li>■ OEMs/suppliers don't want monopoly situations with an integrated solution from one provider, there is little acceptance for proprietary tool chains. Standardized interfaces are preferred instead.</li><li>■ Nevertheless, whether an integrated tool (or two) will establish a "quasi" standard is not decided yet. Interviewees expect a consolidation on the tool vendor side.</li></ul>



## The potential of frontloading by using model-based specification and functional design and advanced methods for architecture and network design is not fully realized yet

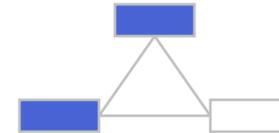
The Tools Used	Status of Tool Usage
<p><b>Requirements capturing &amp; management</b></p> <ul style="list-style-type: none"> <li>■ DOORS</li> <li>■ UML</li> <li>■ ...</li> </ul>	<ul style="list-style-type: none"> <li>■ DOORS is used by OEM and suppliers, but there is still room for efficiency improvement in communication between OEMs and suppliers due to missing standards and missing tool support of distributed development.</li> </ul>
<p><b>Analysis and functional design</b></p> <ul style="list-style-type: none"> <li>■ UML</li> <li>■ STATEMATE</li> <li>■ Matlab</li> <li>■ Ascet-SD</li> <li>■ CAPE/C</li> <li>■ DaVinci</li> <li>■ ...</li> </ul>	<ul style="list-style-type: none"> <li>■ Model-based specification is in the focus of OEM investments and on the way to be used in series development.</li> <li>■ Some OEMs (especially BMW) and large tier-1's are fairly advanced compared to others who are still in experimental mode.</li> </ul>
<p><b>Partitioning, Architecture &amp; network design</b></p> <ul style="list-style-type: none"> <li>■ Proprietary solutions</li> <li>■ Cape/C (Cape)</li> <li>■ DaVinci (Vector)</li> <li>■ Integrio (ETAS)</li> <li>■ VNA/LNA (Volcano)</li> <li>■ ...</li> </ul>	<ul style="list-style-type: none"> <li>■ No tool available covers all aspects of partitioning, architecture and network design.</li> <li>■ Some fairly advanced methodologies and tools with the potential for a "true frontloading" have been developed (e.g. in the TITUS project involving Vector and ETAS as tool vendors or the "Holistic approach" of Volcano Technologies), but failed to get wide acceptance so far, since at least some large OEMs were heavily invested in existing processes and were not ready yet to implement the necessary changes in organization, processes and staff qualification.</li> </ul>



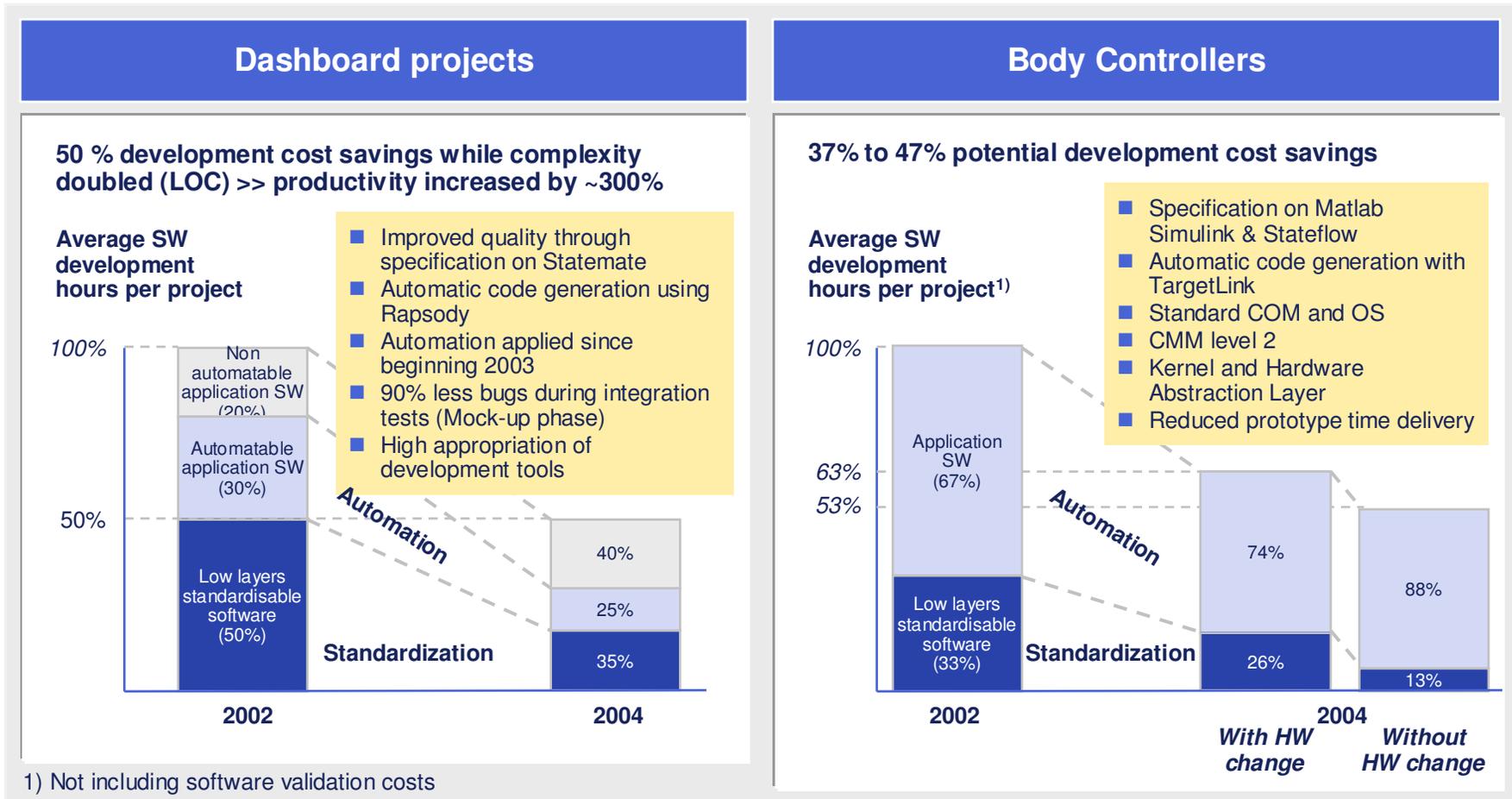
## Integration test and validation still require considerable resources, especially on the OEM side

