Consolidate.

Simplify.

Leverage.

The Best of Both Worlds: Agile Development Meets Product Line Engineering at Lockheed Martin

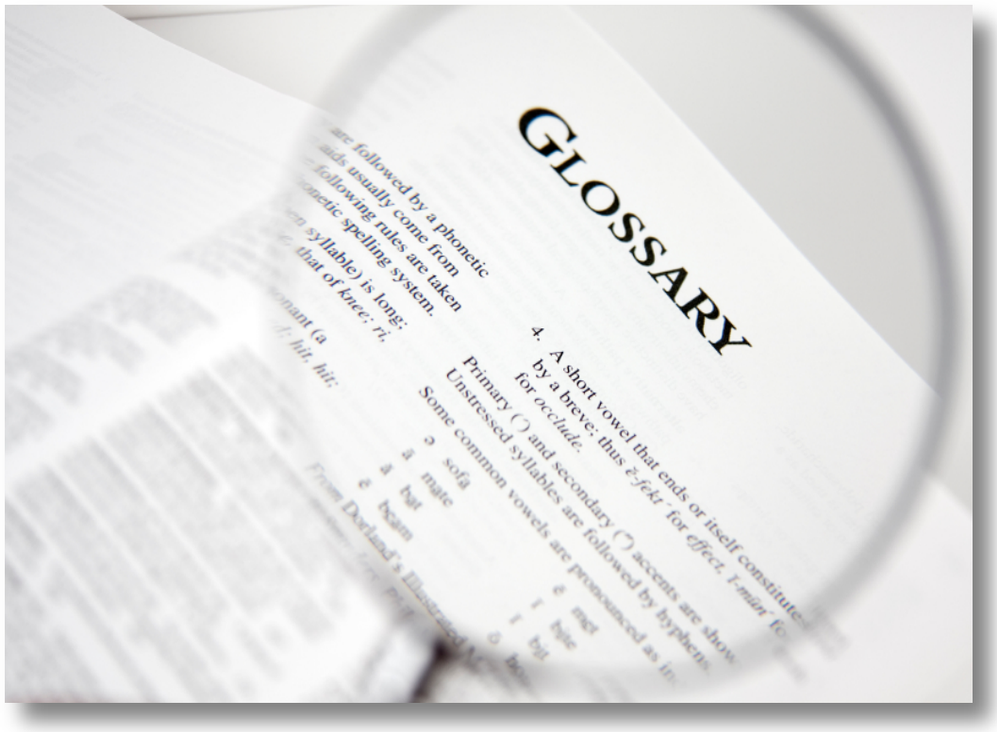
Susan P. Gregg, Rick Scharadin
Lockheed Martin Missions Systems and Training

Dr. Paul Clements
BigLever

Product Line Engineering (PLE) Defined

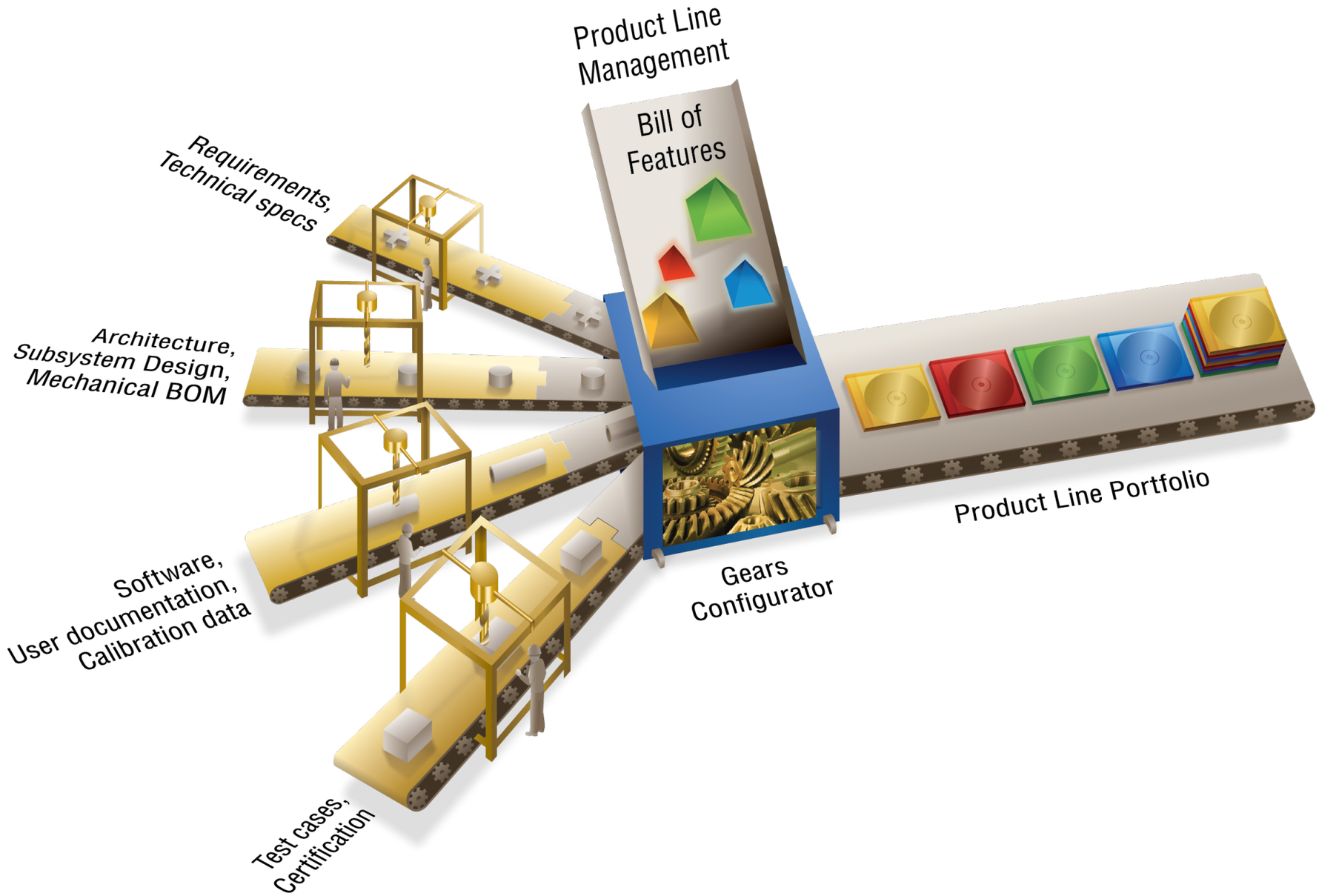
- **Product Line**: a family of similar products with variations in features and functions

