IBM Watson IoT

Beyond RE: Solving the really hard problems in engineering and development

Daniel Moul
Senior Offering Manager
IBM Watson IoT Engineering

REConf 2019

IBM

Session description

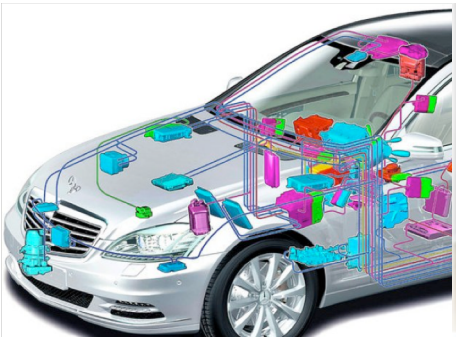
Requirements are only meaningful if they are defined and used within a development process – ideally one that guides teams to maximum impact and maximum efficiency. This statement has many implications for your development process and the many tools your developers and engineers use as they bring your development process to life. A foundational enabler is the ability to create a shared development context for your teams and evolving it in a controlled way: across multiple tools from multiple vendors, maintaining dependencies among the data in these tools, and with effective change management. This session will briefly survey ways this is typically addressed today with the various compromises that are inherent in these approaches, then explain the concepts and promise of “global configurations” in your engineering tool chain as enabled by OASIS OSLC Configuration Management and implemented by the IBM Continuous Engineering solution for software and systems engineering.

Smart and connected products challenge existing engineering processes

- More features, mostly in software
- Higher quality / recall avoidance
- Multi-tier value chains
- Growing regulatory demands
- Time-to-market pressure

Complexity is rising!

	Lines of code
Mars Curiosity rover:	**0.5m**
F-35 fighter jet:	**25m**
Premium car:	**100m**
Fully autonomous car:	**800m**




The challenge

Design and develop a complex product
from very many ...

- Software and hardware components
- Engineering specialties, teams, suppliers and subcontractors spread around the planet
- Engineering tools & their data

Then create many product variants

Then evolve them in way that is

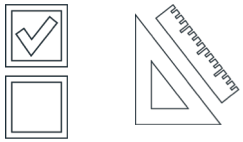
- Controlled
- Cost effective
- Time effective
- High quality

Version 10  + Version 3  + Version 9  + Version 8  + Version 6  + 30,000 other things = 1 engineering environment configuration

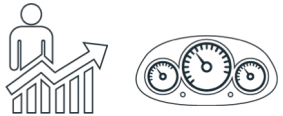
+ Traceable engineering tools

+ Open and effective integrations

- Systems engineering
- Requirements management
- Change management
- V&V
- Version management / configuration management
- Reuse / product variants
- Traceability
- Auditability
- Reporting
- Good process
- Good people



Correctness



Efficiency

$$\sum_{1}^{n} LocalOptimization$$

$$\sum_{1}^{n} LocalOptimization \neq GlobalOptimization$$

$$max\left(\sum_{1}^{n} LocalOptimization + \sum_{1}^{m} GlobalOptimization\right)$$

IBM Engineering Lifecycle Management: transforming smart products engineering

Early design verification

Verify at all stages of the product lifecycle with model based engineering and digital twins

Scaled agility

Effective agile engineering with digital governance, real-time feedback, team collaboration, and continuous delivery

Engineering insights with AI

Use AI and advanced analytics to improve quality and support engineering decision making

Digital continuity

Enable cross discipline digital threads to streamline impact of change analysis and standards compliance