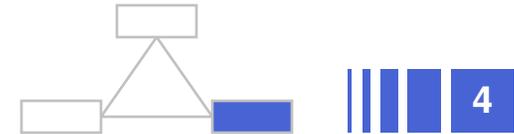
The Tools Used	Status of Tool Usage
<p><b>Code generation</b></p> <ul style="list-style-type: none"> <li>■ ASCET-SD</li> <li>■ Matlab</li> <li>■ ...</li> </ul>	<ul style="list-style-type: none"> <li>■ Code generation is used to some extent by most study participants, but still not efficient enough to be in all areas (especially for software in body domain).</li> <li>■ Wider acceptance has been reached especially in the powertrain area, where memory is less an issue.</li> <li>■ There is also the not completely solved issue, that high process discipline is necessary in order to change the model and go through automated code generation instead doing quick bug fixes on code level. Since "semiautomation" becomes inefficient and risky if further changes are necessary, this might still be an issue in mid-term.</li> </ul>
<p><b>Component/ Integration test / HIL Validation</b></p> <ul style="list-style-type: none"> <li>■ Proprietary tools</li> <li>■ dSpace</li> <li>■ ...</li> <li>■ Test Director</li> <li>■ ...</li> </ul>	<ul style="list-style-type: none"> <li>■ Test automation is seen by the suppliers as the area with the highest improvement potential in terms of integrating the tool chain, while some OEMs have already achieved a high level of automation in test case generation.</li> <li>■ Nevertheless, OEM test efforts are considerably high, especially in integration tests, whereas "true frontloading" is only done by a few OEMs avoiding the need for integration tests at all (see Volvo Bus example on p.15<sup>1)</sup>).</li> </ul>

1) Other OEMs which implemented a similar approach are Volvo Car, LandRover, Jaguar, AstonMartin and MG.



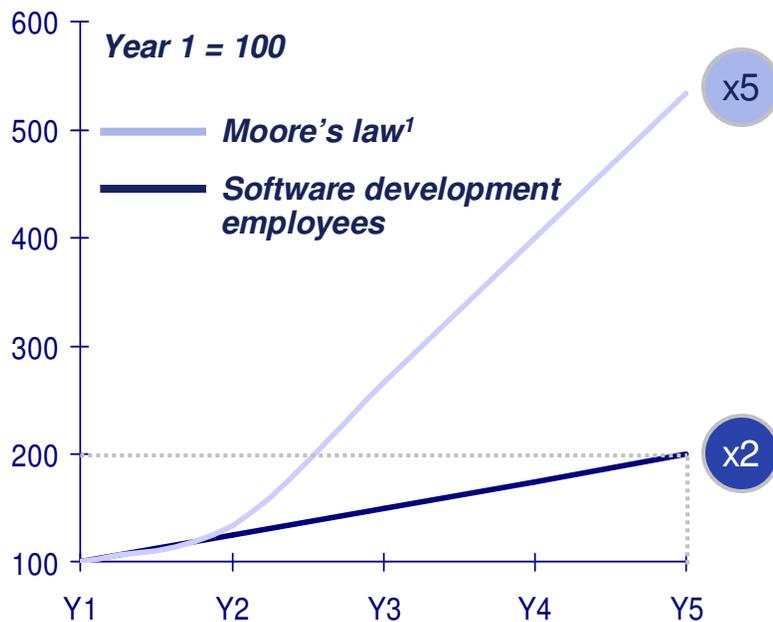
Whereas interviewees estimate an average cost reduction potential of 30% due to standardization and automation, our own project experience suggests possible reductions of up to 50%





## Delocalization is used by suppliers to primarily reduce increasing software development labor costs

Driven by increasing electronic capabilities all suppliers declare that SW headcount is set to double over the next 5 years



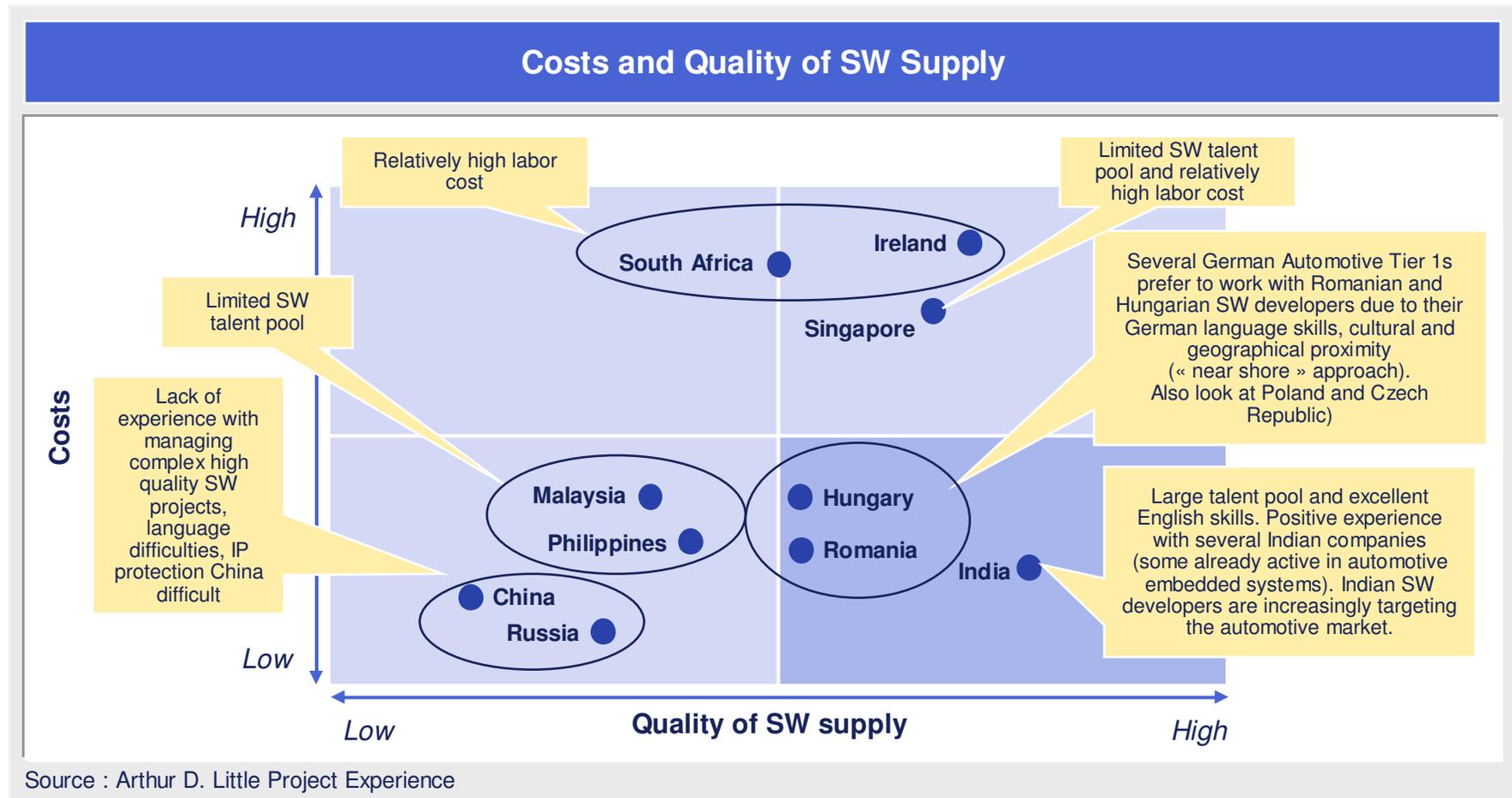
1) Moore's law (prediction) : the number of transistors which could be put on a single computer chip would double every 18 months