- **Product Line Engineering**: the engineering of a product line using a *shared set of engineering assets,* a *managed set of features,* and an *efficient means of production…*

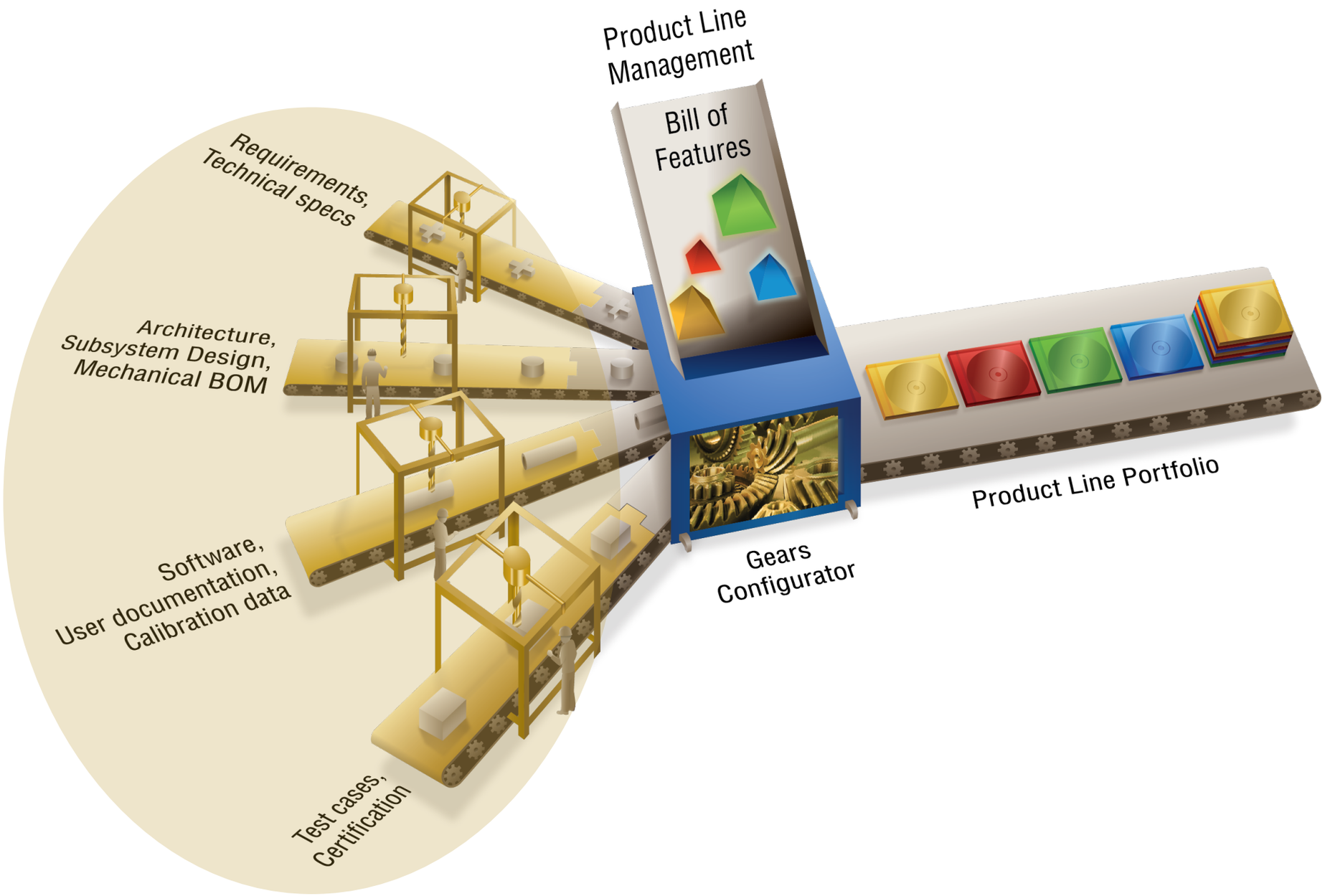


- taking advantage of the **commonality** shared across the family
- efficiently and systematically managing the **variation** among the products

