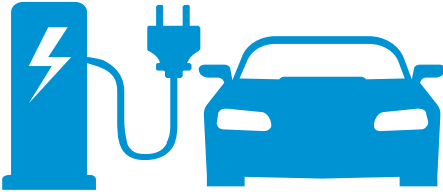McKinsey&Company

Productivity drivers in Automotive SW development
Numetrics R&D Analytics

Contents

The shift from "hardware performance" to "software experience" will continue at full speed

Trend	Description	Impact on importance of software
Luxury feature proliferation	▪ The share of premium cars in the compact class has been rising in recent years ▪ New premium compact cars like the Audi A1 come with a full infotainment offer like mid-range cars ▪ More and more premium features are being included in entry level models	▪ SW enabled features become more and more mainstream ▪ Advanced SW features are needed to maintain a low-cost differentiation for premium cars
Electric vehicle	▪ Share of EV[1] could reach 50% of sales by 2030, depending on regulatory factors, lower battery costs, range and charging station availability ▪ Several countries are providing incentives and deploying EV charging infrastructure to drive adoption	▪ Increase of interconnected electronic components (e.g. battery control unit) and the complexity of software needed to manage them
Autonomous and connected cars	▪ Several players are developing autonomous cars ▪ The level of driverless features such as self-parking and automatic braking are continuously increasing ▪ The number of networked cars will rise 30 percent a year for the next several years; by 2020, one in five cars will be connected to the Internet	▪ Software is taking over some of the mission-critical functions and will eventually replace human drivers

1 Electrified Vehicles, including hybrid, plug-in, battery electric, and fuel cell

With increasing functionality driven by SW, quality issues become a major concern and have significant impact

	Issue	Impact
FIAT CHRYSLER	• SW security flaw made cars vulnerable to hackers	• Recall of ~1.4M cars
	• SW incompatibility between EV control unit and battery control module may cause propulsion system to shut down	• Recall of ~5,600 Electric Vehicles
TOYOTA	• SW flaw may cause the hybrid system to shut down while driving	• Recall of 1.9M hybrid cars
	• SW flaw may cause VSC, ABS and traction control functions to shut down	• Recall of ~260,000 Vehicles
HONDA	• Flaw in the continuously variable automatic transmission software may subject the drive pulley shaft to high stress	• Recall of 143,000 cars in the US
	• Cell voltage sensor incorrectly interprets electrical noise and may cause a sudden loss of power	• Voluntarily recall of 6,786 hybrid cars
LAND ROVER	• SW flaw may cause vehicle doors to be unlatched	• Recall of 65,000 cars
	• Flaw in the anti-lock braking system may disable stability and control safety systems	• Recall of 2,687 SUVs
Mercedes-Benz	• Flaw in engine control unit SW may cause engine to stop while stopping at a traffic light	• Recall of ~3,000 cars

Schedule slips due to SW complexity and inaccurate project plans can also impact revenues and market position

	Issue	Impact
HYUNDAI	▪ The integration of Apple Carplay, which should be included in Hyundai Sonata is delayed more than two years, due to integration problems. The features were announced as a major selling point in 2013/2014	▪ Lagging behind OEMs such as Chevrolet and Honda that have already introduced the Carplay
GM	▪ General Motors delayed its launch of the Super Cruise semi-autonomous driving feature that would have debuted on Cadillac CT6 ▪ Specific reasons were not announced, however it may include problems regarding combination of software / hardware	▪ Market position as a technology leader (e.g. Vs. Tesla's AutoPilot semi-autonomous capabilities
FIAT CHRYSLER	▪ In 2015, new Jeep Renegades have not been given to dealers as planned due to software problems ▪ The launch of the Jeep Cherokee had to be put off for several months due to the software that controlled its nine-speed automatic transmission and all-wheel-drive system	▪ Reputation and revenue hit due to delayed time-to-market
AUDI	▪ Audi has postponed by one year the introduction of the A8 luxury sedan in order to make it the brand's first car capable of autonomous operation	▪ Delays in the introduction of the new A8, aimed at beating the segment's leader, Mercedes' S-Class

Tier 1 automotive suppliers face common challenges in software development

NOT EXHAUSTIVE

What SW is developed?	**Connected car and electrification** trends broadening scope of software far beyond basic firmware and critical system integration **Constantly evolving / unclear requirements** from OEMs and marketing causing significant churn
How is SW developed?	- Using **processes largely tailored to HW with nascent SW development and integration processes** and synchronization milestones in place - Using complex software and verification tools which leads to **quality issues** due to lag of verification/validation processes in place - Increasing collaboration with 3rd party content / platform providers, OEMs and emerging cloud / OTA applications fosters **integration challenges** - With a **low level of re-use** because of **limited platform-driven architecture**, knowledge sharing and centralized SW management
Where is SW developed?	**Software development teams** largely based in traditional HW development centers where SW talent is not readily available **Increasing use of outsourced** SW vendors in China, India and eastern EU
How is SW development enabled ?	- Strong **"get it done" mindset** limits focus on creating robust processes, practices and tools - Limited automated and mandatory **tracking of main SW quality KPI's** (e.g., test coverage) for all projects **Culture and work environments** not suited to attracting and retaining the best SW developers

Contents

Analysis of productivity drivers and trends have revealed several insights into Automotive software development

1 There is an **unsustainable, widening gap between automotive projects' complexity and R&D productivity** levels

2 There is a significant **gap between automotive top and bottom performers** in productivity, development throughput and released quality

3 **Requirements, specifications and test plans are more granular in automotive** compared to industry

4 Outgoing automotive **SW quality is strongly affected by requirements stability,** especially for mission-critical applications

5 Number of **defects found during development is a strong leading indicator of outgoing automotive SW quality**

6 **Reuse improves quality**, especially in mission-critical applications, **but only when it exceeds a threshold**

7 Projects with **low reuse leverage (<30%) are also likely to suffer from significant schedule slips**

8 **Platform maturity impacts automotive** productivity, and mission-critical especially, **more than non-auto productivity**

9 **Fragmentation of automotive SW teams can negatively impact development productivity**

① There is an unsustainable, widening gap between complexity and productivity levels that needs to be addressed

Productivity relative growth over time

Complexity relative growth over time
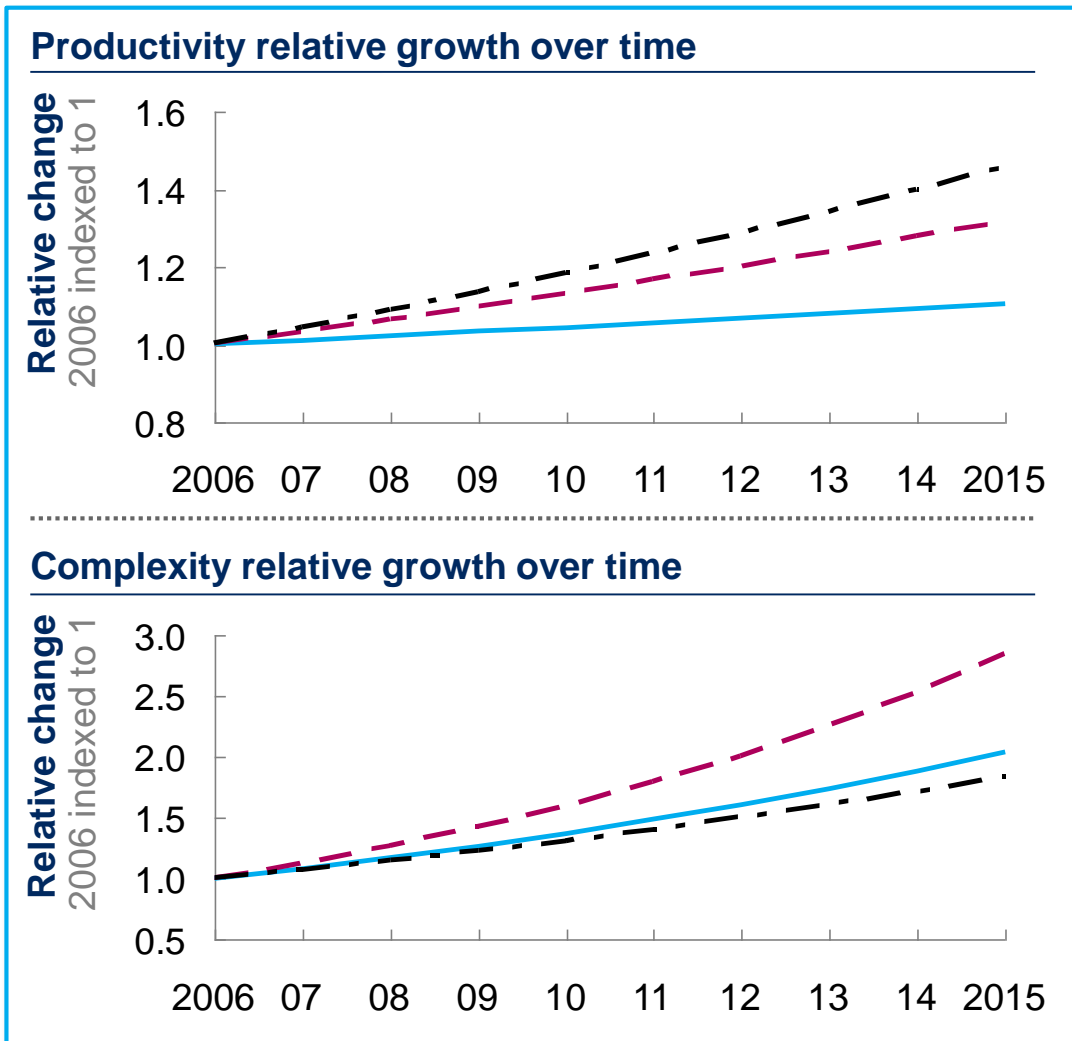
Legend:
- Infotainment
- Mission critical
- Industry