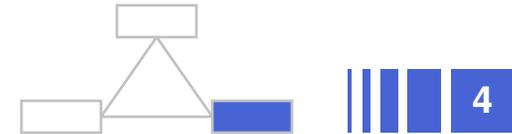
Source : 7 suppliers responded specifically to our questions related to delocalization

Delocalization is a major lever to offset this net increase in software development labor costs while enhancing SW organization's flexibility

- Through delocalization, all interviewed companies target to reduce software development labor costs by 30% to 50%
- Benchmarked suppliers intend to reach this objective within 5 to 10 years
- Besides SW costs reduction, suppliers' second major interest through delocalization is to increase software organization flexibility
- Two respondents indicate that enhancing software products quality was their third main objective through delocalization

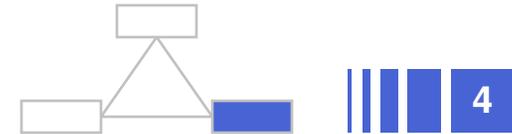
## India is the preferred candidate for SW delocalization but Eastern European countries should be studied as well





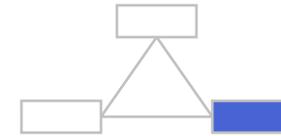
## Until now delocalization has centered around the actual implementation and coding activities

Delocalized software process activities		Comments																																							
<table border="1"> <tr> <td>0</td> <td>Requirements</td> <td>↔</td> <td>Validation</td> <td>3</td> </tr> <tr> <td>0</td> <td></td> <td></td> <td></td> <td>4</td> </tr> <tr> <td>2</td> <td>Specification</td> <td>↔</td> <td>Integration Test</td> <td>4</td> </tr> <tr> <td>3</td> <td></td> <td></td> <td></td> <td>6</td> </tr> <tr> <td>3</td> <td>Design</td> <td>↔</td> <td>Unit Test</td> <td>5</td> </tr> <tr> <td>5</td> <td></td> <td></td> <td></td> <td>7</td> </tr> <tr> <td>5</td> <td colspan="2">Coding</td> <td></td> <td></td> </tr> <tr> <td>7</td> <td></td> <td></td> <td></td> <td></td> </tr> </table>	0	Requirements	↔	Validation	3	0				4	2	Specification	↔	Integration Test	4	3				6	3	Design	↔	Unit Test	5	5				7	5	Coding				7					<ul style="list-style-type: none"> <li>As preliminary knowledge steps, module coding and testing are the first delocalized software development activities</li> <li>"Specification" is also considered as a core competence by all respondents; only two of them have already delocalized it and another one plans to delocalize it in the future</li> <li>Software project management is performed inshore as long as delocalized SW units have not reached the required level of maturity to handle internally project management activities</li> <li>Advanced technology software programs are not delocalized (core competence)</li> </ul>
0	Requirements	↔	Validation	3																																					
0				4																																					
2	Specification	↔	Integration Test	4																																					
3				6																																					
3	Design	↔	Unit Test	5																																					
5				7																																					
5	Coding																																								
7																																									
<p># Number of suppliers (out of 7) having already delocalized part or all of this activity</p> <p># Number of suppliers (out of 7) planning to delocalize part or all of this activity next</p> <p>Delocalization scope</p>																																									