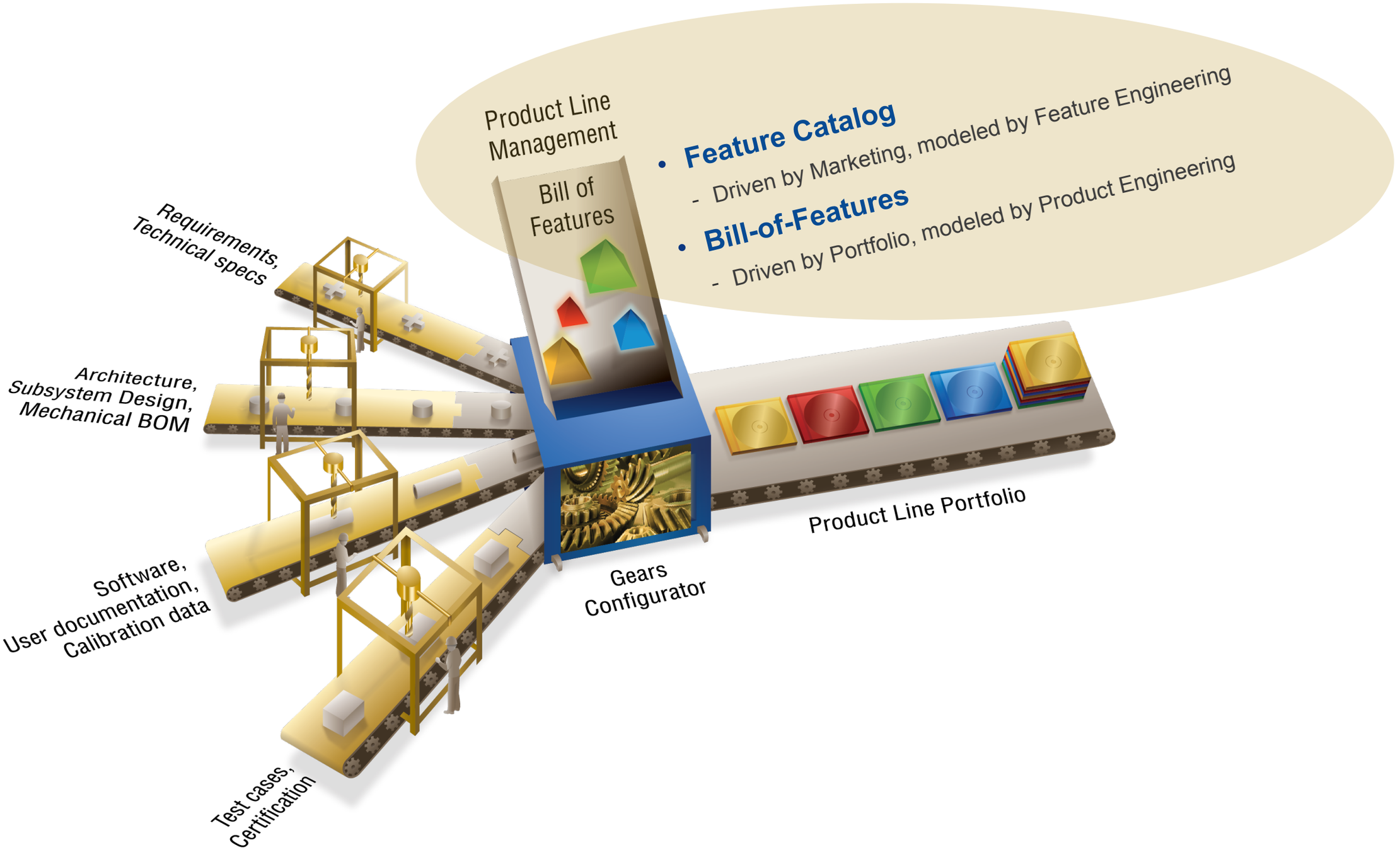
An Efficient Means of Production for Product Lines



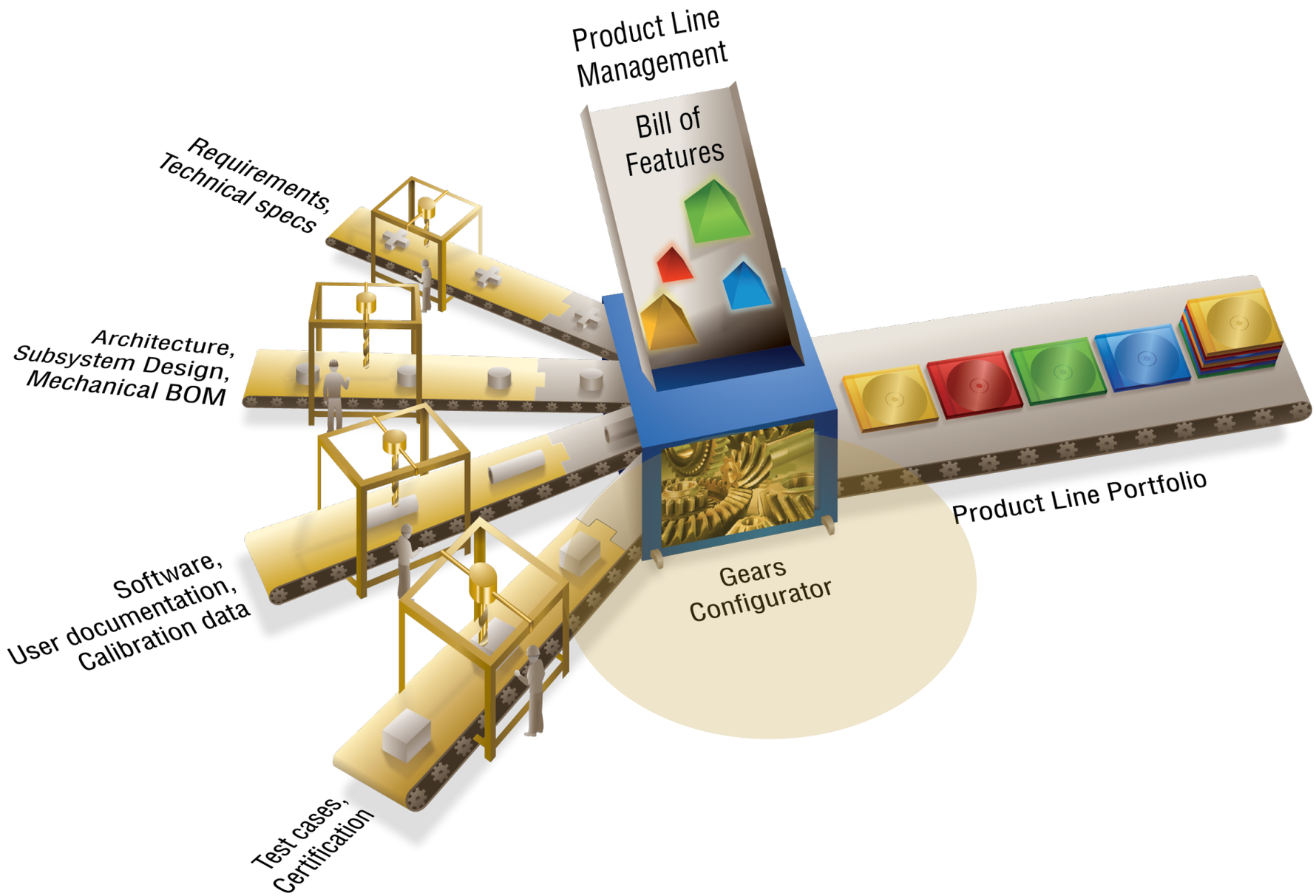
Shared assets are like the factory's supply chain.