- Over the past 10 years for automotive:
  - **Productivity**, and especially infotainment has grown **slower** than the industry
  - **Complexity**, and especially mission-critical has grown **faster** than the industry
- The result is a **widening gap between complexity and productivity** levels, that is likely to **result in performance and schedule challenges**

2 There is a significant gap between automotive top and bottom performers in productivity, development throughput and quality

Average indexed to 100

Legend:
- Bottom quartile
- Average
- Top quartile

Productivity

Complexity units per man week

- Bottom quartile: 53
- Average: 100 (−38%)
- Top quartile: 152 (+52%)

Development throughput

Complexity units per week

- Bottom quartile: 41
- Average: 100 (−59%)
- Top quartile: 230 (+130%)

Quality

Residual design defects[1]

- Bottom quartile: 300 (+200%)
- Average: 100
- Top quartile: 29 (−71%)
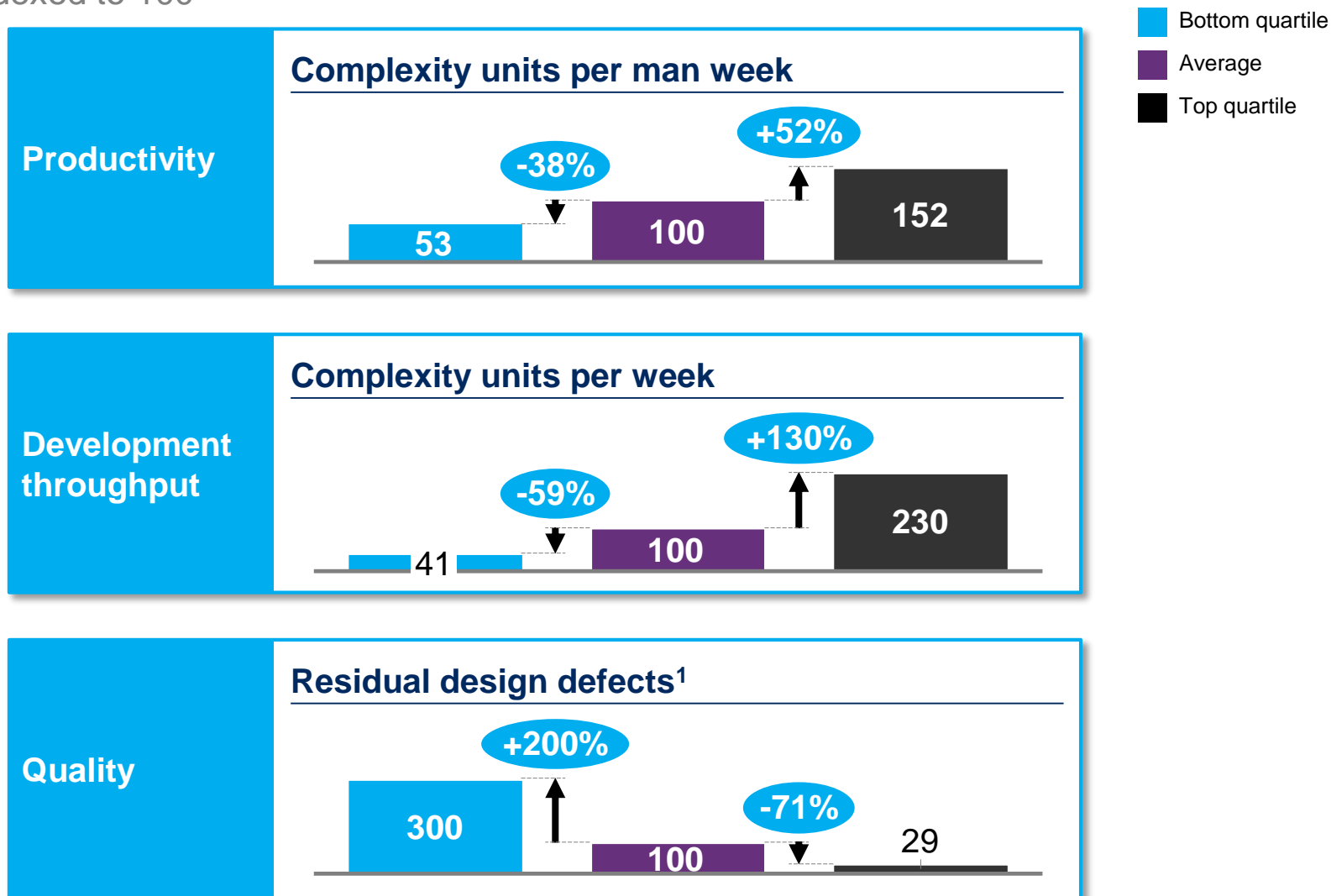
1 Known residual defects per new lines of code, at release

Requirement specifications and test plans are more granular in automotive compared to industry

③

Quality Metrics (non-auto indexed to 100%)

Legend:
- Infotainment
- Mission-critical
- Non-auto

New tests per new LOC[1]
- Infotainment: 196%
- Mission-critical: 163%
- Non-auto: 100%
- +96%

LOC[1] per Requirement
- Infotainment: 47%
- Mission-critical: 56%
- Non-auto: 100%
- -53%
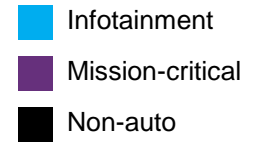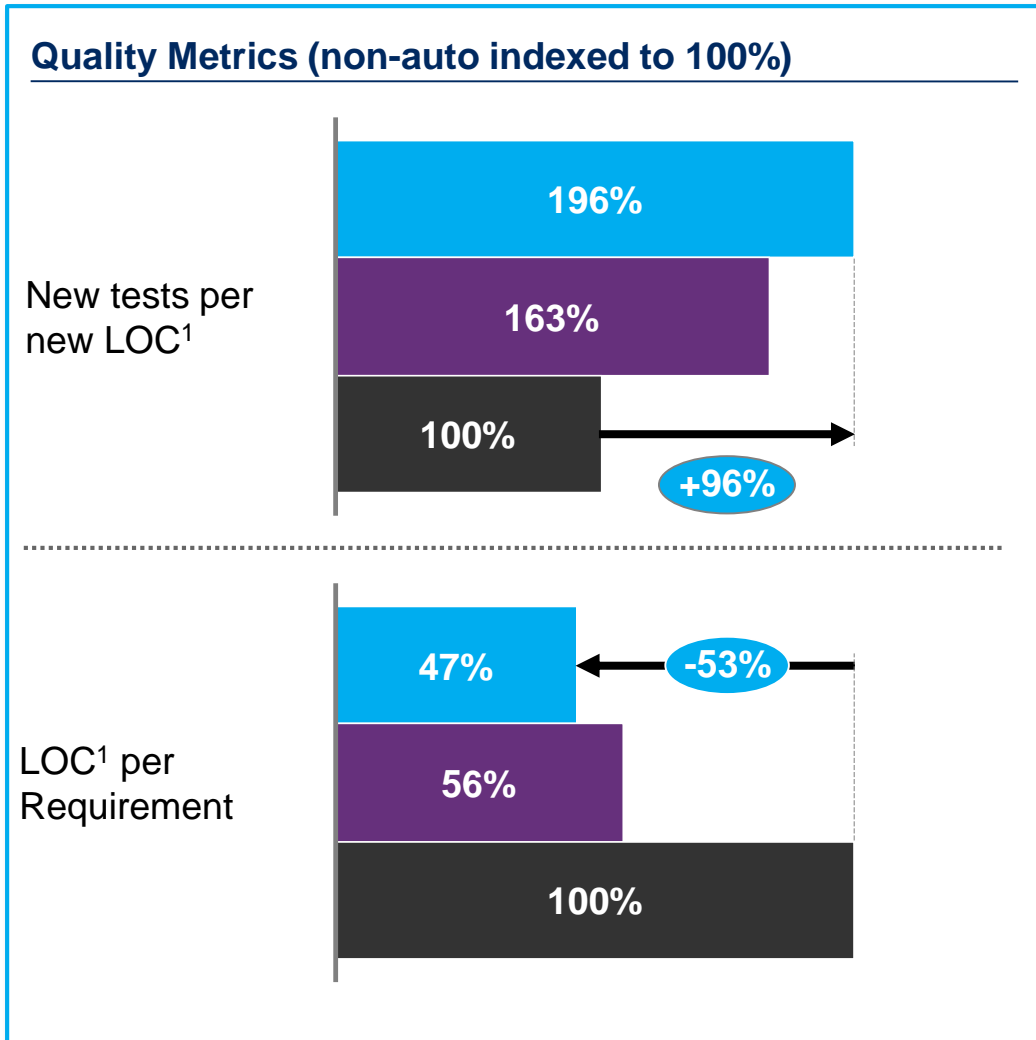
- There are **more new test case per LOC in automotive** SW Vs. non-auto
- Infotainment may introduce more feature diversity, **requiring even more new test cases**
- Automotive SW **requirements are specified at a more granular level**, requiring fewer LoC to implement
- Higher level of granularity can improve coding speed but **also increases the need for requirement & test traceability**