Strategic reuse and product line engineering

Reuse engineering data in parallel development and product variants

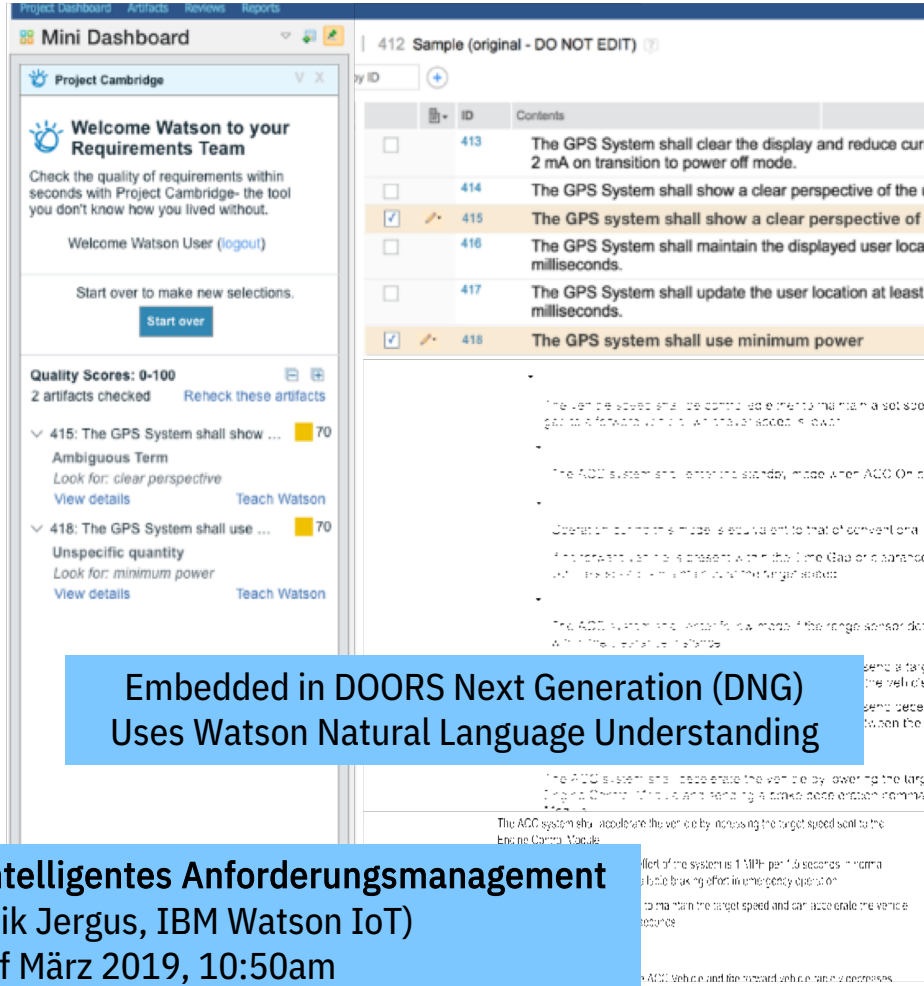
Correctness

Efficiency

IBM Requirements Quality Assistant

Reduce risk and ambiguity using Watson AI

- Scores requirements against criteria consistent with the **INCOSE Guidelines for Writing Good Requirements**
- Authors receive coaching from Watson to improve the quality of their requirements
- Pre-trained to detect 10 quality issues
- Add additional dimensions of quality, or customize to your industry or company



Embedded in DOORS Next Generation (DNG)
Uses Watson Natural Language Understanding

Requirements Quality Assistant - Intelligentes Anforderungsmanagement mit IBM Watson (Dominik Jergus, IBM Watson IoT)
Wednesday, 13th of März 2019, 10:50am

IBM Engineering Lifecycle Management: transforming smart products engineering

Early design verification
Verify at all stages of the product lifecycle with model based engineering and digital twins

Scaled agility
Effective agile engineering with digital governance, real-time feedback, team collaboration, and continuous delivery

Engineering insights with AI
Use AI and advanced analytics to improve quality and support engineering decision making

Digital continuity
Enable cross discipline digital threads to streamline impact of change analysis and standards compliance