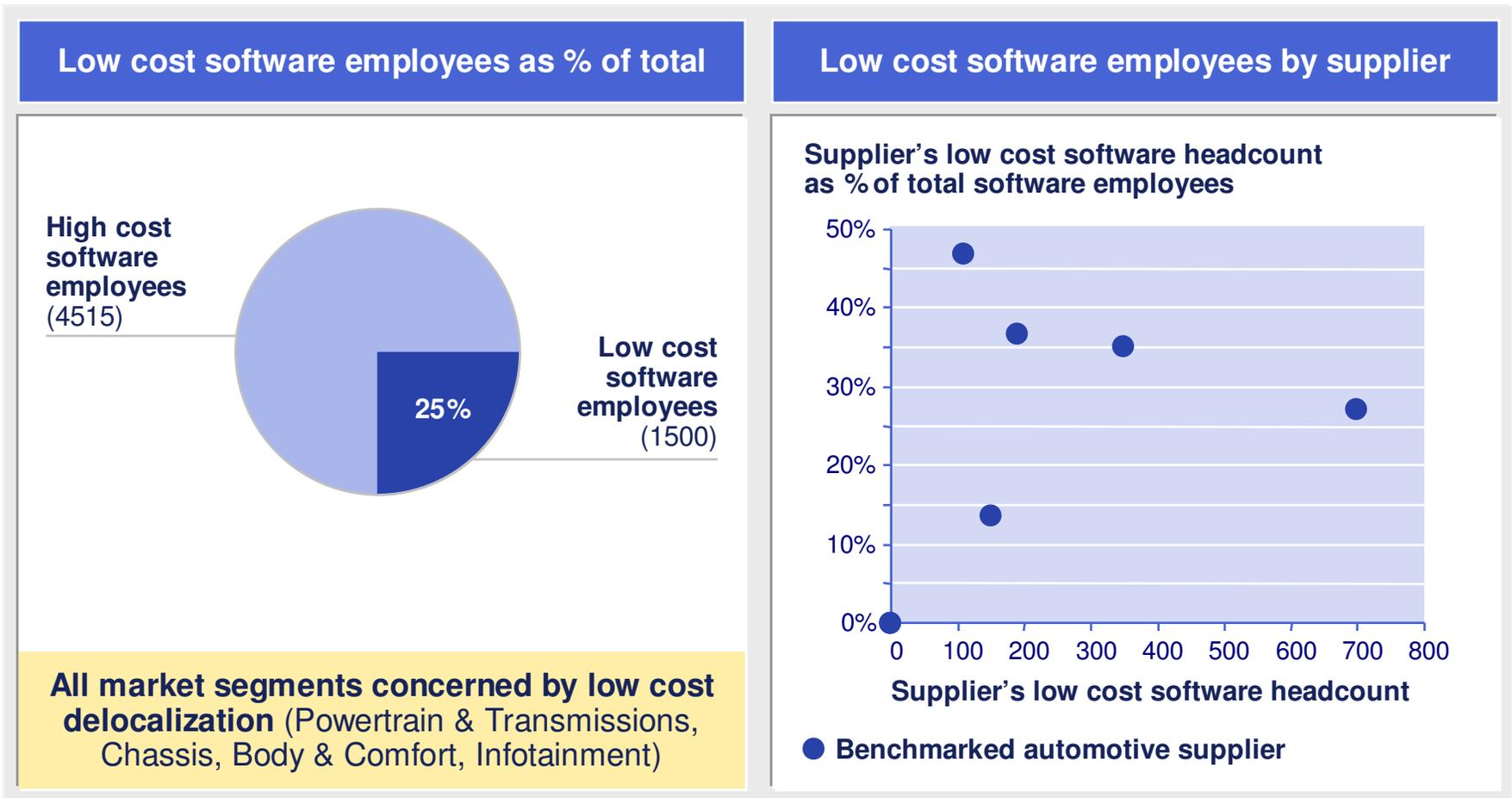


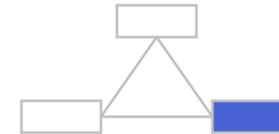
## However, all companies interviewed also plan to transfer their function development operations

Delocalized Software Layers	Comments
<p><b>4</b> <b>7</b> Application functions</p> <p><b>5</b> <b>7</b> COM, Devices, Hardware Abstraction Layer</p> <p><b>OS</b></p> <p><b>#</b> Number of suppliers (out of 7) having already delocalized part or all of this (these) software layer(s)</p> <p><b>#</b> Number of suppliers (out of 7) planning to delocalize part or all of this (these) software layer(s) next</p> <p> Delocalization scope  Low software layers</p>	<ul style="list-style-type: none"> <li>■ 4 suppliers have already delocalized application functions as well as low software layers development of different components / sub-systems</li> <li>■ So far only two suppliers have delocalized some critical application functions development. The other suppliers plan to do it in the future once the offshore software team has acquired the needed competencies (project management, technology and functions expertise...)</li> <li>■ Respondents recognize that low software layers development is more easy to delocalize as requirements / specifications are more simple to define and to stabilize</li> <li>■ Two suppliers only develop application functions and devices and totally outsource HAL, OS &amp; COM layers development to specialized IT companies or co-develop them with other automotive suppliers</li> </ul>



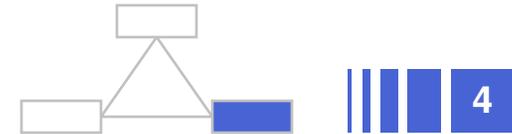
## Significant volumes have already been transferred by large suppliers