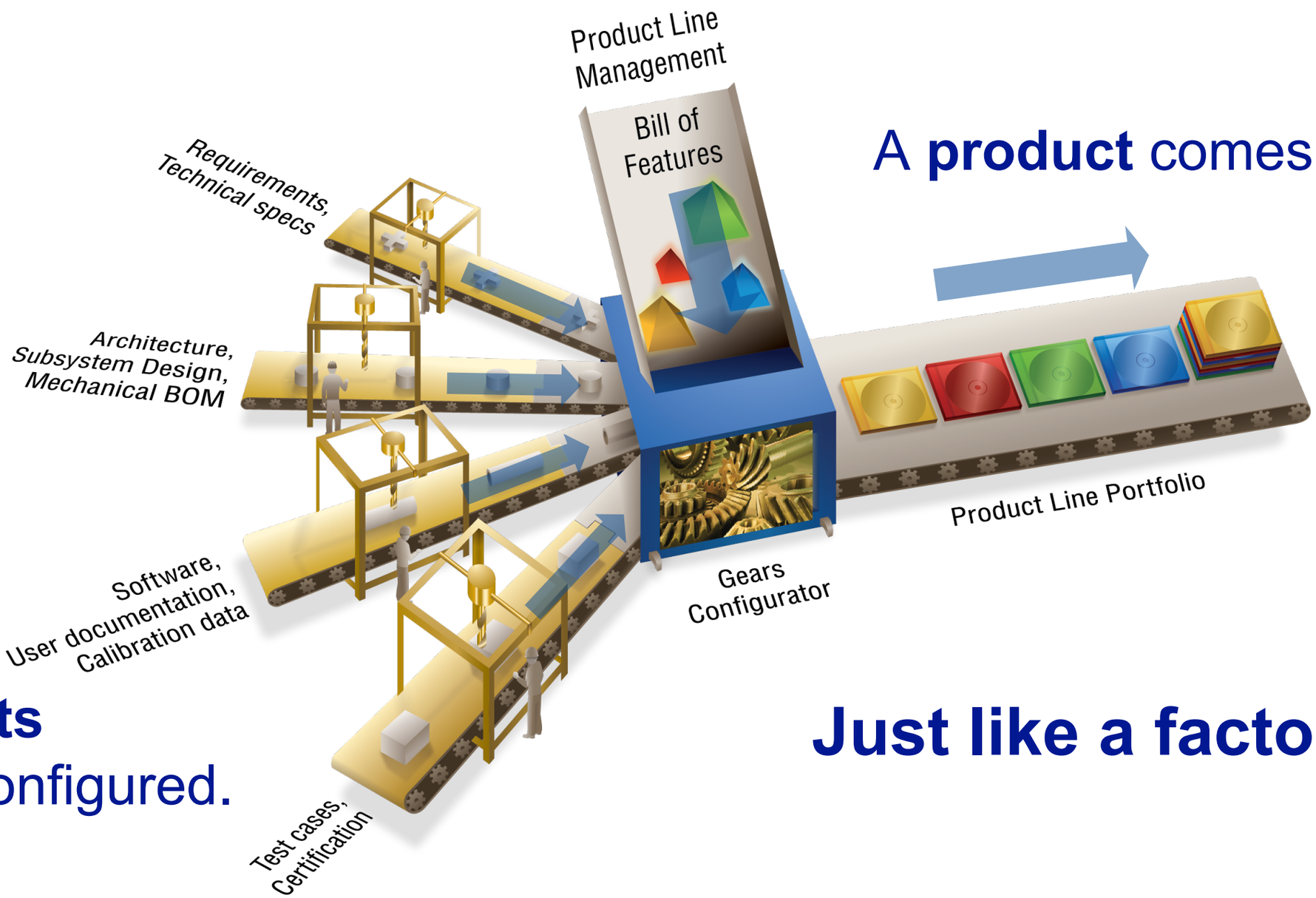
Features describe capabilities that vary among products.