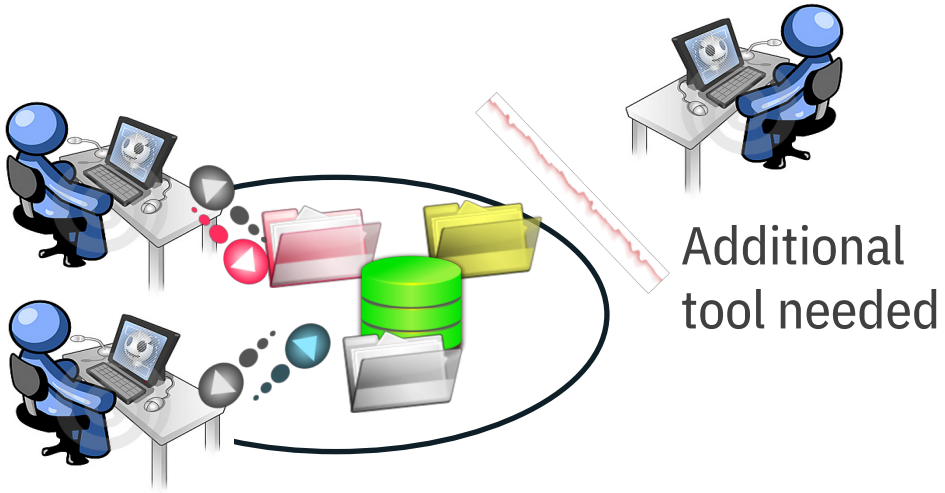
Strategic reuse and product line engineering
Reuse engineering data in parallel development and product variants

Correctness

Efficiency

Half-way solution (1) single-repository version/configuration mgmt

- Not practical: tools come from many vendors
- Can’t use best-of-breed tools across HW & SW configuration management
- Life happens: mergers, acquisitions, re-organizations



Half-way solution (2) file-based version/configuration mgmt

- Your files in SCM are not in sync with the artifacts in the tools
- Loss of artifact versioning, history, and audit trail
- Hard to create and maintain dependencies between resources
- Queries and reports on past baselines require reconstructing tool data
- Difficult to monitor or enforce adherence to policies

User-managed and file-based Software Configuration Mgmt

Files are lowest common denominator

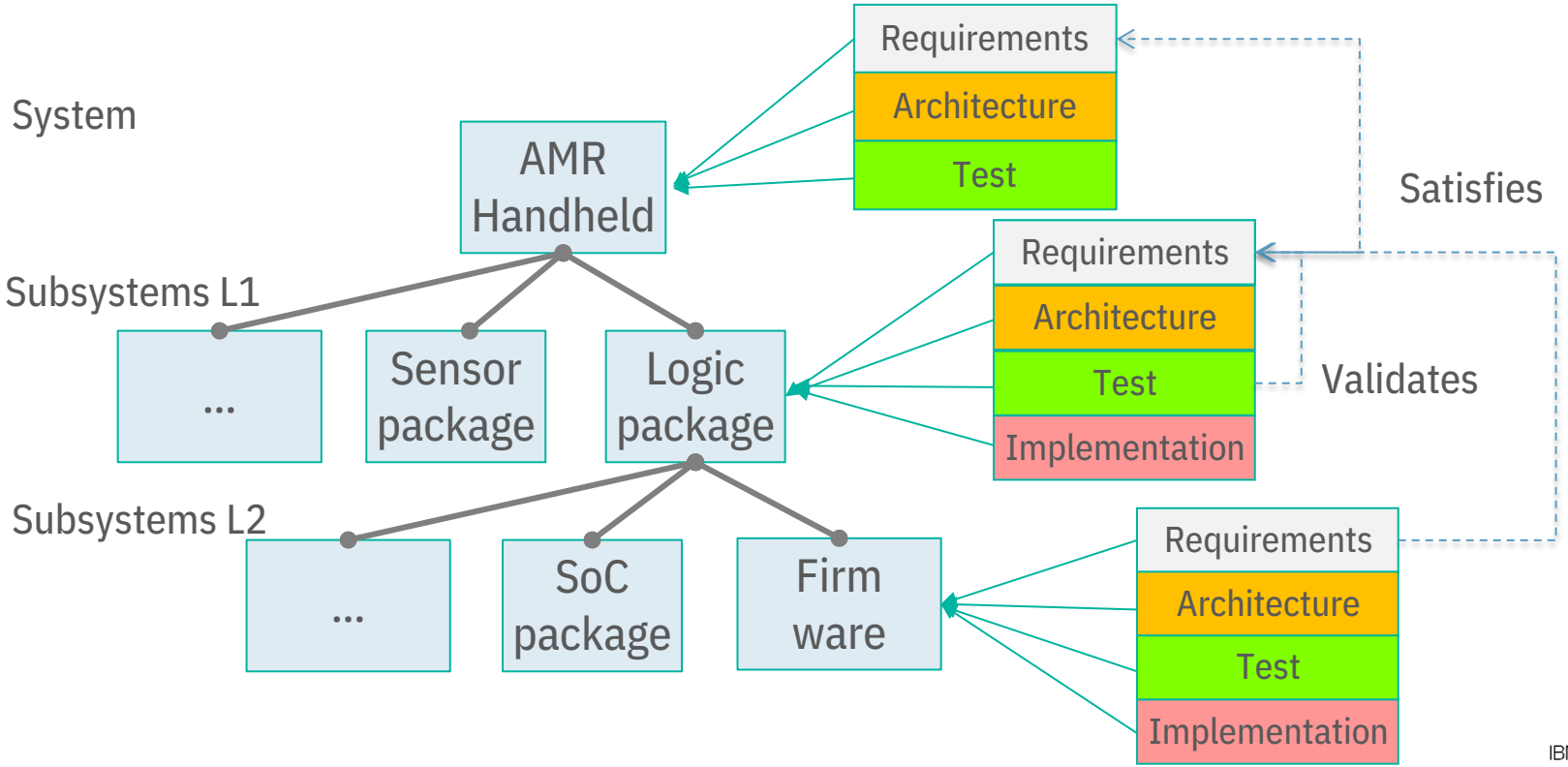
Creating shared development context: assumptions and conclusions