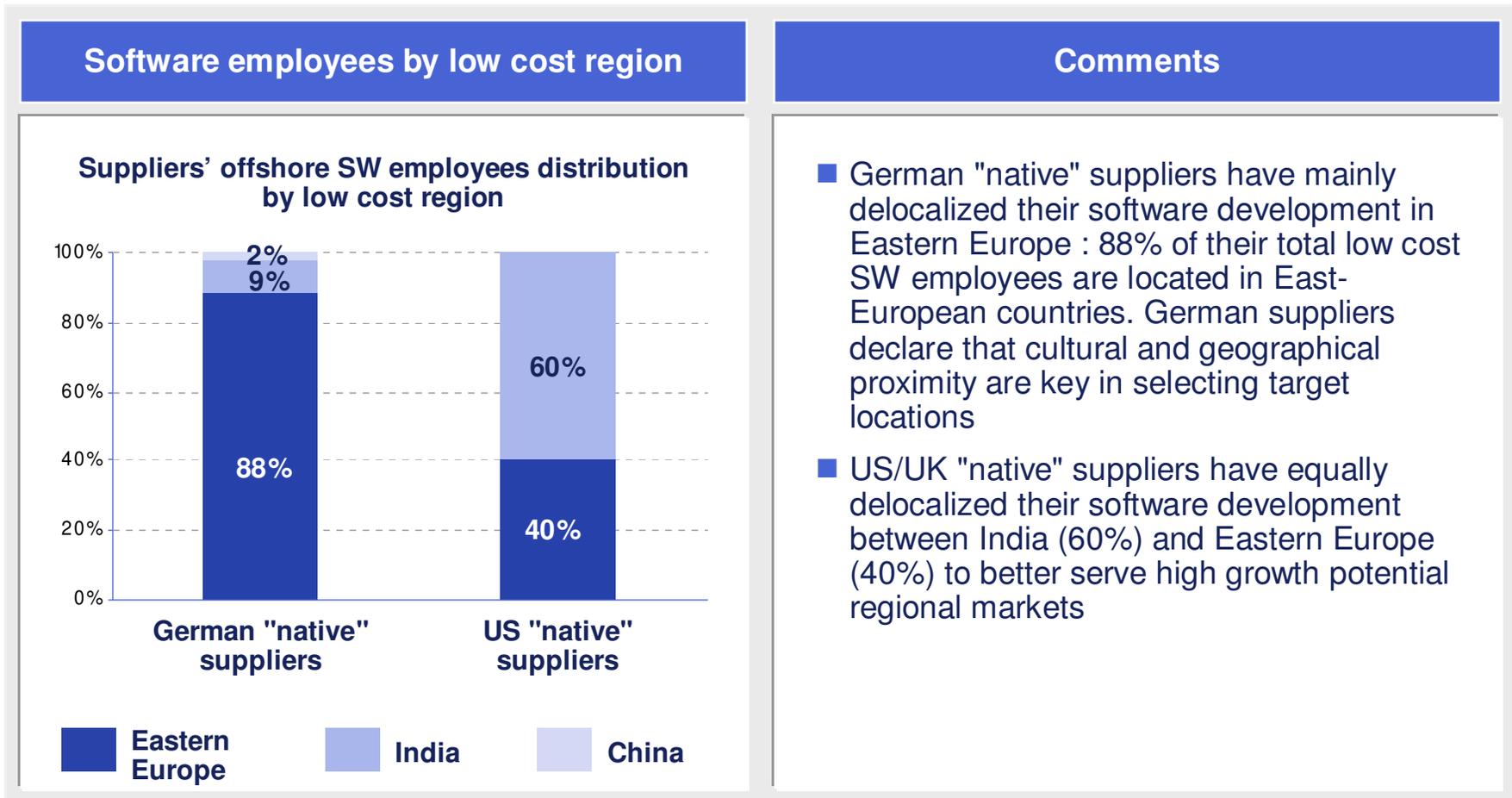


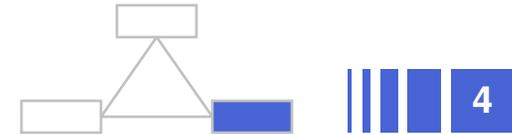
Besides low labor costs and skills availability, cultural proximity (e.g. language) is a determining choice factor of target locations (1/2)

Software employees by low cost country			Suppliers' distribution by country			
Country	Low cost SW employees distribution by country	Hourly cost by software employee	Country	Number of suppliers located in the country	Supplier's origin	
Romania	46.7%	€15			Germany	US
India	31.3%	€14	Romania	2	2	0
Poland	10.0%	€22	India	3	1	2
Bulgaria	7.3%	€20	Poland	1	0	1
Hungary	3.3%	€20	Bulgaria	1	0	1
China	1.3%	€12	Hungary	1	1	0
			China	1	1	0
Number of low cost locations by supplier						
S1	S2	S3	S4	S5	S6	S7
3	2	2	1	1	0	0

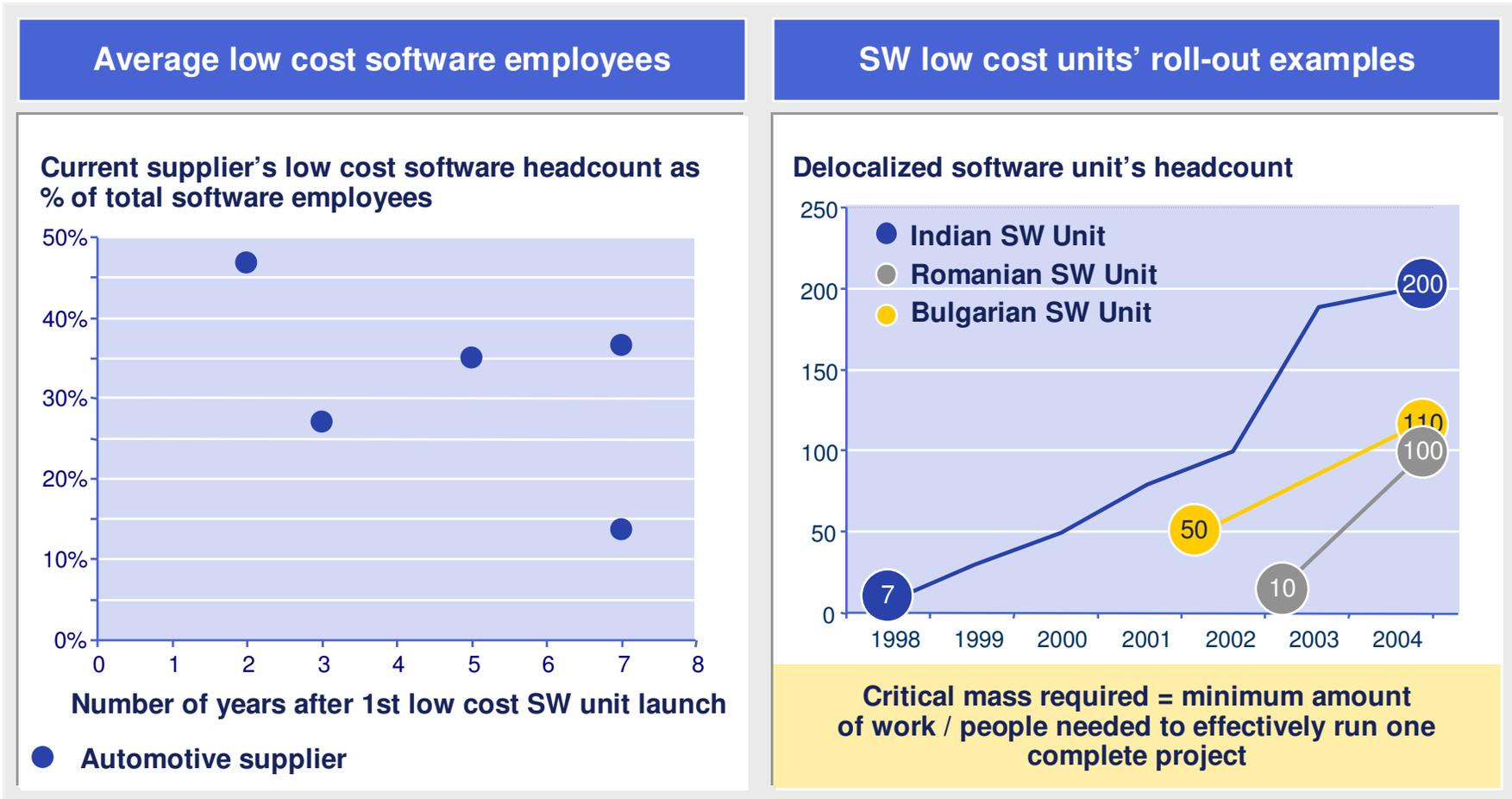


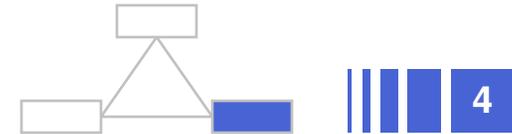
## Besides low labor costs and skills availability, cultural proximity (e.g. language) is a determining choice factor of target locations (2/2)



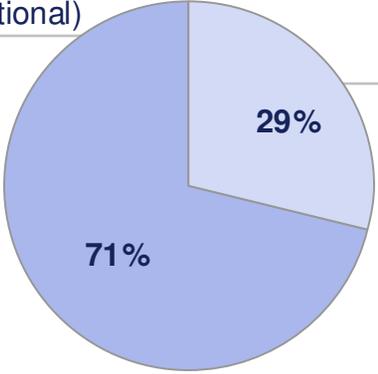


The first software development delocalization operations have been initiated 7 years ago by two benchmarked suppliers – other suppliers started later