Assets are configured according to the *feature profiles* of the products you want build.



Features come in.

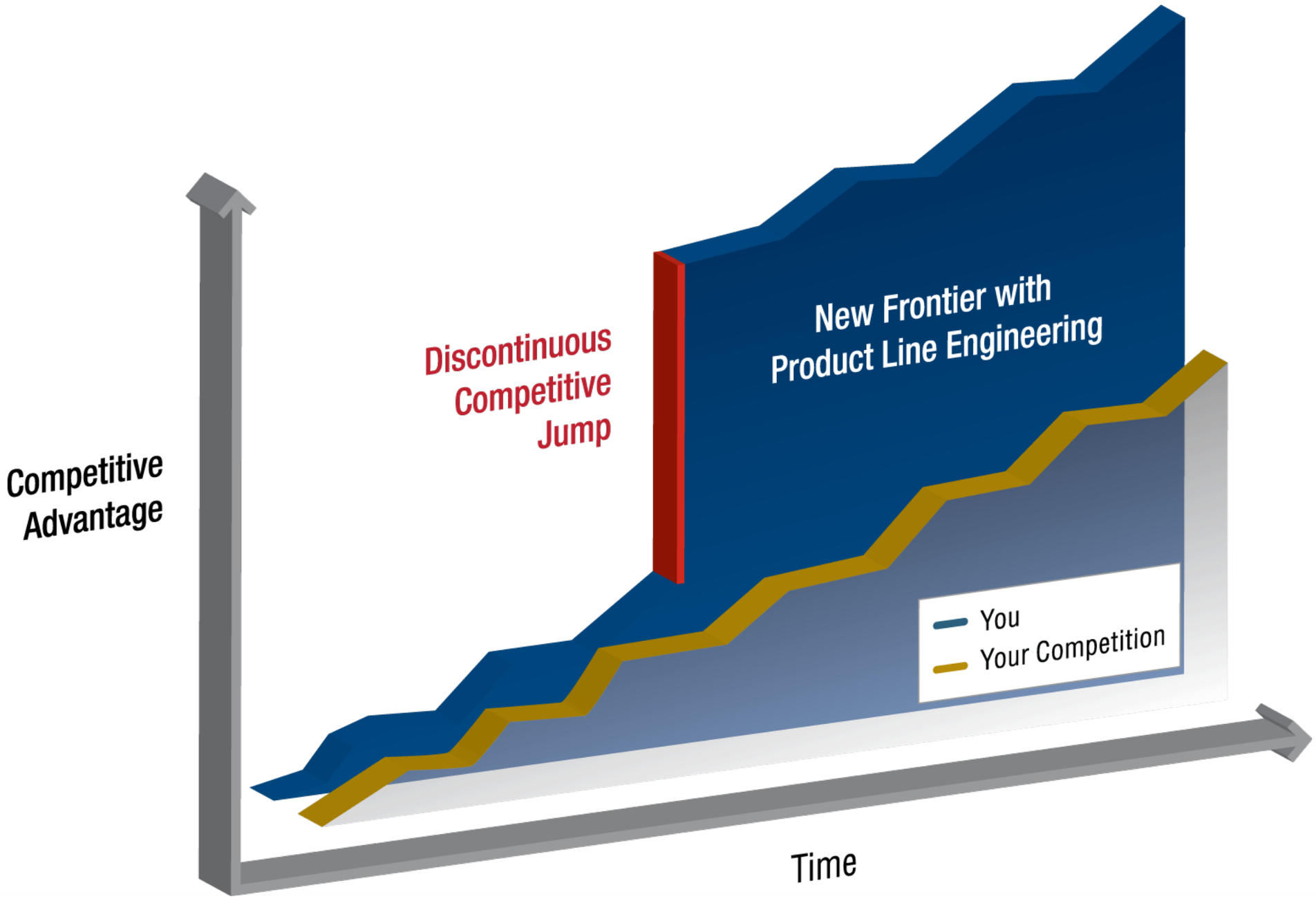


A **product** comes out.

Assets are configured.

Just like a factory.