Open world assumptions	⇒	OASIS OSLC
Non-homogenous tools, data, teams, processes	⇒	W3C linked data, federated data stores
Shared configuration context	⇒	Configuration Mgmt specification defines “global configuration” dev streams and baselines
Products are systems of systems	⇒	Global config hierarchies
Automation improves correctness and efficiency	⇒	Resolve link and resource versions based on GC

A configuration includes…

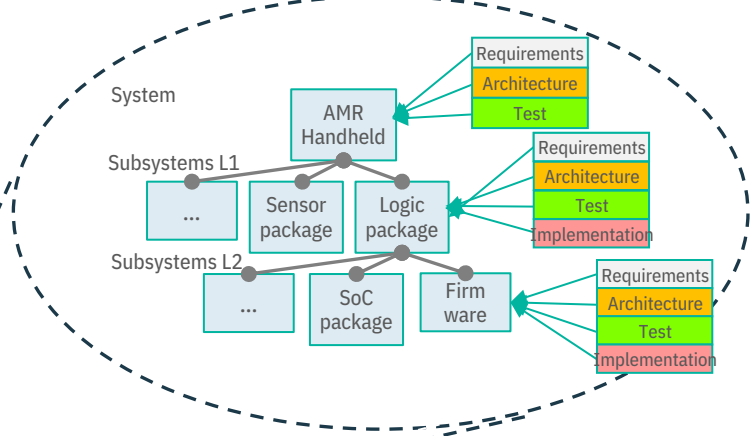
Versions of the artifacts	→	Requirements, designs, documents, test plans, test cases, calibrations, source files
Build environment	→	Type systems, database schemas Tools, scripts, compilers, library and operating system version and patch information User environment (options, settings, ini files, config files, etc.)
Revision history, including change comments	→	Who changed what, when, and why
Links between artifacts	→	Links need to be versioned just as other properties of artifacts And then navigated in the context of the relevant configurations (including baselines)

