

IBM Analytics

Engineering Change and Configuration Management

Gavin Arthurs P.E
Systems Solution Architect
IBM Federal
CSEP, OCSMP



Version and configuration management of engineering artifacts

The Vision

- Enable engineering / product development teams to
 - Achieve high levels of reuse and parallel development
 - Define and work in context of the products and components they are developing
 - Engineer product lines
- Across engineering domains
- Employing version and configuration management across these domains
- Use federated, linked lifecycle data that is created and managed by multiple tools from multiple vendors
- Using open standards and specifications

“Product lines are a development paradigm allowing companies to realize order-of-magnitude improvements in time to market, cost, productivity, quality, and other business drivers. Software product line engineering can also enable rapid market entry and flexible response, and provide a capability for mass customization”

- SPL handbook by SEI

What do we mean by Product Line Engineering?

Maximizing reuse of engineering assets across product variants

- Reuse architecture based on configurable components
- Avoid duplication -> maintenance nightmares across hundreds of product variants

Specifying a product definition

- The product consists of which components [sets of versioned engineering artifacts]?

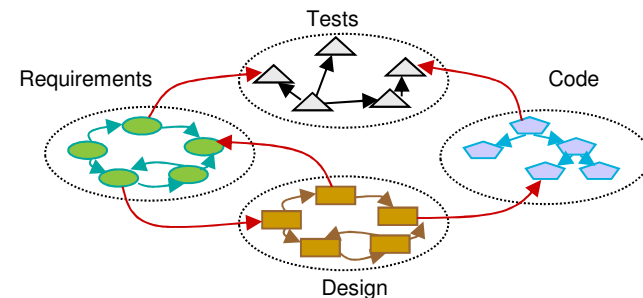
Consistently manage artifact versions and product variants across all lifecycle disciplines

- Create cross-component, cross-tool baselines
- Query and do analysis in multi-version/multi-variant environments

Effectively handling change propagation to a multitude of variants

- “Where does this change need to go?”

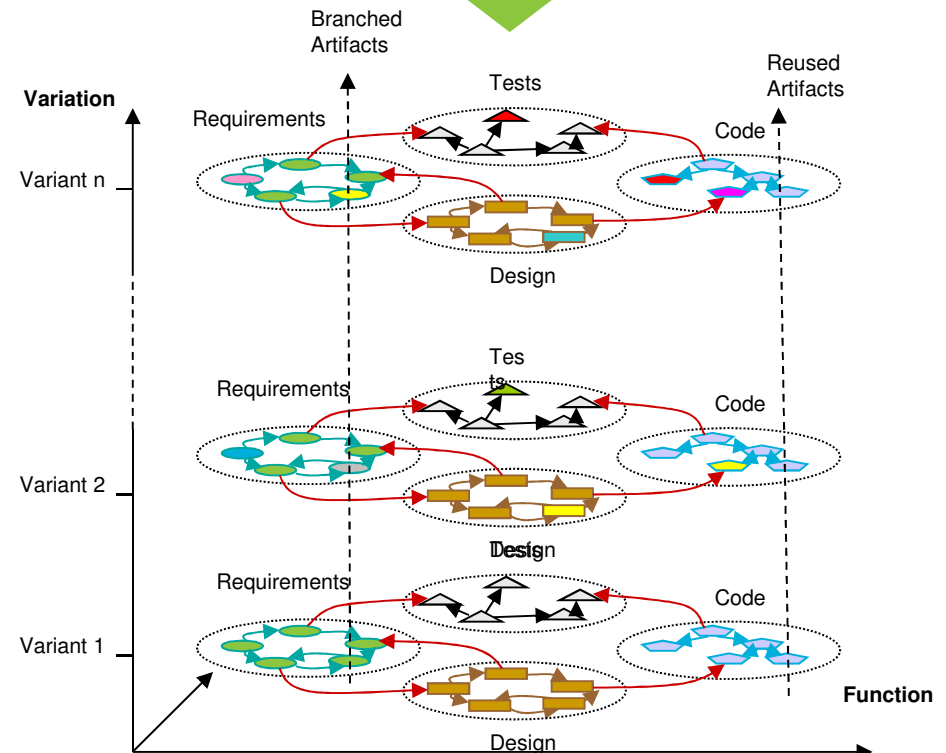
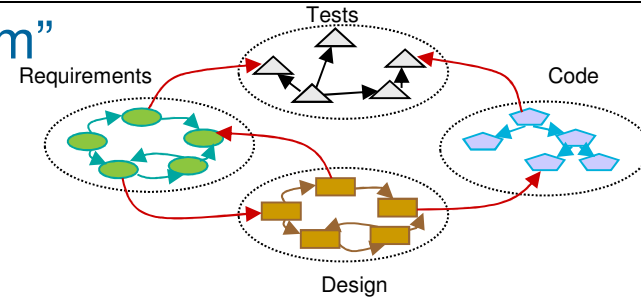
Effectively creating new product variant based on functional parameters (“features”)