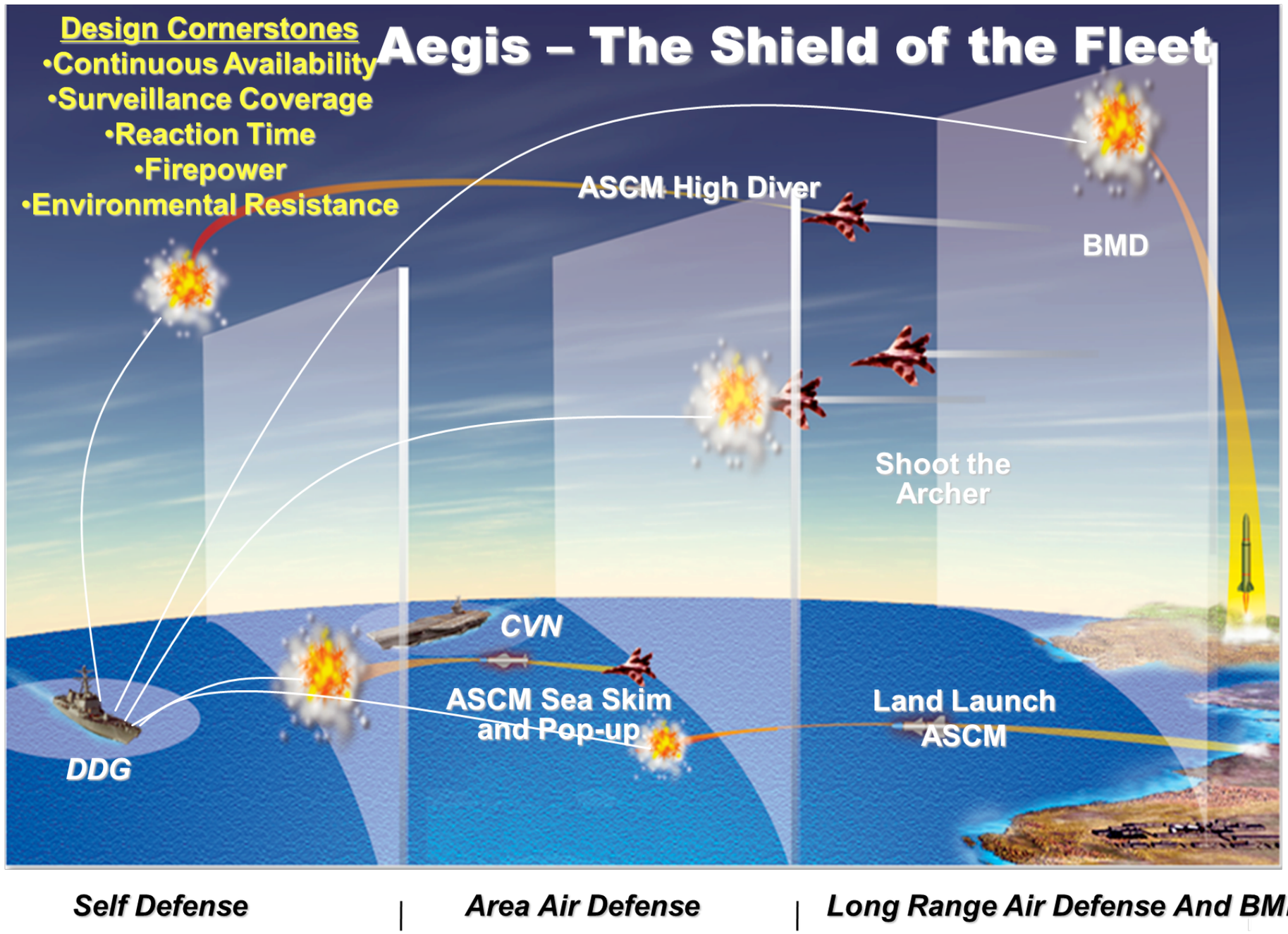
Common Motivation in PLE Success Stories: Competitive Advantage



Organization	Who are they?	What is their product line?	Driving problem	PLE results
LOCKHEED MARTIN	Worlds #1 defense contractor	AEGIS Weapon System	High cost of old approach threatened loss of entire contract	Over 100 ship deployments: $47 million saved per year[1,2,3]
GENERAL DYNAMICS	World's #4 defense contractor	Live Training Transformation, family of large-scale training systems for US Army, Air Force, and Marines	Innovative low-cost solution required to win and keep major contract	Over 300 training range deployments: $660 million saved over 12 years[3,4,5]
GM	World's #1 auto-maker	Largest, most complex product line comprising over 10,000,000 instances	Vehicles taking too long to bring to market; expensive and error-prone processes	Will save "hundreds to thousands of man/years per year, worth tens to hundreds of millions of dollars per year" for one asset type alone[6,7,8]
NetApp	World's #2 data storage company	High-end server storage systems	Unable to accommodate growth in market	2x-5x improvements in scalability, productivity, time-to-market, and product quality[9]
HomeAway	World leader in on-line vacation property rental	Product line of e-commerce web sites hosted in over 200 countries worldwide	Broad variation in sites around the world; needed to go live ASAP	First product went live in 60 days[10]

AEGIS Weapon System

- 1,500 people
- Over 10MSLOC
- 100,000's of requirements
- Multiple Govt. agencies with oversight



Beginnings

- AEGIS began as a family of separate programs.
  - In the early 2000s, there were nine major programs for the U.S. Navy.
  - Each was concerned with one or more ships in the AEGIS family.
  - Each program operated in an isolated manner.
    - There were independent management structures, multiple review teams, varying processes and tools, redundant program plans, different architectures, and multiple independent requirements and source libraries.
    - To create a new program (or baseline), all specifications and source code from a previous program (sometimes still in development) would be copied with new development and maintenance then conducted independently and in parallel.
    - Programmatic and technical decisions could be made for one baseline independent of other baselines, *which was in fact seen as an advantage in terms of schedule and technical flexibility.*

Mid-2000s: Business forces change

- In the mid-2000s, a number of business and technical forces encouraged and enabled AEGIS to become a true product line.
  - The Navy let it be known that paying to fix the same defect (or make the same enhancement) multiple times (once for each program) was cost-prohibitive.
  - The Navy began to push its contractors to follow technical approaches that encouraged reuse, opened up competition, and employed commercial off-the-shelf (instead of purpose-built) hardware and software.
    - This “Open Architecture” initiative touted modular components with published interfaces.
    - In theory, any contractor could bid to provide any of the components.

PLE: Key to competitiveness

- Lockheed Martin made the commitment to become the most competitive of all of the potential contractors.
  - To do this, they explicitly and purposefully adopted product line engineering as their development paradigm.
- By 2009 they
  - Had adopted Gears as the tool to configure their shared assets
  - Were employing the factory-based 2GPLE approach
  - Were building the largest requirements and code baseline in AEGIS history.
  - Had merged separate anti-aircraft warfare and missile defense software components into a common integrated air and missile defense system that could be configured to support any of the ships in the family.