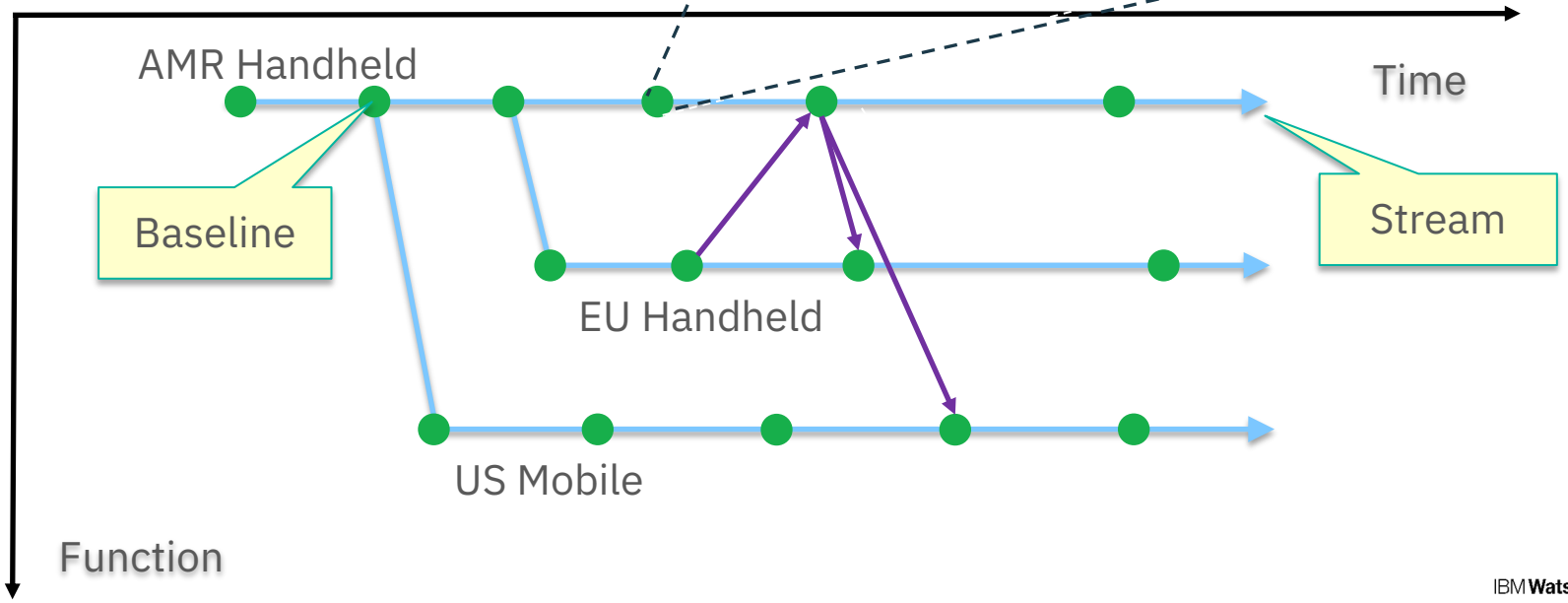
Complex products are a hierarchy of streams and baselines



Complex products in a product line

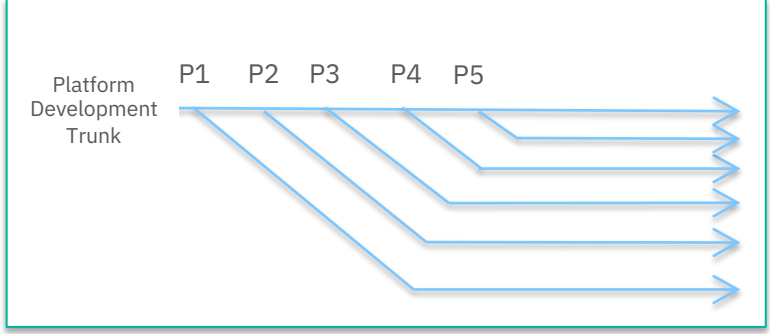
Evolves over time …
with variants to meet different needs