More automation – less manual bookkeeping

- Tools manage the product definition – not spreadsheets that you manually maintain
- Tools present to the engineers the right artifacts at the right versions – and the right links between artifacts

From “single system” to “multi-variant system”



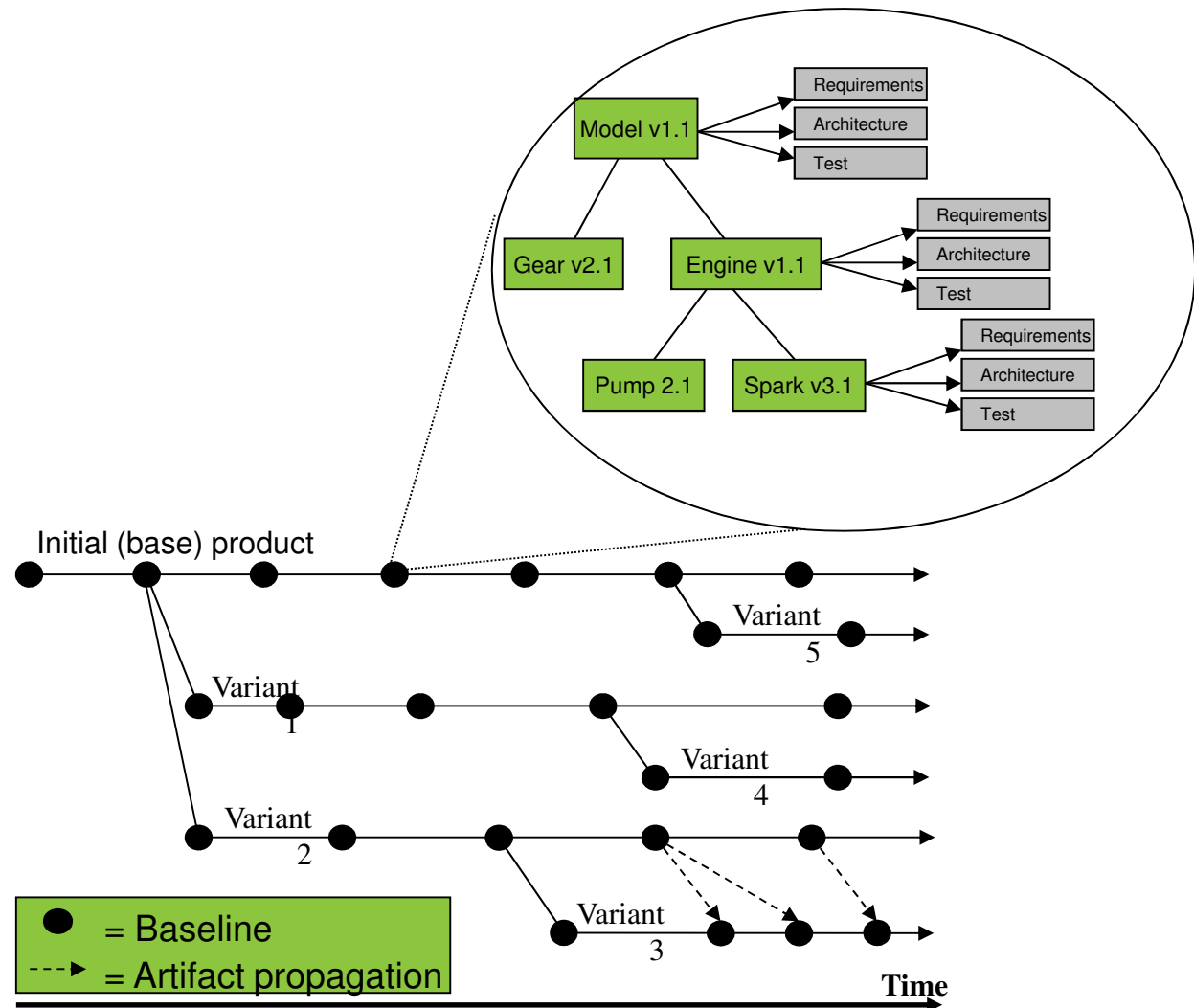
For SSE / CLM it is transforming from a “single system” solution to “multi-variant”, adding a variant dimension to the entire lifecycle