LOCKHEED MARTIN

Integrated weapons system for 100 ships

With Gears...

Entire family managed with a single set of requirements and a single set of source code.

Coast Guard produced new cutter spec in **2 weeks** compared to **3-4 months**

US Navy reports **$47M** cost avoidance per year

US Navy Aegis Cruisers & Destroyers

US Navy Littoral Combat Ships

US Coast Guard Nat'l Security Cutter

Aegis Ships for International Navies

POWERED BY BigLever Software, Inc.

2011: LCS joins the family

- In 2011 Lockheed Martin was awarded the contract for the third member of the Littoral Combat System (LCS) ship class.
  - In previous years, they would have spun this program off on its own development and maintenance trajectory after copying all of the relevant assets
  - Now it was incorporated as a new member of the product line.

2011: Coast Guard joins the family

- In 2011 the U.S. Coast Guard (with Navy encouragement) made the decision to enter the AEGIS product line family.
  - They achieved a much quicker deployment of code and requirements for the Coast Guard – weeks instead of months.
  - This sent a strong message that AEGIS was on the right track with its product line approach.

A Product Line of Product Lines

Operational Readiness

Weapon Control System

AEGIS Training

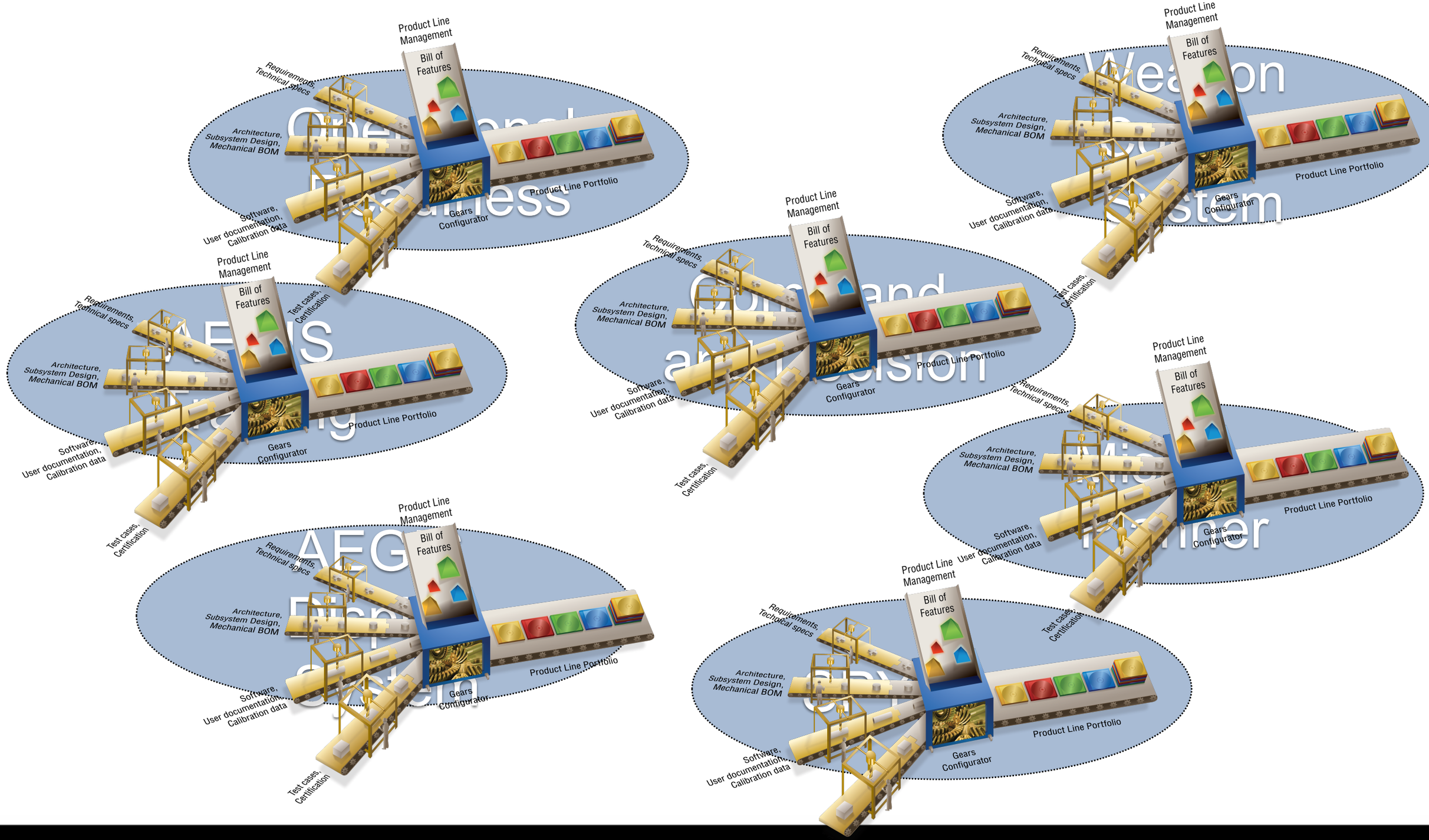
Command and Decision

Mission Planner

AEGIS Display System

SPY Radar

A Product Line of Product Lines



PLE Governance “Pillars

- Regular, predictable build rhythm
  - Three releases a year, January, May, and September
  - This “1-5-9” rhythm brings great stability to the program
- Requirements review cycle
  - March, July, and November (a “3-7-11” rhythm)
  - Joint Lockheed Martin/customer exercise
- Governance boards
  - Structure and processes to ensure clear and consistent direction
  - 3 boards, 3 levels: Strategic, Programmatic, and Technical
  - These boards adjudicate cross-program priorities and activities