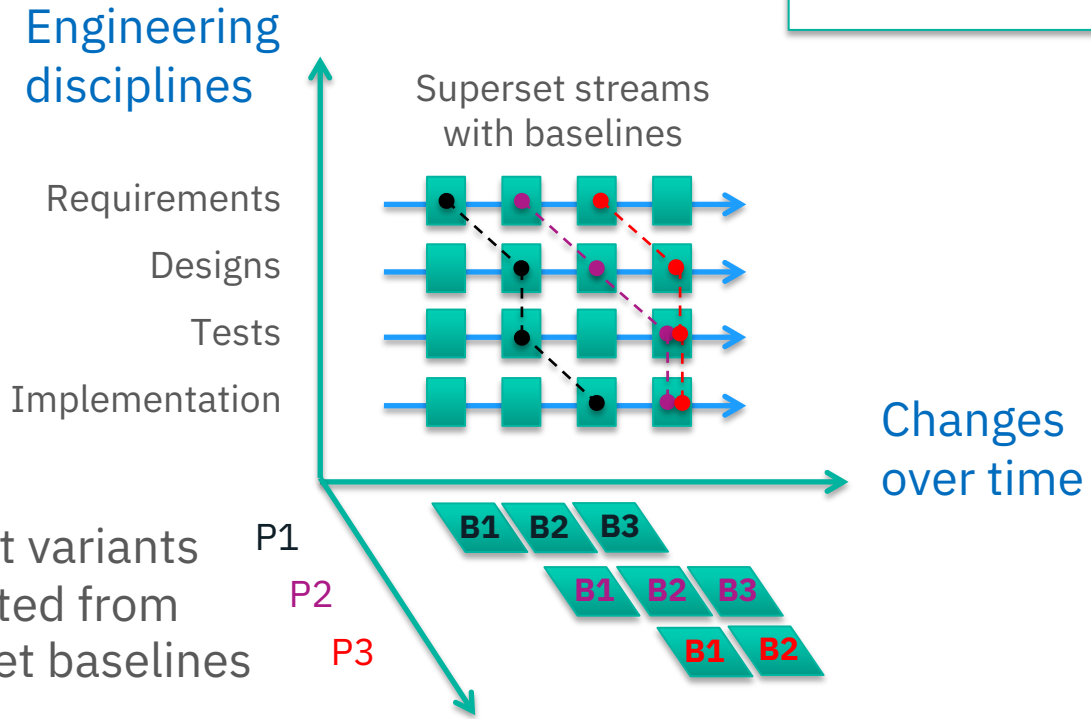


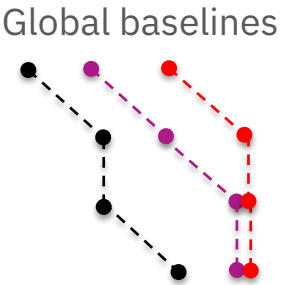


Generating product variants

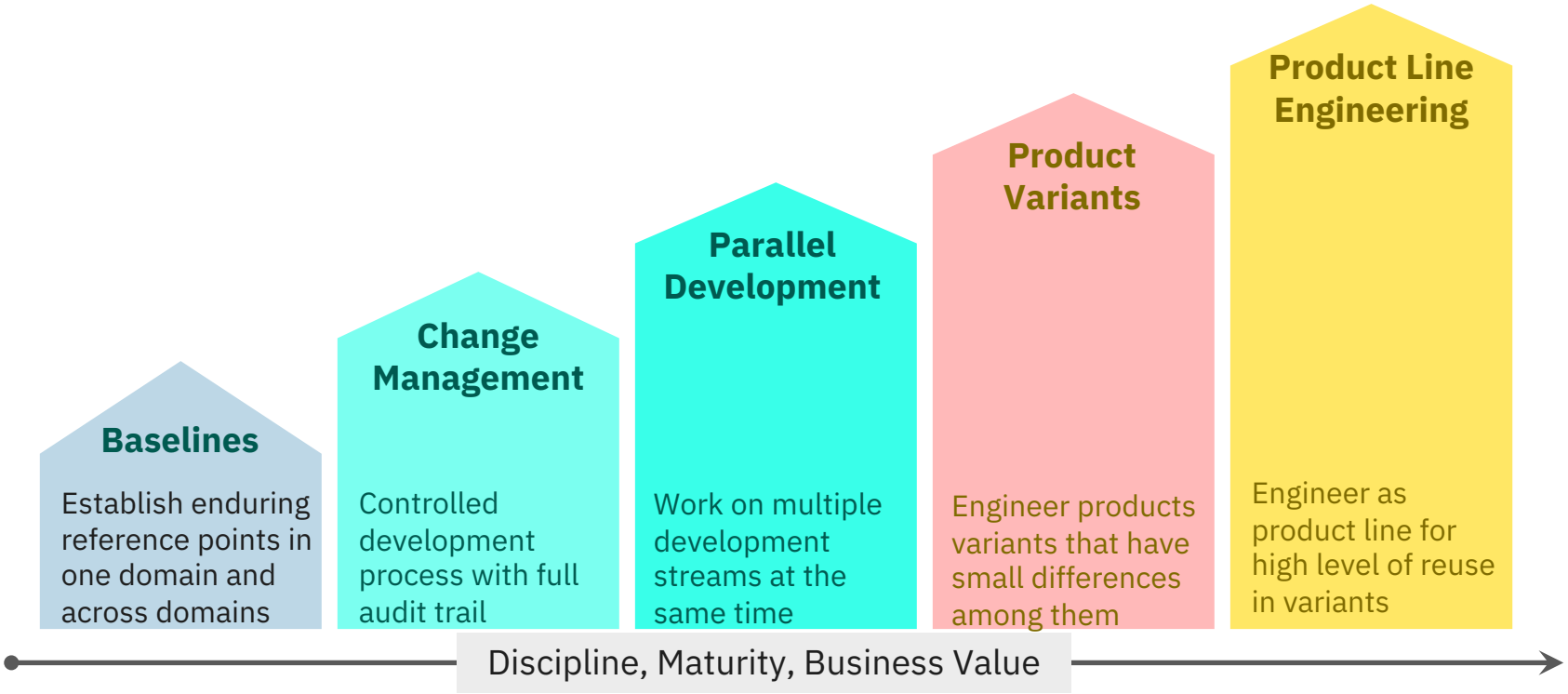
Where our feature-modeling partners fit pure-systems and BigLever