1 LOC – Line of Code

④ Outgoing automotive SW quality is strongly affected by requirements stability, especially for mission-critical applications

Residual Defect Density
Number of defects per kLOC[3]

Functional requirements changed[1]
% of total requirements
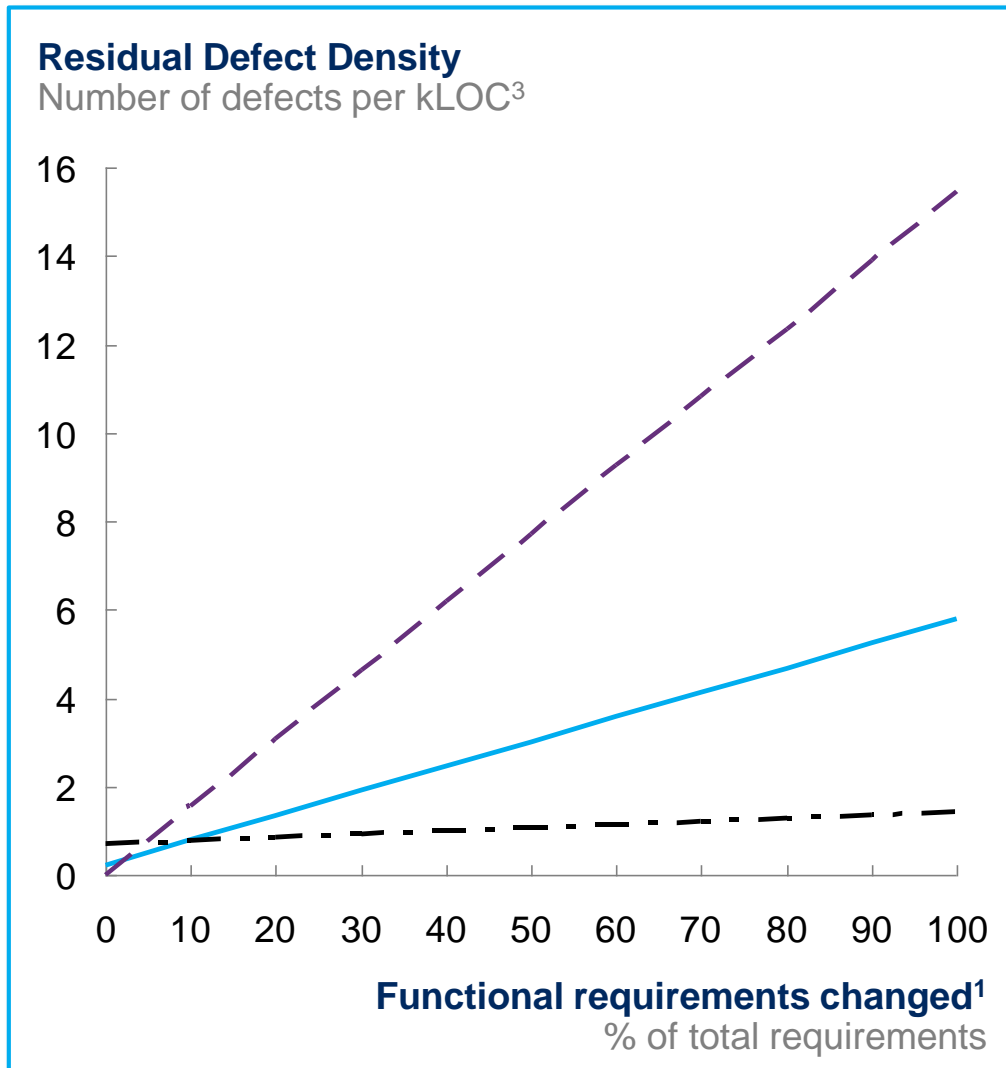
Legend:
- Infotainment
- Mission-critical
- Non-auto

- Spec changes have a larger impact on the known defects at release Vs. non-auto, especially in mission critical applications

- Impact of specification changes is higher in automotive probably due to the granularity level and complexity of requirements

- Disciplined requirements change management is a key to improving auto SW quality

1 Includes: additions, modification and removal of features

5 Number of defects found during development is a strong leading indicator of outgoing automotive SW quality

Infotainment
Mission-critical
Non-auto

Residual Defect Density[2]
Number of defects per kLOC[3]

Design Defect Density[1]
Number of defects per kLOC[3]
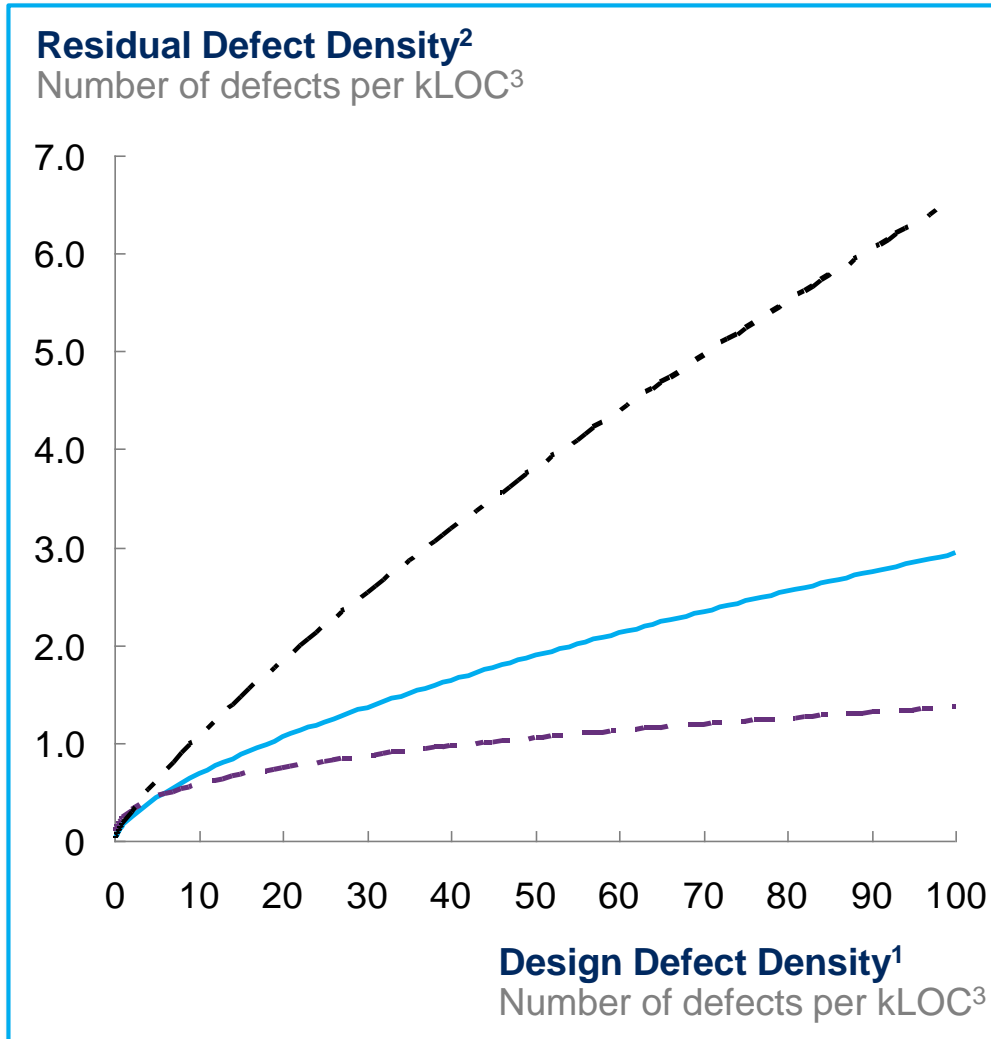
- Design defect density (DDD) is a strong leading indicator of released software quality