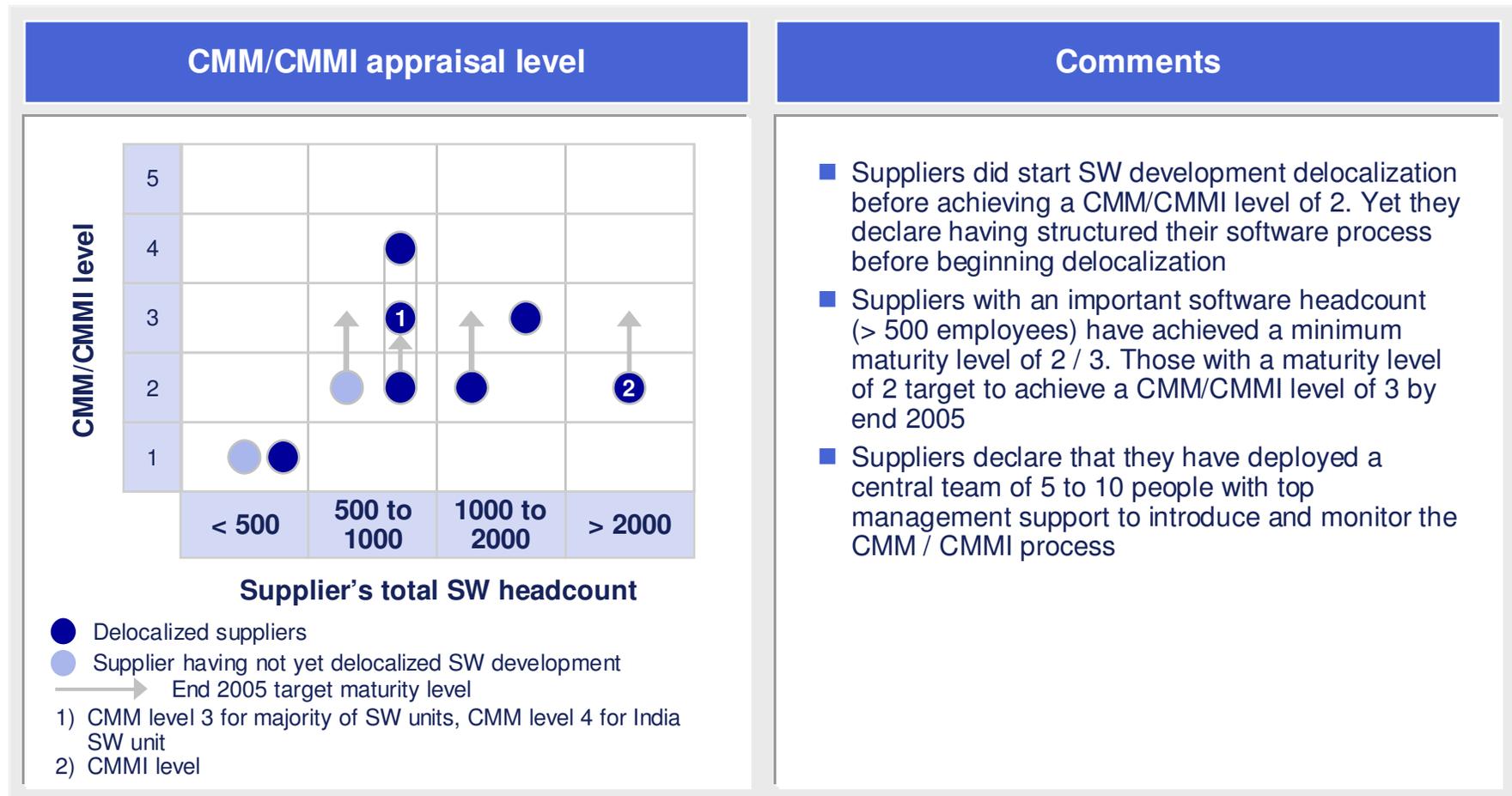




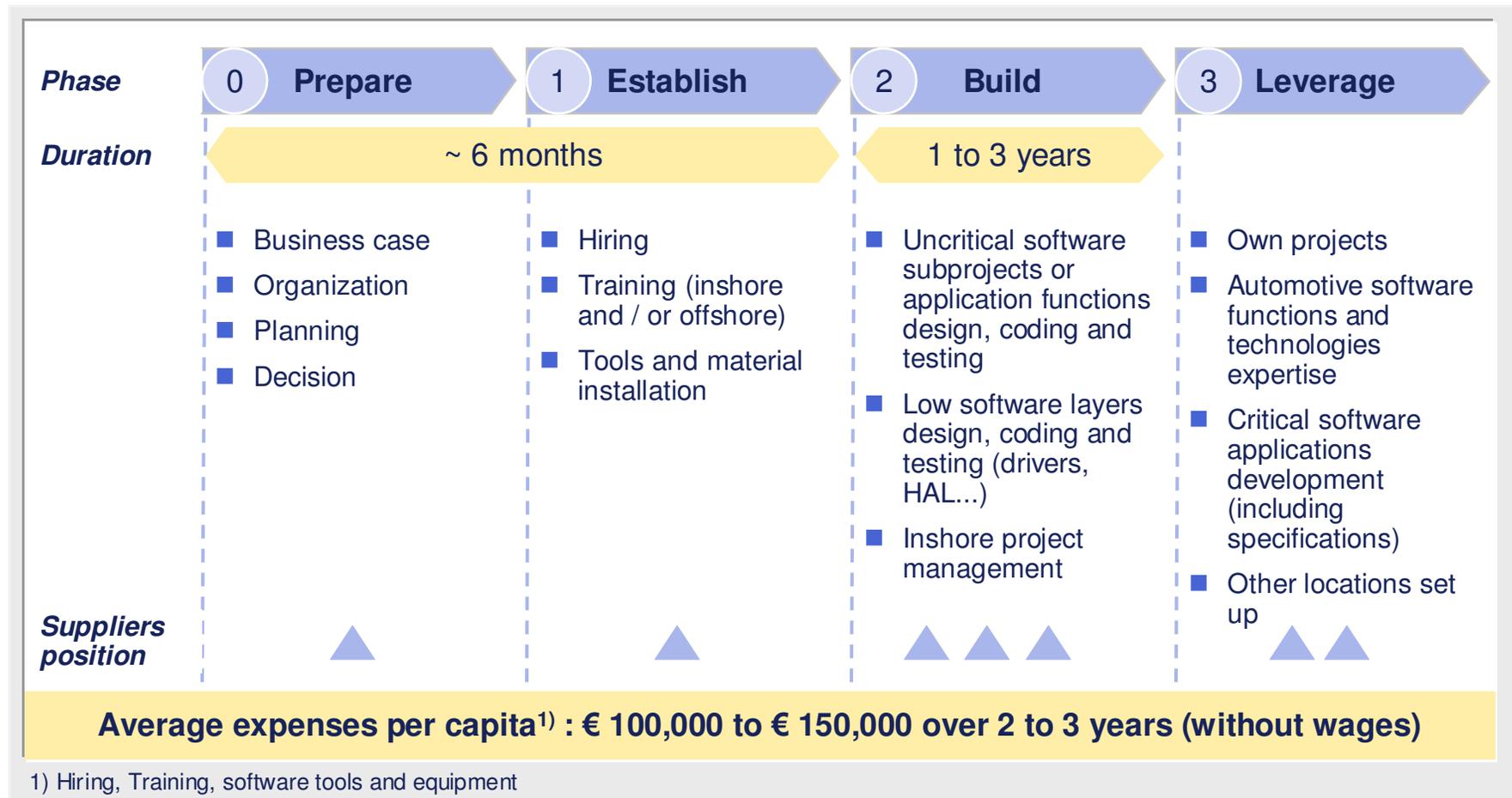
**Outsourcing is generally not the preferred "Operating Model"; operations are usually kept under one's own control**