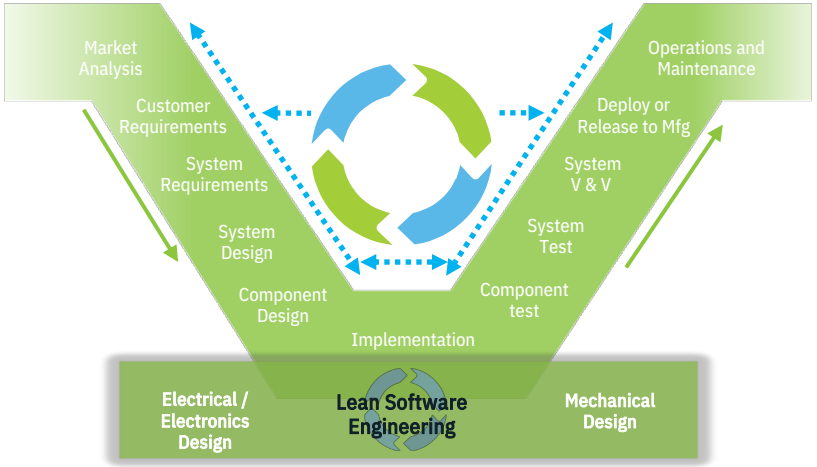


Which is your target?



We need **systems thinking** to design complex products … and to design modern engineering processes and tools

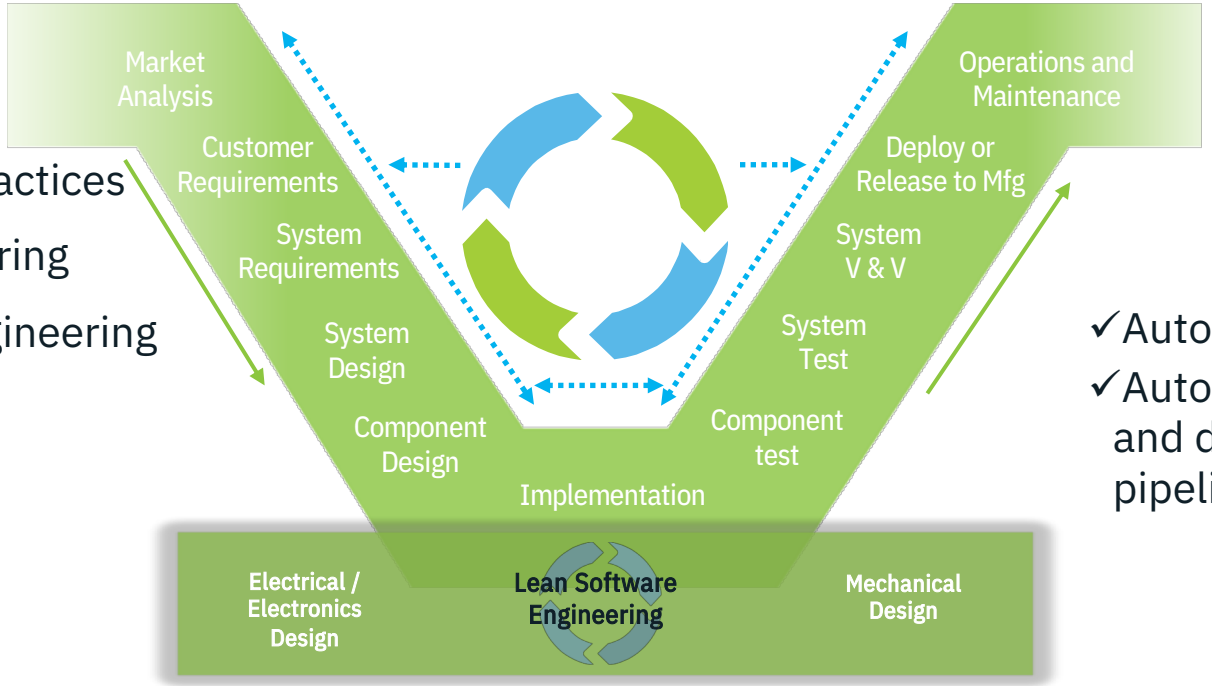
1. A system is not the sum of its parts, it is the product of the interactions of its parts
2. Performance of a system is dependent on how the parts fit and work together
3. Performance improvement programs can fail because they optimize individual parts at the expense of the whole
4. Finding and removing deficiencies is not the best way to improve the system
5. Discontinuous improvement (creativity, breaking away from the past) can be more impactful than incremental improvements