- Automotive, and especially mission-critical SW seem to be doing a better job than average in finding and fixing in-development defects

- A likely reason for this difference is the higher level granularity of requirements and test cases in automotive compared to other industries

1 Known defects at release   2 Defects found during development   3 LOC – Line of Code

6 # Reuse improves quality, especially in mission-critical applications, but only when it exceeds a threshold

Residual Defect Density[1]
Number of defects per kLOC[3]

Reuse Leverage[2]
% of Design Complexity avoided due to reuse[3]

% of projects in each category
Percent

Reuse Leverage[2]
% of Design Complexity avoided due to reuse[3]

Legend:
- Infotainment
- Mission-critical
- Non-auto
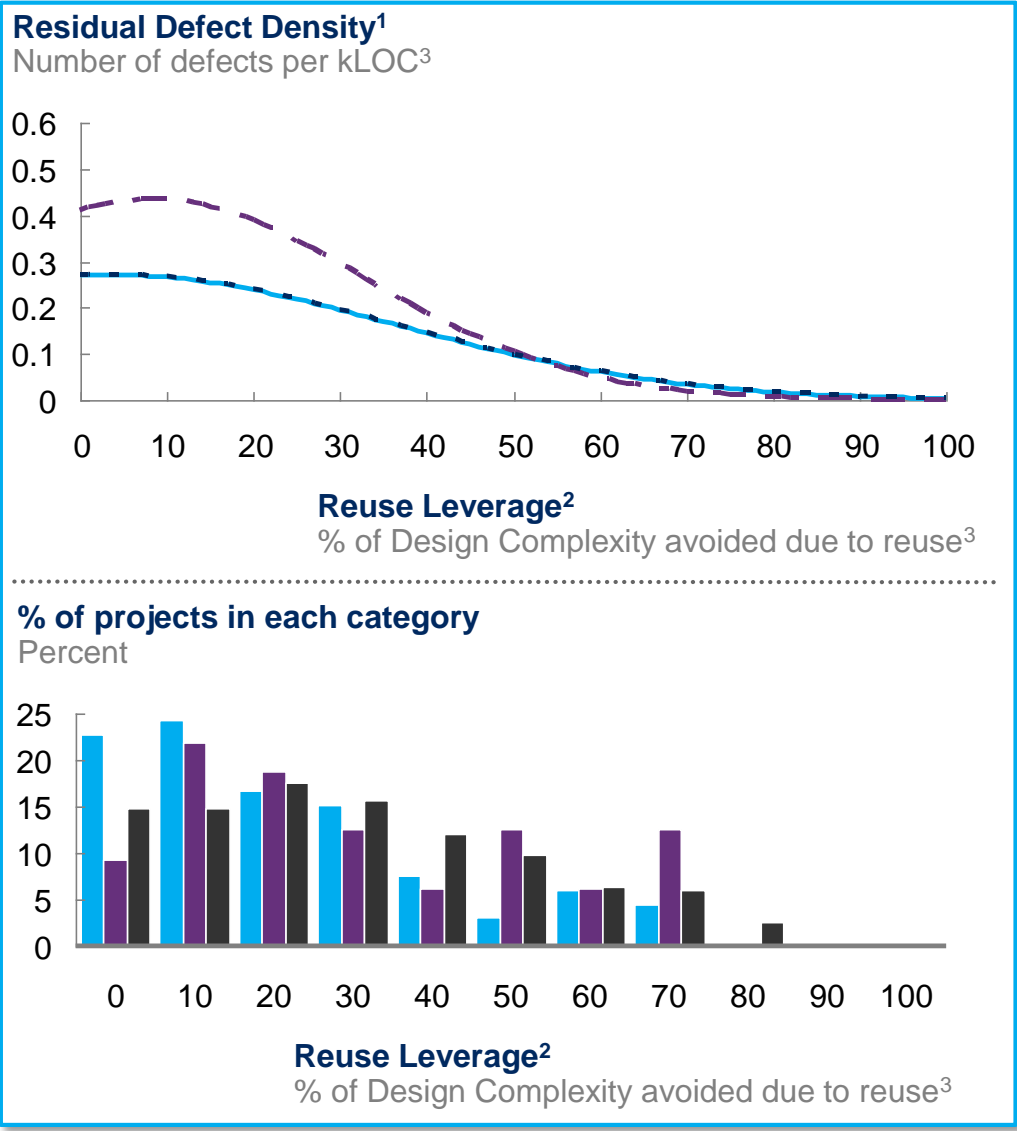
- Increased reuse is more impactful for mission critical, compared to infotainment and non-auto applications

- Reuse below ~20% has minimal quality impact for infotainment and actually introduces more defects for mission critical applications

- Infotainment has low reuse levels with ~25% of projects below 10% reuse level

- Mission critical has higher reuse, which improves quality significantly

1 Defects found post release
2 Reuse leverage is a normalized metrics representing the effort saved due to reusing code and test cases

7 Projects with low reuse leverage (<30%) are also very likely to suffer from significant schedule slips

Automotive schedule slip
Percent over original plan

Reuse Leverage[1]
% of Design Complexity avoided due to reuse[3]
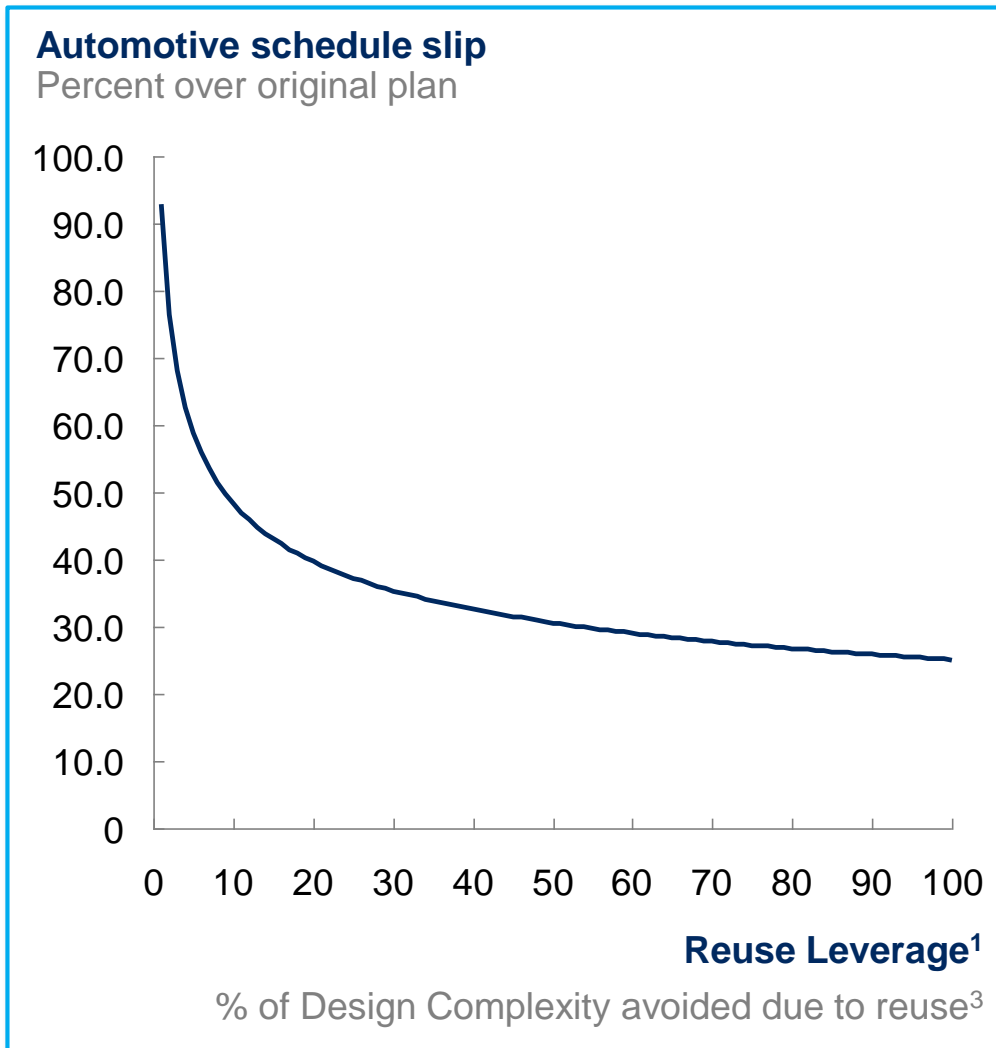
- Reuse leverage of below 30% is highly associated with increasing probability of schedule slip with up to 100% slip for no reuse