Some key capabilities

1. **Configuration management of requirements and across engineering domains**
2. **Parallel development** of more than code
3. **Hierarchical, global baselines**
4. **Definition of products** as sets of reusable “components” containing the respective engineering artifacts
5. **Automation** to select product variant’s components **via parameters** or feature model
6. Automation to enable **parameters** defined in product definition to be **used in component’s engineering artifacts** (requirements, designs, tests, software, ...)

A conceptual reuse model for multi-variant development

- Each product variant is essentially a **stream** of evolving engineering **artifacts**
- Artifacts have **versions** and logically grouped in **components**
- A stream evolution is a sequence of global **baselines** which record a particular state of the entire artifact set
- New variants can be **branched** from existing variants at certain baselines and evolve in parallel streams
- Changes may propagate across streams both upstream and downstream



Key idea: a simple equation – multi-stream

Configuration management of engineering artifacts in domain tools – domain components

PLUS

Products defined as Hierarchical sets of components

PLUS

Product variants defined by as product definition branches and the domain component configurations

EQUALS

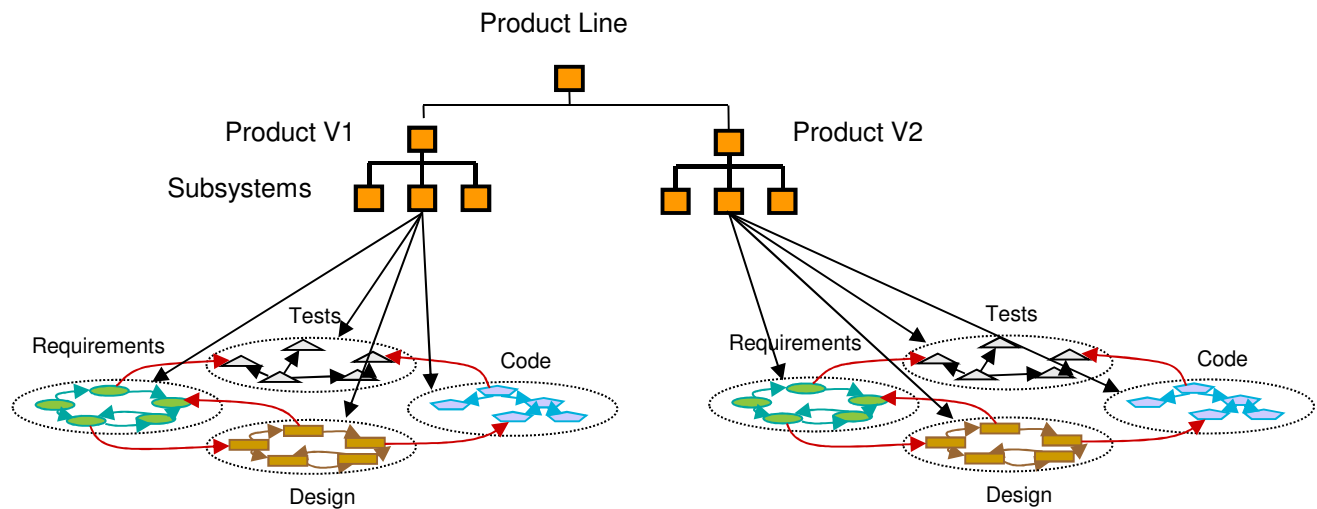
“Multi-stream” product line engineering

Versioned artifacts, baselines, configurations

Product definition tree

Product definition tree + global configurations + domain configurations

Each tool implements global configuration consumer spec. How each tool does version management “under the covers” doesn’t matter.