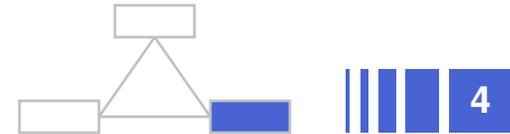
Low cost SW development operating model	Comments
<p><b>100% owned subsidiary unit</b> (5 out of 7 suppliers already operational)</p> <p><b>Planned strategic alliance / JV with low cost partner</b> (3 out of 7 suppliers)</p>  <p><b>Low cost outsourcing</b> (0%)</p>	<ul style="list-style-type: none"> <li>■ Suppliers having already initiated low cost software development operations (5 out of 7 suppliers) have set up 100% owned subsidiary units to execute their delocalized activities</li> <li>■ Before setting up their fully owned low cost software units in 1997, two benchmarked suppliers used to outsource part of their software development to global IT companies in 1992 having offshore embedded software capabilities</li> <li>■ Two suppliers having not yet delocalized their SW development, plan to start their offshore operations next year through historical local JV partners having developed embedded software capabilities (target countries: India, China, Hungary)</li> <li>■ One supplier having two SW units in eastern Europe plans to set up JVs with Asian partner to establish software units in India and China</li> </ul>

## Suppliers with more than 500 software employees have a minimum CMM/CMMI level of 2 and target to achieve a maturity level of 3 by end 2005



## All companies seem to follow a similar process to prepare and implement their delocalization

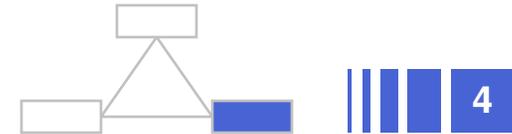




## Our study highlights a number of key success factors for delocalization of embedded software development

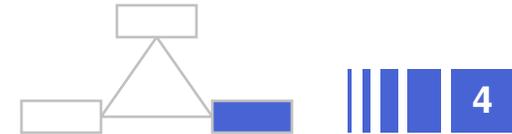
### "Delocalization of embedded software development" - Key Success Factors

- Apply sufficiently mature and controlled in-house SW development capabilities and processes (minimum CMMI level 2 or 3)
- Use standardized tools for designing and testing complex system functionalities and distributed architectures in the different phases of the V-design cycle
- Thoroughly analyze the benefits and pitfalls of different delocalization scenarios as well as lessons learned from embedded SW delocalization initiatives of other automotive and non-automotive companies
- Use a consistent approach and obtain buy-in across the organization in order to support the company's long term product and technology strategy by standardized SW platforms, tools and development processes



**Software process maturity together with effective management of people and transfer of responsibilities are among the key elements to doing it right**

Doing things right	Quotes
<p><b>Standardize and improve software process maturity before you delocalize on a large scale</b></p> <p><b>Effectively manage offshore employees</b></p> <p><b>Ensure competence and responsibility transfer</b></p> <p><b>Have a long-term vision and footprint</b></p> <p><b>Manage communication at the top management level</b></p> <p><b>Choose the right time to offshore</b></p>	<p><i>“You absolutely have to restructure the development process, receive the top management support, set up and specify methods and fixed interfaces before going there”, a US supplier.</i></p> <p><i>“Show them the value added for the company, treat them like your colleagues in Germany and strengthen their self confidence. Give them clear goals; accept their work package estimations”, a German supplier</i></p> <p><i>“You need to generate a sense of responsibility and progressively delegate high value added work to them”, a German supplier.</i></p> <p><i>“We plan to go there for 10 to 15 years not for 5 years, so you need to have a long-term worldwide footprint vision and strategy as well as a rigorous planning and organization before going”, US supplier.</i></p> <p><i>“Clear and positive top management communication must go with such an operation”, a German Supplier.</i></p> <p><i>“Hopefully we need to hire new people. We had big problems the first time we did decide to offshore software development when our company was not making money. The best time to do such things is during growth periods“, a US supplier.</i></p>



**SW development delocalization also exhibits some risks, among which employee retention and communication issues are the most critical**

Key risks & pitfalls	Quotes
<b>Employee retention</b>	<p><i>“In the beginning there was a turnover rate in India of about 20% to 25%. This means that colleagues were trained and then left the company, for example to the USA”, a German supplier.</i></p> <p><i>“Locally hired personnel does not feel committed and even linked to the company, and quit rapidly”, a German supplier.</i></p>
<b>Language miscommunication</b>	<p><i>“Language is a real problem we have with our east-European units. English is a must”, a German supplier</i></p>
<b>Workload fluctuations</b>	<p><i>“You need to maintain a high occupation rate (more than 80%) to make it economically positive”, a German supplier</i></p>
<b>Losing know-how</b>	<p><i>“If you exceed the ratio of 2 internal SW employees to 1 offshore colleague, you must install your own development organizations independent of the home locations. E.g. some business is done completely in offshore countries (acquisition, design, development and production) and with this you have to transfer 100% of know-how to the offshore locations without having a backup “at home””, a German supplier</i></p>
<b>Automotive technologies know-how</b>	<p><i>“At the beginning there were problems because the colleagues in Romania never had seen or used a central door locking system, but they are expected to develop such things”, a German supplier.</i></p>
<b>Responsiveness</b>	<p><i>“Sometimes things (specifications) change so quickly, our Indian colleagues’ response time have not always been satisfactory”, a US supplier</i></p>