- Marginal impact of reuse on schedule slip diminishes as reuse level increases

- Brand new automotive projects are very likely to be underestimated for duration

1 Reuse leverage is a normalized metrics representing the effort saved due to reusing code and test cases

Platform maturity impacts automotive productivity, and mission-critical especially, more than non-auto productivity

8

Productivity Vs. HW maturity level
Percent, indexed to productivity with available and mature HW platform

Legend:
- Available & Proven (blue)
- Prototype-only (purple)
- Not Available (black)

Non-Auto: 100%, 98%, 89 (-11%)
Infotainment: 100%, 83%, 73 (-27%)
Mission-critical: 100%, 83%, 56 (-44%)
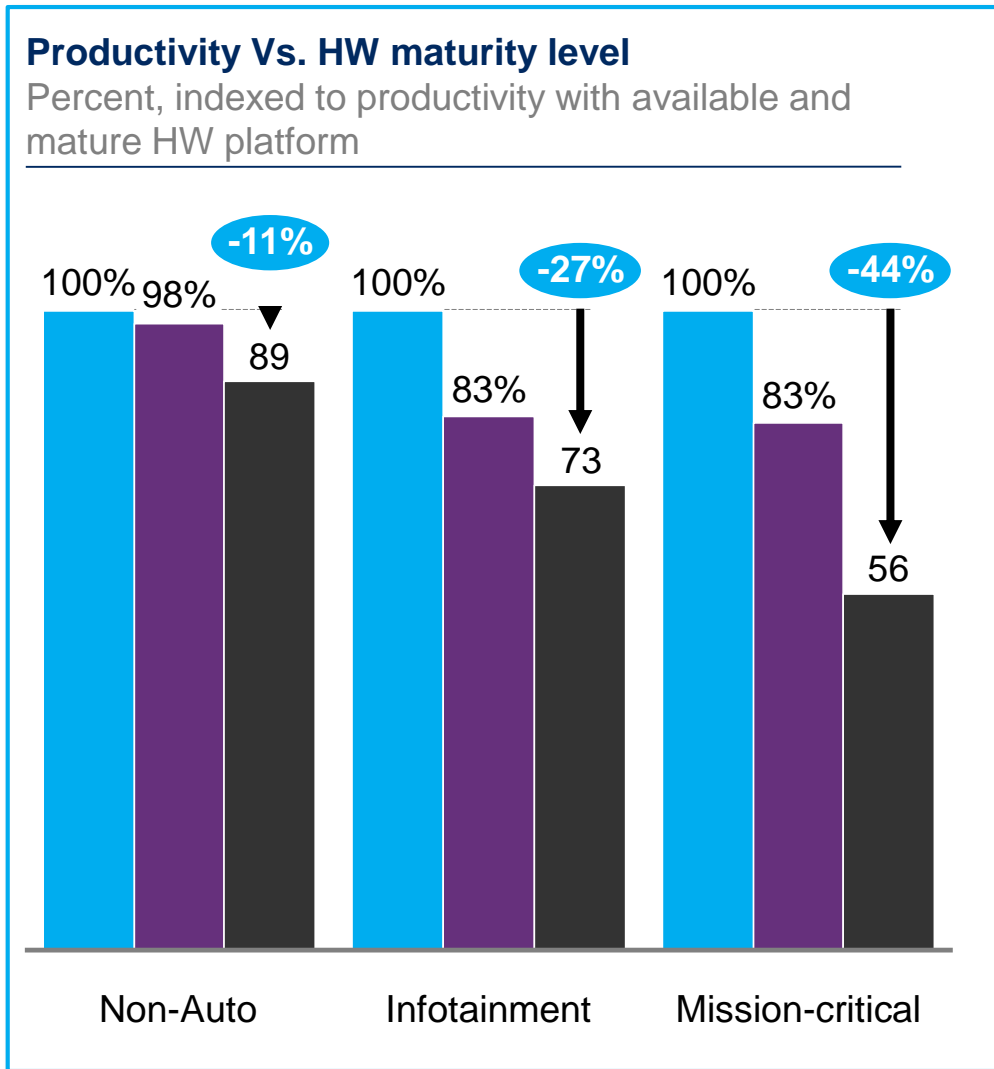
- Automotive SW productivity is more strongly impacted by hardware maturity than the rest of the industry

- Automotive, and specially mission-critical productivity are impacted much more by prototype or non-available HW

- Potential reasons could include developing more HW specific code in automotive Vs. using APIs in non-auto, level of late stage HW spec changes and availability of HW simulators and testing tools

9 Fragmentation of automotive SW teams can negatively impact development productivity

-5% compound loss/site

-11%

100%
89%
83%
79%
76%
74%
72%

1 2 3 4 5 6 7

Number of R&D sites

Indexed productivity
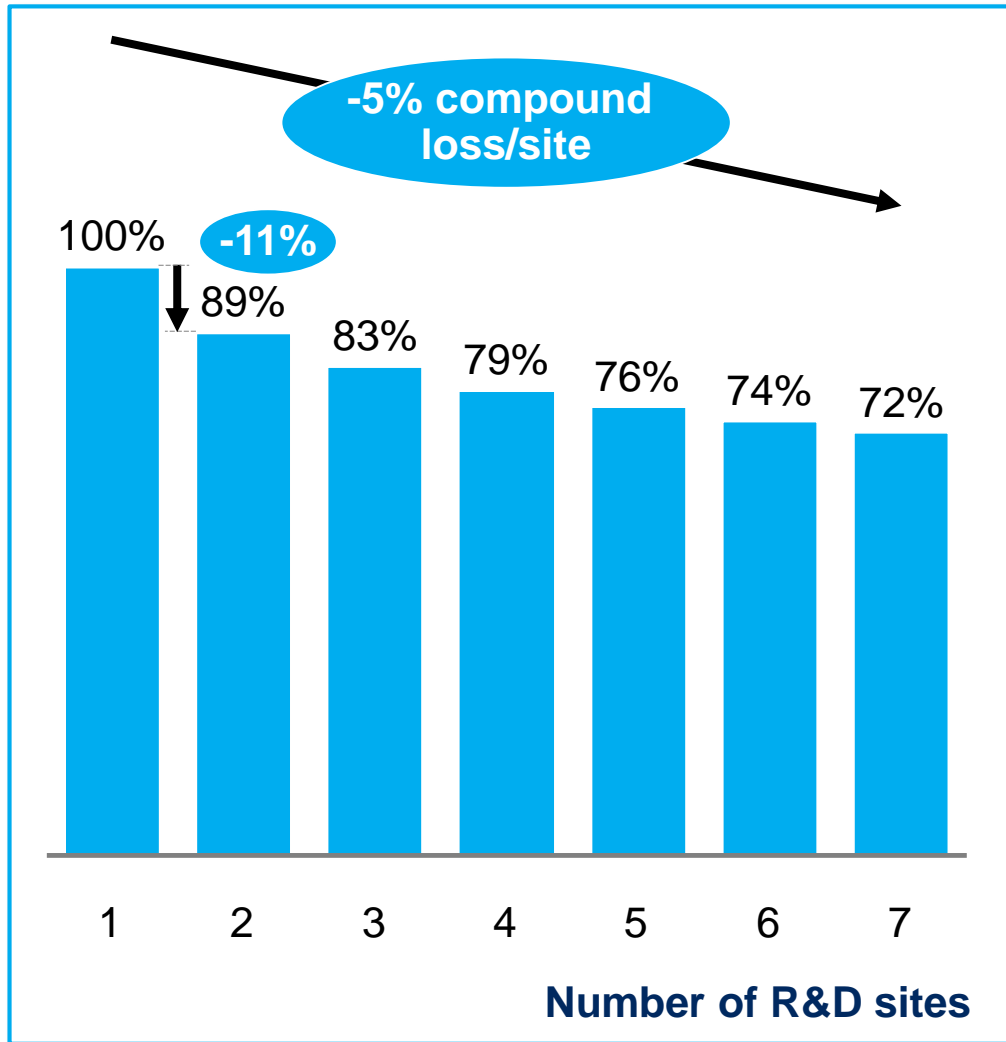
- Separating automotive SW development across multiple sites introduces inefficiencies and can reduce overall productivity
- Biggest productivity drop[1] of ~11% when introducing a second site, with lower impact from additional sites