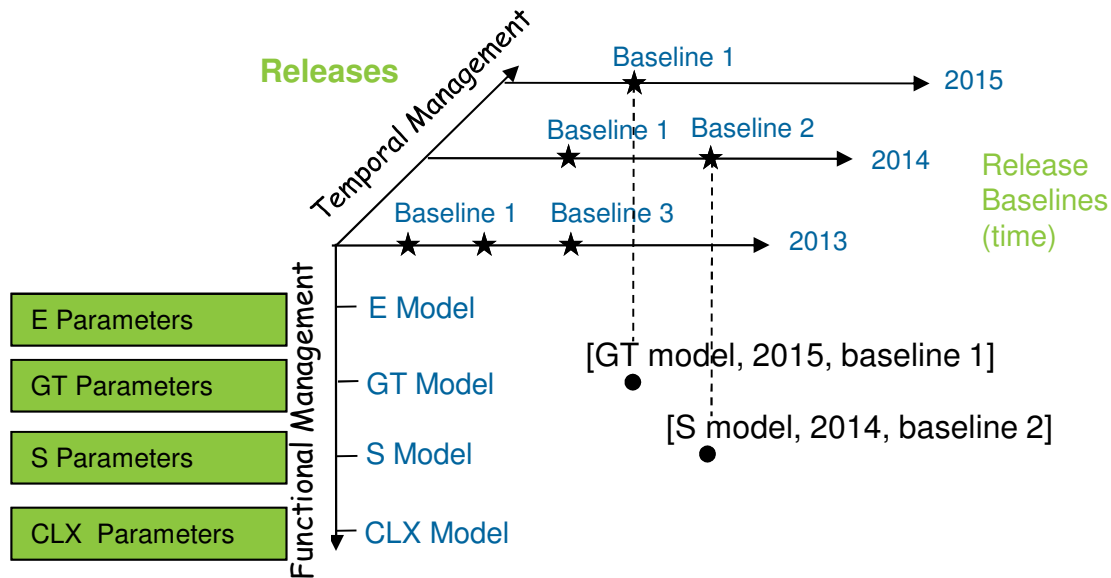
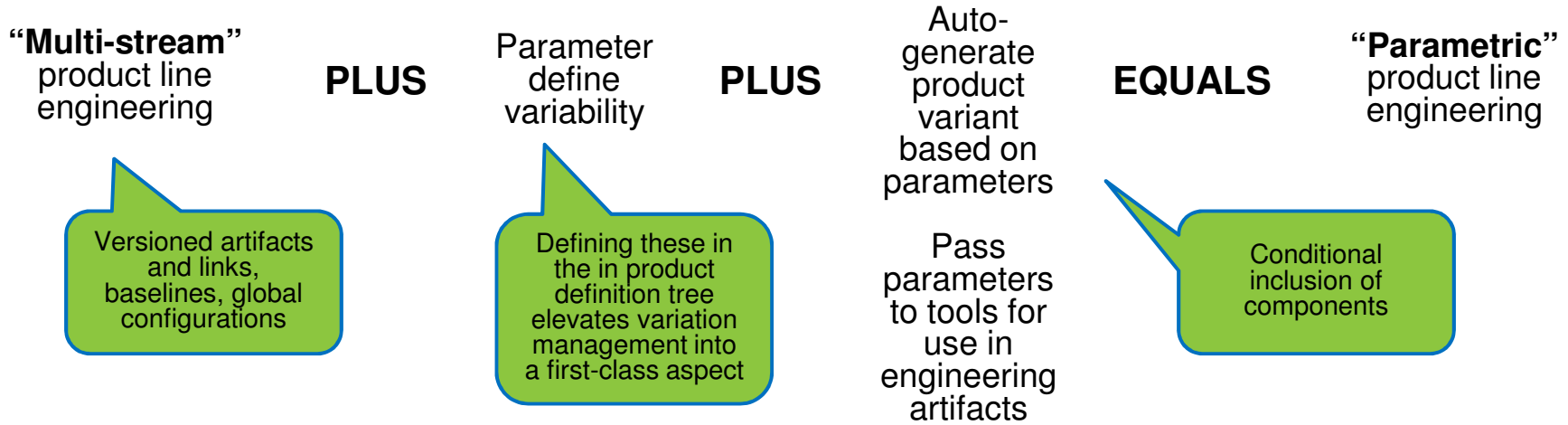


