

1.

STATUS QUO OF THE AUTOMOTIVE SECTOR

Automotive is no longer in a »cycle« but in a structural transition with several pressures hitting at once. Competitive intensity has increased across segments, as new entrants and rapidly scaling Chinese OEMs drive pricing pressure, shorten time-to-market, and raise feature expectations. At the same time, the product itself has changed: vehicles are becoming software-defined systems that require stronger integration across hardware, electronics, software, data, cybersecurity, and lifecycle compliance. This shifts competitiveness away from pure industrial scale toward end-to-end system capability and iteration speed.

In parallel, the geopolitical and economic context has fragmented the previous globalization logic. Tariff uncertainty, localization requirements, and regional supply chain constraints are driving **»local-for-local«** architectures. This increases duplication, interfaces, and coordination effort – precisely where many legacy governance models are slowest. In many organizations, decision rights remain unclear between HQ and regions; steering is committee-heavy, prioritization is delayed, and accountability is diluted.

Operationally, **complexity** has become a primary **performance limiter**: high variant counts, customer-specific exceptions, inconsistent toolchains, and overlapping initiatives create rework and absorb engineering capacity. Financially, investment needs remain high [technical disruption, restructuring], while margins face pressure from discounting, input cost volatility, and quality risks during accelerated platform transitions.

The Talent dynamics amplify these constraints. Competitive advantage increasingly depends on scarce software delivery, systems architecture, test automation, and AI-enabled engineering productivity. **Blunt restructuring often removes exactly the capacity and motivation required for transformation**, triggering a quiet decline in speed and capability. The result is a strategic contradiction: cost programs are launched to protect competitiveness but executed in ways that weaken the ability to innovate.

2.

THE PAST: LINEAR CYCLES – AND WHY THE RHYTHM KEPT SHIFTING

Historically, many automotive organizations followed a sequencing model: **innovation during growth, restructuring during downturns**. This worked tolerably well when product cycles were long, regulation and competition evolved gradually, and global scale rewarded centralization. The organization learned a rhythm of »expand« versus »freeze«.

Over the last 15 years, that rhythm has become unreliable. Technology disruption turned mandatory [SDV, cybersecurity, data compliance]. Competitive pressure accelerated, making delayed investment strategically costly. Regulation and geopolitics introduced constant change rather than discrete waves. Complexity rose structurally through parallel powertrains, variant growth, and fragmented digital stacks.

The consequence: in restructuring phases, companies paused or diluted innovation when it needed acceleration; in innovation phases, portfolios expanded without stop-rules, causing initiative overload and weak execution. The environment stopped allowing clean separation. The limitation of the past model is not intent, but the assumption that restructuring and innovation can take turns. Under current conditions, they must coexist structurally.

THE NEW MANAGEMENT: RESTRUCTURING & INNOVATION AT THE SAME TIME

Sequential transformation is no longer viable in automotive. The old logic – innovate in up-cycles, restructure in down-cycles – breaks under permanent disruption, regional fragmentation, and accelerating competition. Competitiveness now requires two parallel cycles managed as one operating system: a »Performance Engine« that removes structural complexity and frees capacity, and a »Future Engine« that converts uncertainty into scalable products and measurable growth. The differentiator is not communication, but governance: one portfolio, explicit trade-offs, stop-rules, and protected critical capabilities.



DANIEL NIEDERBERGER
Managing Director
MBA TUM | MSc. Frankfurt School

3.

THE FUTURE: PARALLEL CYCLES AS A COMPETITIVENESS REQUIREMENT

The next decade rewards leaders who operate two engines in parallel, integrated through one operating system.

3.1 Performance Engine: restructuring as structural simplification

The goal is not short-term austerity; it is **complexity removal** that permanently frees capacity. Typical levers:

- Variant and portfolio rationalization; disciplined exception management
- Platform / interface standardization (incl. software and data models)
- Organizational simplification (layers, decision rights, faster escalation paths)
- Toolchain consolidation to reduce rework and shadow IT / OT stacks
- Quality stability to reduce firefighting and warranty risk

Outputs must be measured beyond cost: cycle-time, rework, engineering throughput, decision lead time.

3.2 Future Engine: innovation as measurable capability building

Innovation must move from scattered pilots to a disciplined pipeline:

- 2–3 explicit growth bets with outcome KPIs (time-to-market, adoption, margin)
- Software delivery model enabling iterative releases and integration discipline
- AI-enabled productivity where it reduces engineering / operations cycle time (requirements, testing, validation, documentation, diagnostics)
- Proof-based scaling: clear »prove > industrialize« criteria, and stop rules

Protection is essential, so is evidence. Innovation may fail fast, but it must not drift.

3.3 Integration: one portfolio, one decision cadence

Most failures come from two competing systems: a finance-driven restructuring office and a strategy-driven innovation program. The »new management« integrates both into:

- One transformation portfolio containing all initiatives (incl. regional / functional »shadow« programs)
- One accountable owner per initiative, with outcome KPIs and stop criteria
- One cadence designed for decisions, not reporting
- Capacity allocation treated as the real budget (people / time, not only money)

TOP-MANAGEMENT ACTION LIST

- 1. Define the two-engine model in one page**
[scope, rules, metrics, ownership]
- 2. Build one integrated transformation portfolio within 1 months;** remove duplicates; assign owners; define outcomes and stop rules
- 3. Enforce »start = stop«** to prevent initiative overload and restore focus
- 4. Restructure by reducing complexity before cutting muscle** [variants, exceptions, interfaces, toolchains]
- 5. Protect critical capabilities explicitly**
[systems / software, test automation, AI productivity, platform architects, key program leaders]
- 6. Turn governance into a decision machine:**
weekly unblocking; monthly stop / scale; quarterly resource shifts; decision log mandatory
- 7. Measure speed and throughput, not only cost**
[decision lead time, rework, time-to-market, portfolio burn-down, quality stability]
- 8. Clarify HQ-region decision rights** for local-for-local vs global-for-scale; align incentives to end-to-end outcomes
- 9. Select 3–5 AI productivity use cases and scale;**
kill pilots without measurable impact
- 10. Personally sponsor stop decisions and trade-offs –**
if stopping is delegated, nothing stops