1 Productivity drop is partially due to a larger team size which is correlated with multiple R&D locations

Contents

Numetrics is a SaaS-based analytics solution that enables rapid improvements in embedded SW and IC development

Numetrics
By McKinsey

Proven complexity measurement method	Large industry database of peer projects	Established analytics platform

Proprietary complexity

algorithm successfully

applied in **>400**

companies

PENDING PATENTED PENDING
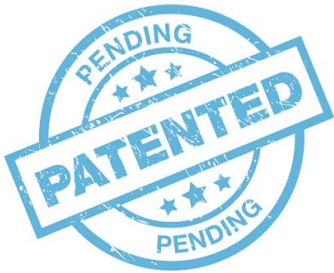
2,000+ IC projects

1,700+ software projects

40+ industry segments

50+ operating systems

20+ programming languages

Industry Benchmarking

Project Planning & Estimation

Root Cause Analysis / Productivity Diagnosis

Portfolio & Resource Planning

Numetrics' analytics engine is based on a proprietary "design complexity" model that normalizes productivity across projects

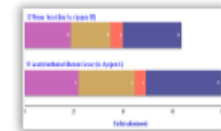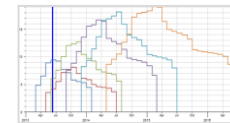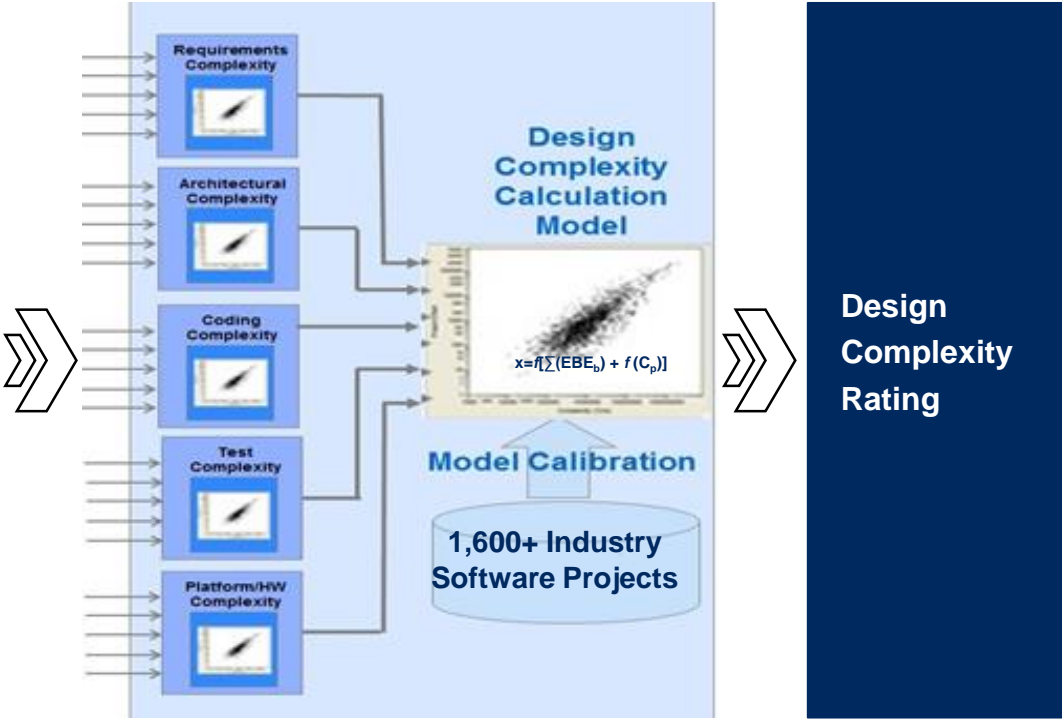**Design/development complexity:**
- A metric representing the total amount of project effort the average design/development team in the industry would expend on the project – quantifies the true, normalized output of the design team
- The complexity model fully takes into account the stochastic nature of product development, which enables the predictive analytics engines to reliably estimate schedule & resource requirements and perform meaningful comparisons of performance metrics across different projects/designs

Complexity inputs

Software Complexity Measures
- Customer requirements
- Functional requirements
- Test cases
- Use cases
- Test types
- Lines of Code
- Architectural layers
- Number/type of components
- Reuse
- Programming language(s)
- Number of variants
- Real-time content
- Available storage space
- Number of platforms
- Platform maturity

Requirements Complexity

Architectural Complexity

Coding Complexity

Test Complexity

Platform/HW Complexity

Design Complexity Calculation Model

$x = f[\sum(EBE_b) + f(C_p)]$

Model Calibration

1,600+ Industry Software Projects

Design Complexity Rating

Numetrics offers performance benchmarking, root cause analysis and project planning (predictive analytics)

What is Numetrics? — **SaaS-based** R&D predictive analytics platform based on a **patented complexity algorithm** to provide:

Performance benchmarking	**Root cause analysis**	**Project planning**

Where can Numetrics be applied?

- **Software (Embedded and application):**
  - **Verticals**: Automotive, Telecom, Financial, Medical devices, Industrial controls, Aerospace & Defense, etc.
  - **OS**': Android, IOS, Linux, Microsoft, Wind River, TI, etc.
  - **Platforms**: ARM, MIPS, Broadcom, Freescale, IBM, Microchip, Renesas, Samsung
- **Semiconductors (ICs):** Across segments, including Analog, Mixed signal, Memory, SOC, FPGA, IP, RF

Performance benchmarking – Creates a productivity baseline to enable internal and industry benchmarking

Performance benchmarking

Create a project-level productivity baseline based on recent projects, and benchmark across multiple dimensions against a database of ~2,000 IC and 1,600+ SW projects

Sample outputs

■ Industry peers   ■ Client projects

Project duration Vs. Design complexity

Productivity Vs. Team size
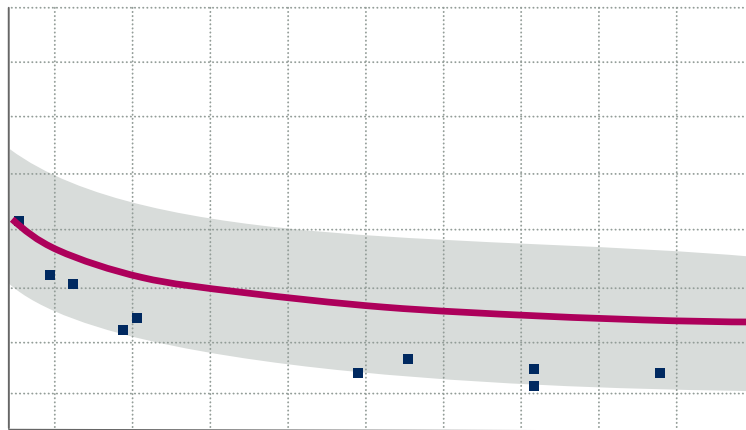
Performance benchmarking – Wide range of metrics can be benchmarked

NOT EXHAUSTIVE

■ Client Software Projects   ☐ Band containing 50% of industry peers

How fast can we deliver SW?

Duration vs. Complexity
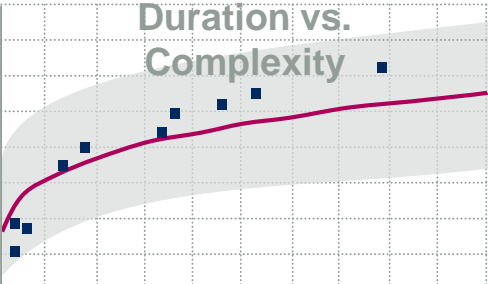
How many people do we need?

Team Size vs. Complexity

How efficient are we?
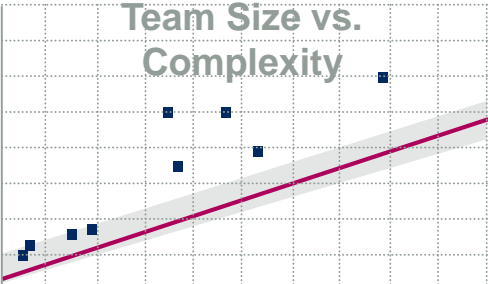
Productivity vs. Team Size
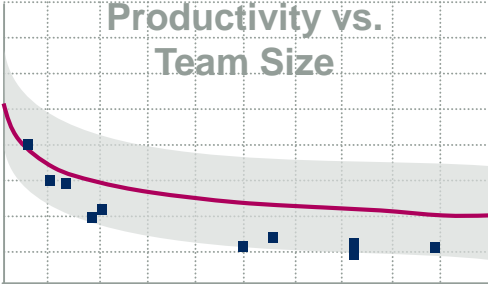
Is our verification strategy effective?