Parametric PLE

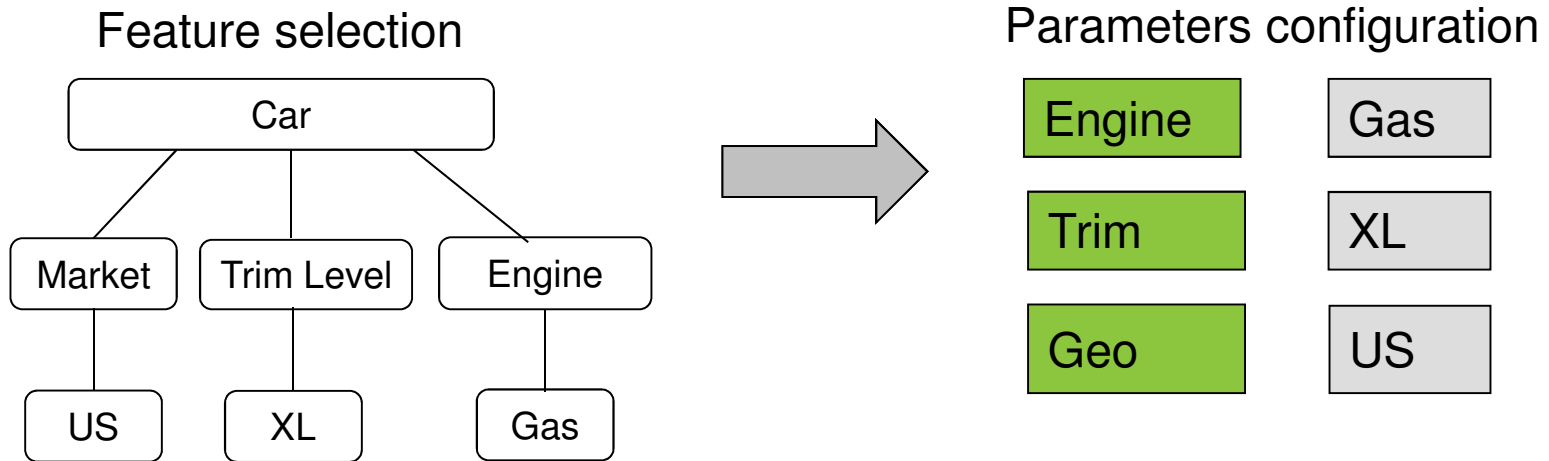
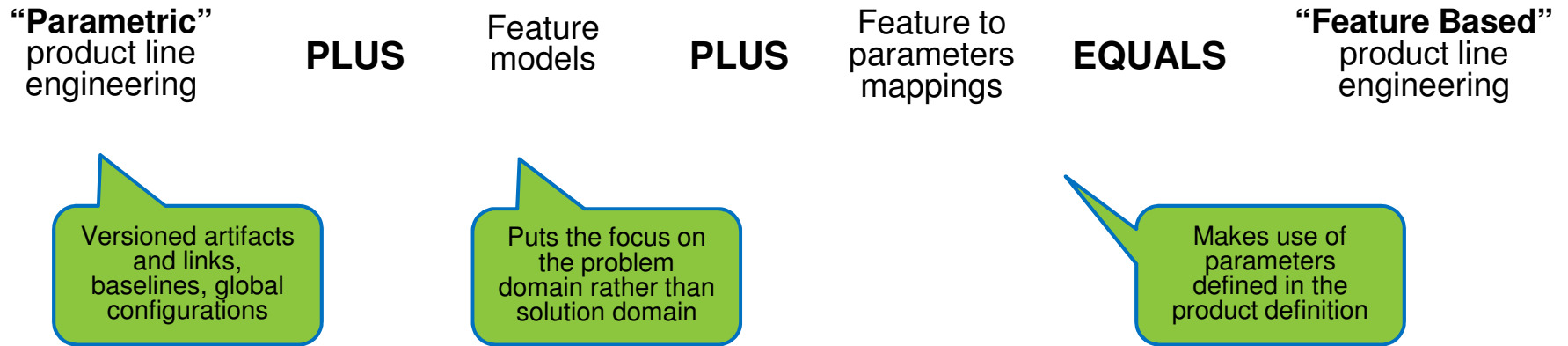
- Facilitating higher level of component and artifact reuse using parameters
- Defining a “platform” with a set of reusable parametric artifacts
 - A parametric artifact includes elements that depend on certain parameters
 - Increases the reuse of individual resources for multiple products
- A derived artifact is created from a parametric artifact by applying concrete values to parameters