Global configurations are a way of realizing systems thinking

Engineering Lifecycle Management with IBM

- ✓Requirements practices
- ✓Systems engineering
- ✓Model-based engineering
- ✓Simulation



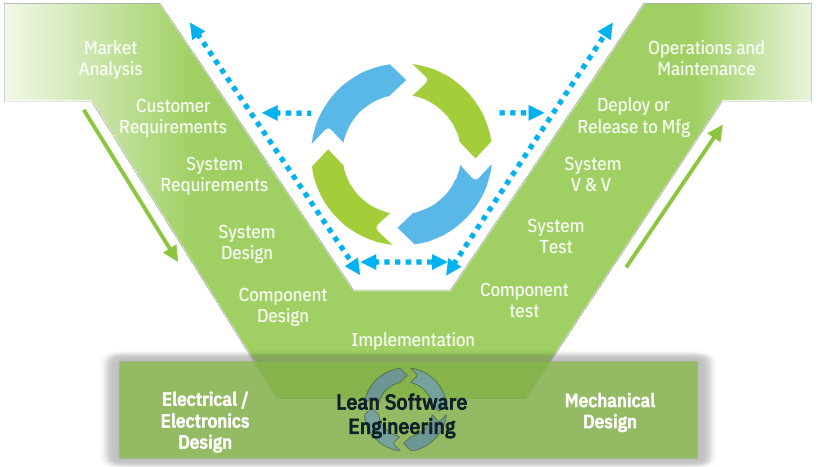
- ✓Automated testing
- ✓Automated build and deployment pipeline (DevOps)

- ✓Change management
- ✓Version and configuration management

- ✓Iterative / agile / lean practices
- ✓Automated reporting and doc generation
- ✓Regulatory compliance

We help teams to design and develop complex software-intensive products and systems

1. With change management and configuration management
2. With traceability
3. At scale
4. With high levels of reuse
5. Addressing standards and compliance
6. Using open standards and integrations to bring together multiple teams using tools from multiple vendors
7. Across the whole Systems V



We help teams to design and develop complex software-intensive products and systems


IBM

Learn more

1. Russell Akoff on Systems Thinking
2. Systems and software engineering: https://www.ibm.com/internet-of-things/solutions
3. Interactive whitepaper: https://www.ibm.com/internet-of-things/learn/continuous-engineering-IoT/
4. Strategic reuse and product line engineering
5. Engage with us at https://jazz.net where we use our tools to develop our tools

Summary

- Configuration management across the engineering lifecycle is an essential need in today’s complex product engineering to deal with the growing complexity
- Global configuration management enables agile engineering, industry compliance, cross program reuse, and product line engineering
- IBM offers unique engineering lifecycle management capabilities with open, federated configuration management
- Advanced PLE use cases with partner feature modeling tools make use global configurations to automate the creation of product variants