Residual vs Design Defects
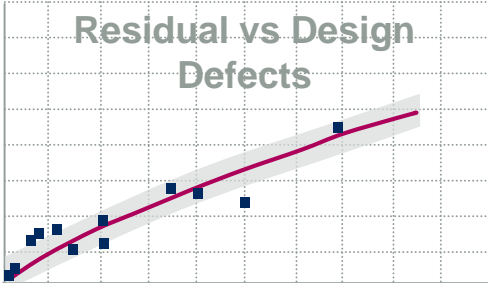
How granular are our requirements?

Tests/Requirement vs. LOC/Requirement

How cost competitive are we?

Cost efficiency vs. Productivity
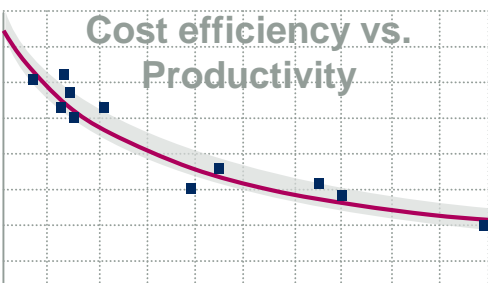
Root cause analysis – Analyzes industry database (best practices) to identify likely causes of low productivity

Root cause analysis

Use analytic tools to find root causes and drivers of low performance, and compare to industry best practices to determine recommended course of action

Sample outputs

Poor spec stability caused significant schedule slip

Schedule slip
Percent over plan

53% — N=10 (Low)
32% — N=6 (Average)
20% — N=7 (High)

Specification stability
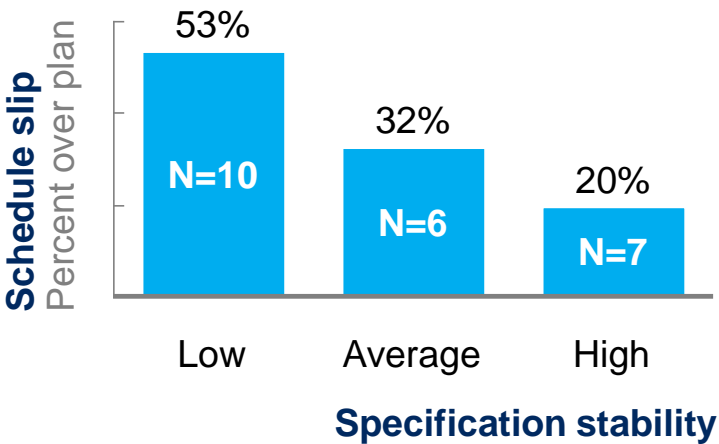
Insufficient effort during design phase caused higher test effort

■ Client projects   ■ Industry Best-in-Class

% of total effort

Role	Client	Industry
Mngmt	8	10
Req	4	10
Design		15 (-67%)
Coding	30	29
Test	42	24 (+75%)
Documentation	6	7

Role
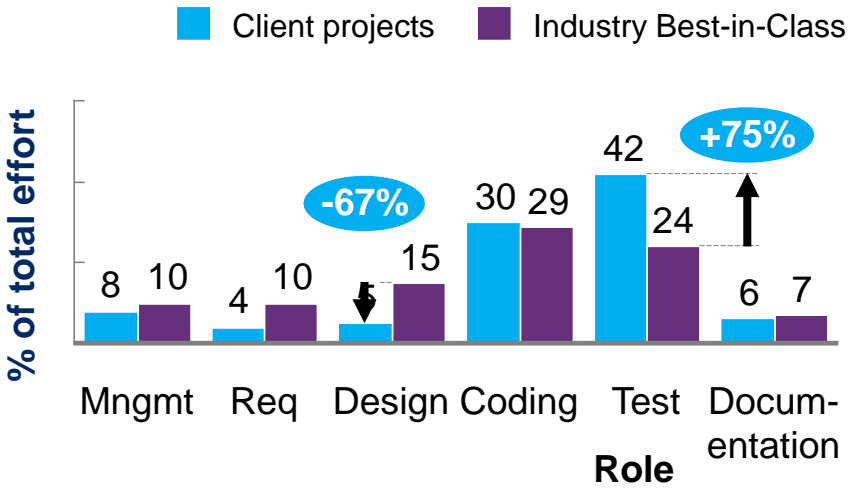
Project planning – Predictive analytics generates robust project plans (resources, schedule) to identify time-to-market risks

Project planning and risk assessment

Use predictive analytics to provide better transparency to schedule and required resources at the project's outset and assess schedule risk due to unrealistic productivity assumptions

Sample outputs

Predicted staffing requirements by role and project phase
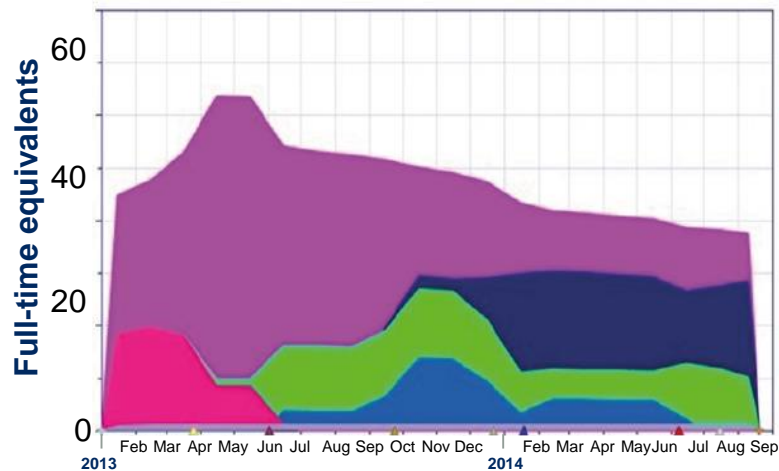
Schedule risk due to unrealistic productivity assumption

Unrealistic productivity assumed for new project
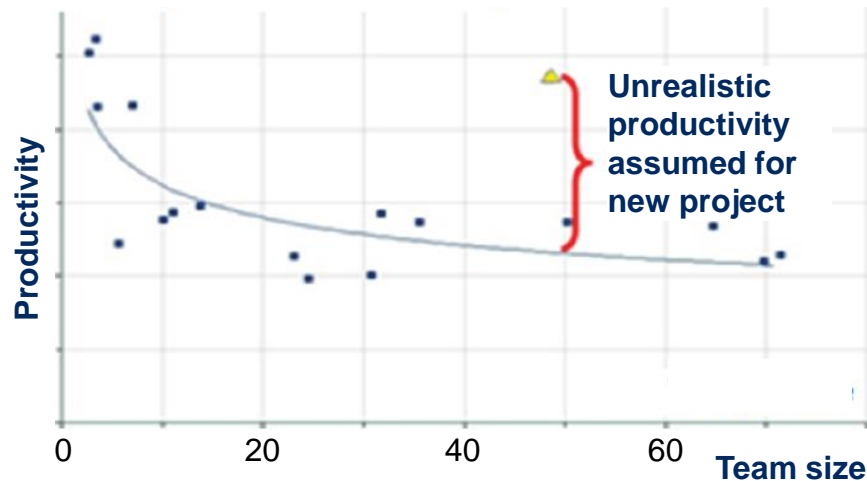
Project planning – predictive analytics is used to optimize schedule and staffing at the project and portfolio levels

"What-if" scenarios to determine tradeoffs and optimize the plan
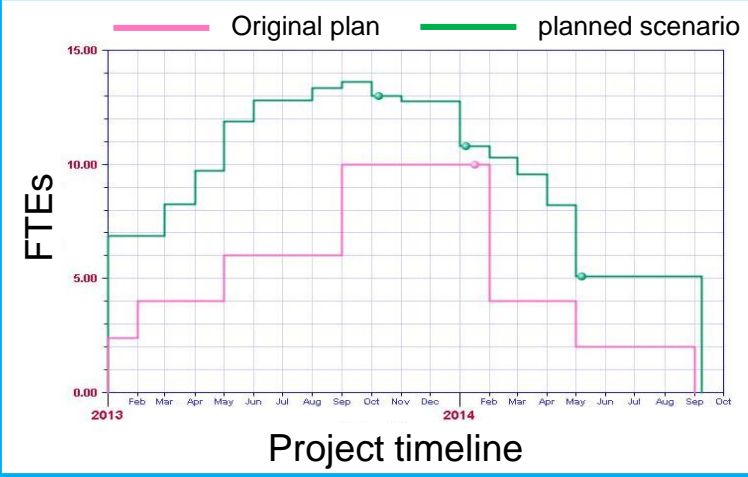
- Planned staffing plan is plotted against the predicted resource requirements to identify gaps