	Assets
Platform	Shared parametric platform Assets
Product	Configured PLE assets for a Product instance using parameters

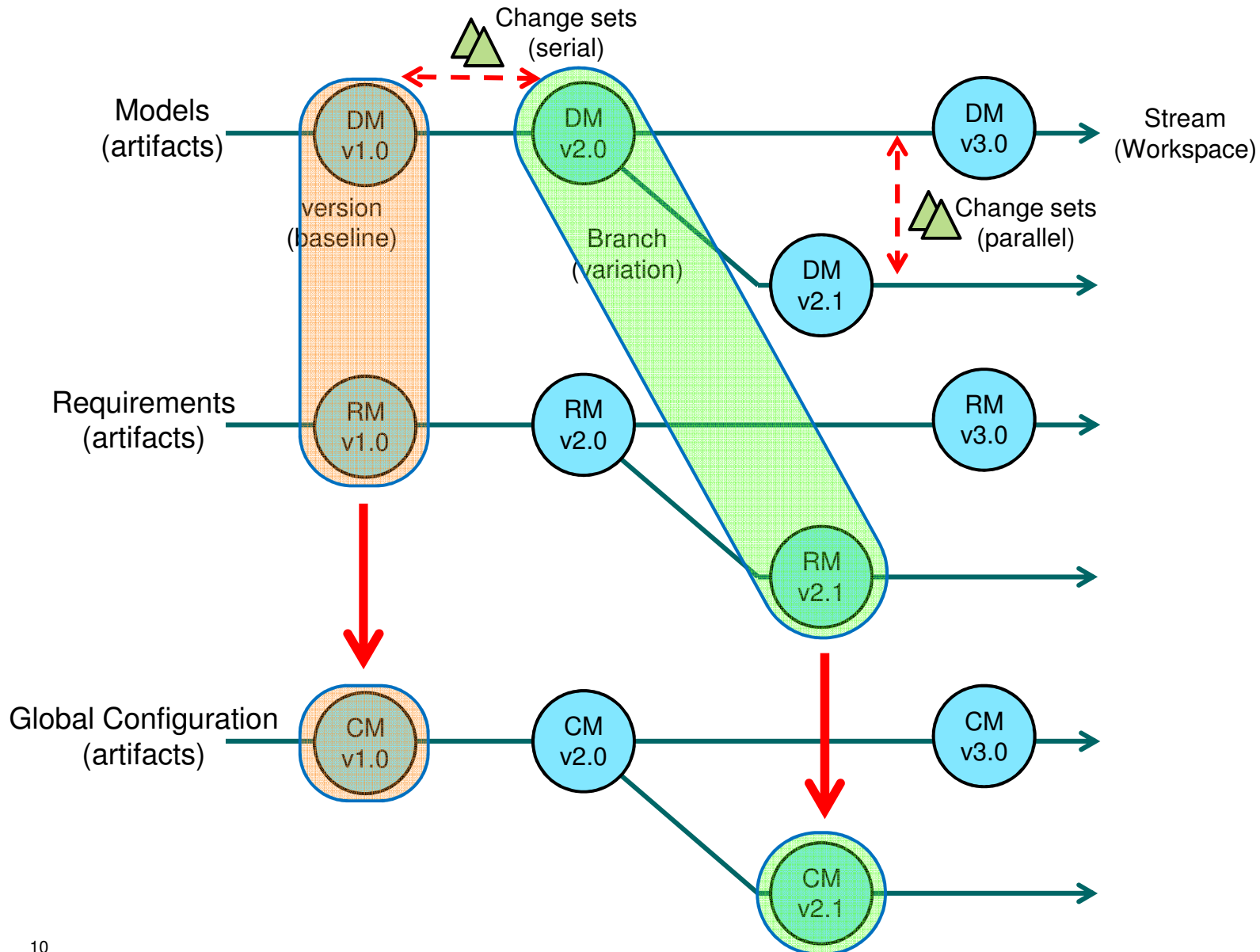
Key idea: a simple equation – parametric



Key idea: a simple equation – feature-driven



Basic workflow and concepts



Demo – Rhapsody and Design Manager