## Contents

---

1	About this study
2	Executive Summary
3	Trends in Automotive Software Development
4	Levers to Increase Efficiency
<b>5</b>	<b>Arthur D. Little</b>

# Founded in 1886 by MIT Professor Arthur Dehon Little as the world's first consultancy

**Services**

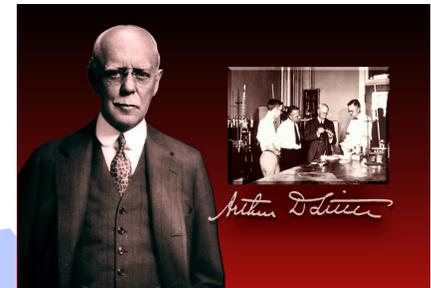
We offer comprehensive services across the full spectrum of top-management consulting (leader in technology and innovation strategy)

**Clients**

We have worked with more than 75% of Fortune 100 companies worldwide

**Global Competence**

We have a global network of more than 1000 consultants in 25 countries worldwide



**Offices:**

**Americas**

- Boston
- Buenos Aires
- Houston
- Carracas
- Mexico City
- Rio de Janeiro
- São Paulo

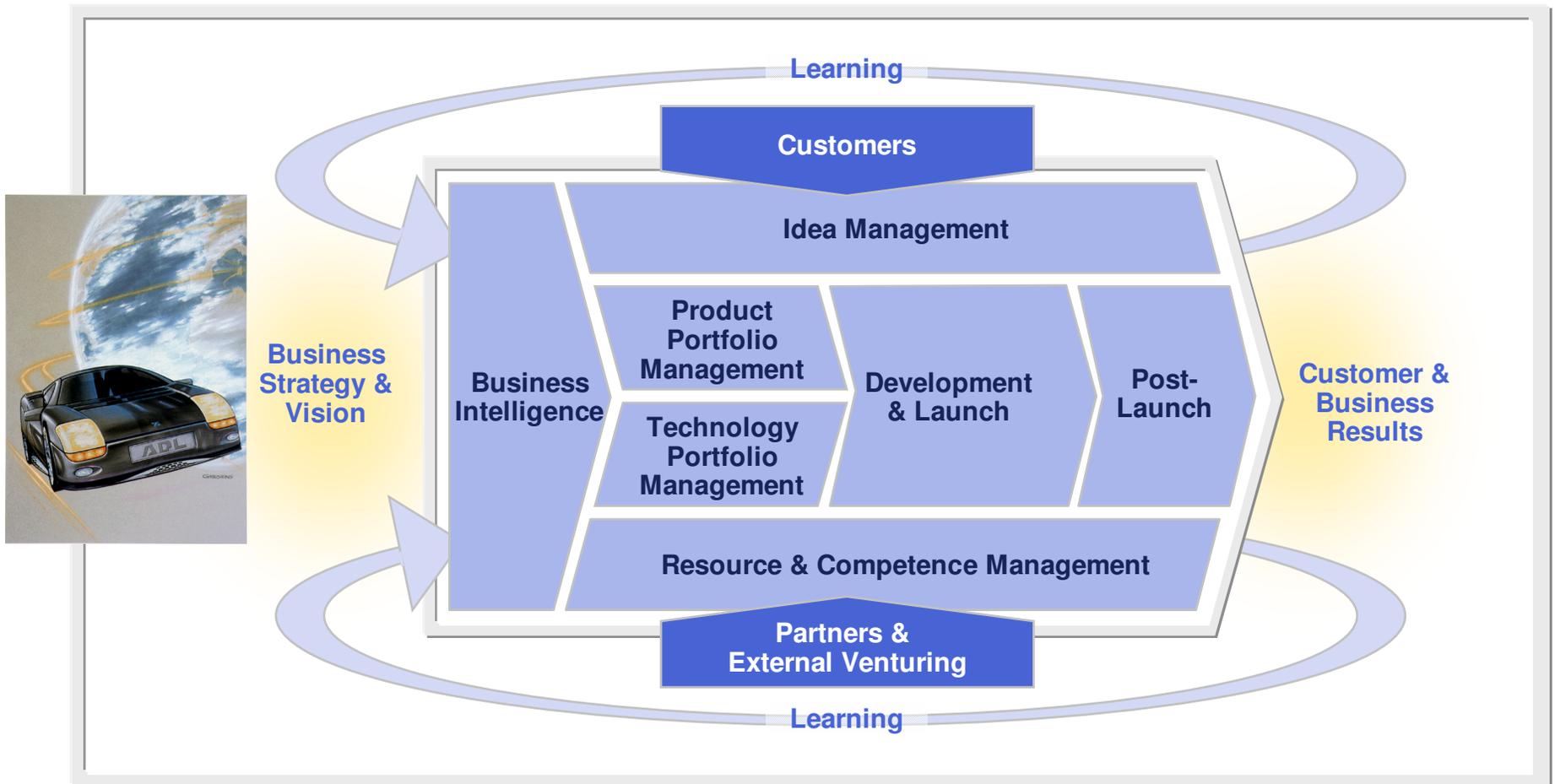
**Europe / Middle East**

- Abu Dhabi
- Berlin
- Brussels
- Cambridge
- Düsseldorf
- Gothenburg
- Lisbon
- London
- Madrid
- Milan
- Munich
- Paris
- Prague
- Rome
- Rotterdam
- Stockholm
- Vienna
- Wiesbaden
- Zagreb
- Zürich

**Asia Pacific**

- Beijing
- Hong Kong
- Kuala Lumpur
- Seoul
- Shanghai
- Singapore
- Sydney
- Tokyo

# We have industry, functional and methodological competence in all fields of technology and innovation management in the Automotive Industry



## We are working with many automotive companies worldwide

### Leverage our experience !

#### Benefits to work with Arthur D. Little

- Good industry knowledge
- Long-lasting relationships with major OEMs and Tier-1's
- Working side-by-side – from concept to implementation
- Arthur D. Little – Partner for Innovation and Growth

#### Examples of our References