- "What-if" scenarios can be run to better understand tradeoffs between specifications, resources, budget and timeline, and to determine the optimal plan for the project

Analytics on required staffing and available resources across multiple projects
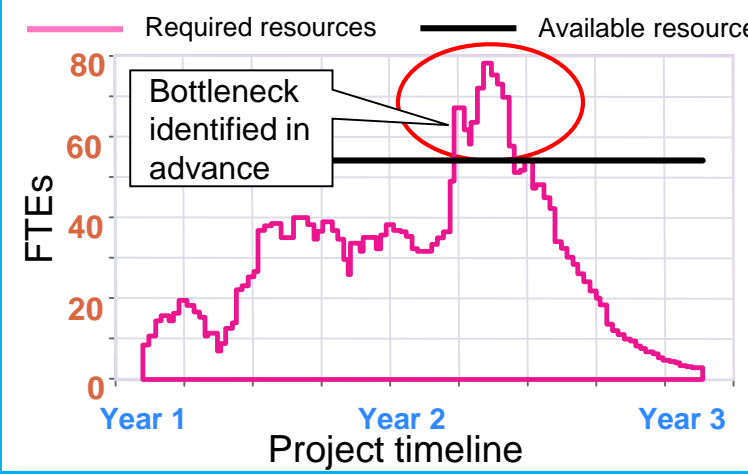
- Estimated staffing requirements by role and project phase across multiple projects is compared to available resources

- Resource gaps and bottlenecks are identified early on with plenty of time to adjust staffing levels, modify scope or reprioritize projects

Contents

Numetrics analytics enables step-function improvement in R&D productivity and time-to-market performance

Before analytics   After analytics

R&D capacity[1]

100%   120-140%

20-40%

Reduction in schedule slip[2] (TTM)

100%

60-90%

10-40%

How can I **get more out of my R&D spend** as complexity increases?
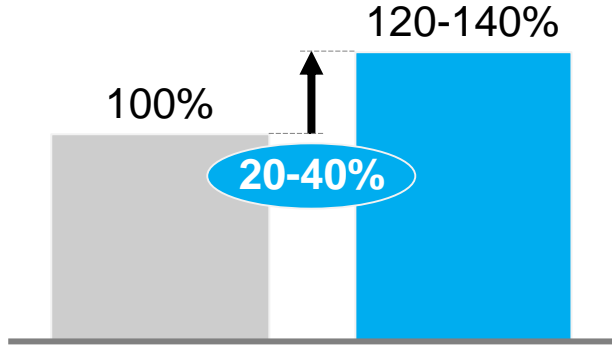
How can I **improve time to market and increase visibility across** the product road map?
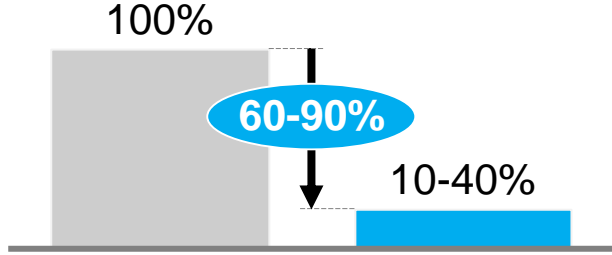
1 R&D Capacity is measured as "complexity units per person-week"
2 Schedule Slip is the amount of schedule overrun, expressed as a % of the original schedule.
   (e.g. if a 100-week project slips 12 weeks, then schedule slip = 12%)

There are several ways to engage Numetrics

	Scope	Engagement model
Analytics focused diagnostic	▪ 4-6 week (depending on data availability), **Numetrics led diagnostic** ▪ **Standalone analytic assessment** of 5-7 completed projects ▪ Provides a productivity baseline , industry benchmarks and analytic root cause analysis	▪ Numetrics team handles data entry, validation, analyses, and reports ▪ Client collects required project data under Numetrics' guidance and support
Deep R&D diagnostic	▪ 8-10 weeks **deep diagnostic, combining analytic and qualitative analyses** ▪ Includes **analytics focused diagnostic, complemented by qualitative tools** such as surveys, project deconstruction, process mapping, interviews and workshops to provide a complete view of productivity and performance drivers ▪ May include **planning of a new project** to determine required resources and schedule risk	▪ Numetrics team handles data entries, validation, analyses, tailored benchmarking and reports ▪ Client collects required project data with Numetrics' guidance
Subscription	▪ Embed Numetrics planning tool in the standard PD process to **continuously track performance** ▪ Use predictive analytics to **increase TTM transparency and optimize resource allocation** ▪ Includes initial benchmark and baseline creation and access to the planning tool	▪ Client trained to input project data and run reports directly using the web interface ▪ Numetrics team runs the analyses and provides insights

Benchmarking and root cause analysis require project data and timelines of several completed projects

	1 Data collection	**2** Complexity and Performance calculation	**3** Benchmarking	**4** Root cause analysis and recommendations
Activities	▪ Identify projects and data providers (often a project/program leader who solicits input from internal project records, architects or developers) ▪ Training on the input requirements (2 hours Webex or on-site) ▪ Start-up workshop: on-site, individual or group (3-4 hours) ▪ Collect data, including: – Project milestones and staffing history – Features / use cases – Team description, tools and methodology, specification changes, and defects data	Numetrics calculates complexity and performance metrics, such as: ▪ Design complexity ▪ Total duration and phase durations ▪ Total effort and phase effort ▪ Schedule slip ▪ Development productivity ▪ Development throughput ▪ Cost per complexity unit and total cost ▪ Reuse and reuse leverage	▪ Numetrics identifies a peer group of projects, as similar as possible to client projects ▪ Client performance is compared to the peer group, differences are highlighted using a variety of analytic tools and techniques including: – XY scatter plots – Radar charts – Tabular data – Phase charts – Histograms	▪ Analytic tools search for root causes for areas of high and low performance (identify drivers of performance) ▪ Use best in class practices to determine recommended course of action ▪ Share results and discuss implications and opportunities for improvement

Initial effort from client is approx. 5-6 hours per project

Numetrics' predictive analytics can help optimize project planning and timely execution

Baseline performance	Input project data	Calculate complexity	Estimate project plan	Identify risks in current plan
Past performance across a range of projects is assessed to build a performance baseline for the organization	New project characteristics (e.g., # features, re-use, platform) and constraints (e.g. resources) are captured	Numetrics' complexity engine, calibrated by a set of industry wide projects, estimates the complexity of the project[1]	Prediction engine estimates resource and schedule plan based on past performance, project data and complexity	Identify resource and schedule risks based on a comparison of predicted plan and project expectations or existing plan

Schedule & Resource Estimation

Schedule Risk Analysis

1 Measured in Complexity Units - A metric reflecting the amount of effort the average development team will spend on the project

Who to contact to get started?

Prasad Kunal

Director, Client Development

prasad_kunal@mckinsey.com

Mike Fogerty

Head of Client Development

Mike_fogerty@mckinsey.com

Ori Ben-Moshe

General Manager

ori_ben-moshe@mckinsey.com

Aaron Aboagye

Principal

aaron_aboagye@mckinsey.com

Contents

The Numetrics database holds 200+ automotive projects

Project database statistics

- 200+ (out of 1,600+ total) automotive SW projects from 20+ companies

- Applications include:
  - Infotainment systems
  - Mission critical systems such as:
    - Body & convenience systems
    - Powertrain
    - Chassis & safety

- Spans drivers, operating system, middleware and UI/application layers

- Team Sizes from 2 to >70 full-time equivalents

- Recent data (<3-4 years)

Sample descriptions of automotive projects

- "Two corner rear leveling of automotive suspension system"

- "Firmware which aligns steered axles after these are steered for maneuvering"

- "PID control of 4-wheel independent steering"

- "I/O drivers for a powertrain control module"

- "Car Navigation system software with dynamic Traffic-based Route Guidance and Visualization on a graphical map"

- "GPS-based timing and data acquisition"

- "Car radio tuner both frequency and digital tuning capable."

- "Battery monitoring and management software for hybrid vehicle power systems"

- "Automotive steering angle sensor and LIN switch gateway"

- "Powertrain Engine Controller software platform"

- "Emergency call and pay-as-you-drive capabilities."

Numetrics is a well-established company with a field proven sets of solutions

1998
Launch of semiconductor benchmarking solution

2001
Launch of semiconductor predictive planning solutions

2004
First embedded SW complexity model

2006
Launch of embedded SW predictive planning solution

2010
First 1,000 SW projects released in industry database

2013
Numetrics acquired by McKinsey

- **Extensive database of ~2000 IC and ~1600 SW projects**
- **Field proven complexity estimation and predictive analytics algorithms**
- **Wide industry coverage including automotive, aerospace & defense, high tech, financial services, medical, etc.**