Agile comes to AEGIS

- In late 2014, corporate management decided that Agile was the way to strengthen Lockheed Martin's competitive position through quality and affordability.
- This goal was levied on the entire organization.
  - Lockheed Martin's Orion spacecraft project
  - AEGIS Weapon System
  - Many more

Scrum

- Scrum is the specific agile approach chosen by LM.
- Core roles:
  - Product owner
    - Represents the stakeholders and is the voice of the customer.
  - Development team
    - Responsible for delivering potentially shippable increments of product at the end of each sprint.
    - A team is made up of 3–9 individuals who do the actual work.
  - Scrum master
    - Accountable for removing impediments to the ability of the team to deliver the product goals and deliverables
    - Helps ensure the team follows the agreed scrum processes

Sprint

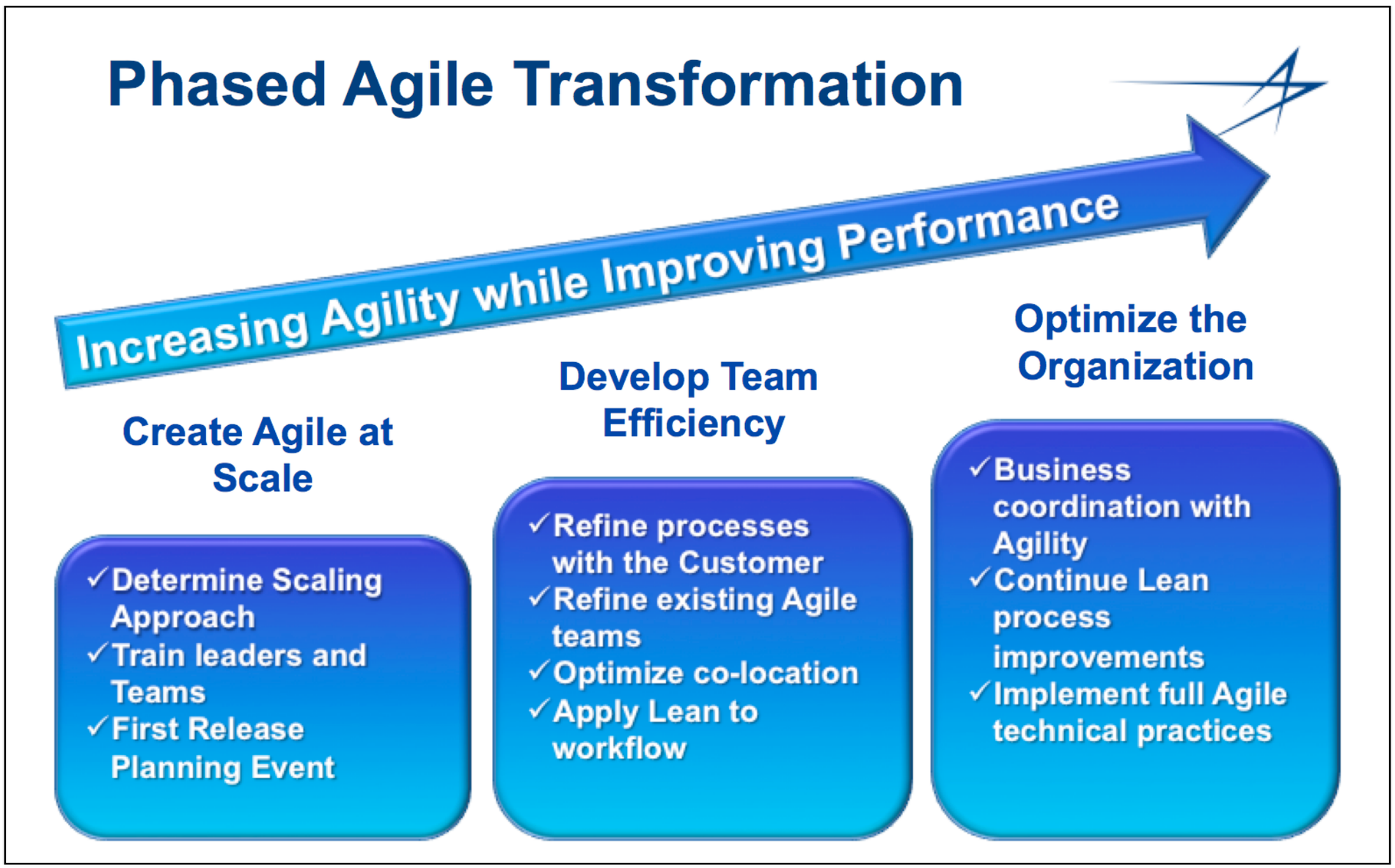
- Basic unit of development in Scrum
- Duration
  - Fixed in advance
  - Normally between one week and one month
  - Two weeks is most common
- Each sprint starts with a sprint planning event
  - Defines a sprint backlog
  - Identifies the work for the sprint
  - Makes an estimated commitment for the sprint goal.
- Each sprint ends with a sprint review and sprint retrospective

Can Agile work for AEGIS? How?

- Agile is, by and large, about
  - a (usually small) single product
  - fielded by a single (usually small) organization
  - by one or more (always small) teams
  - in a series of extremely short iterative development cycles
- In the AEGIS Weapon System product line:
  - The products – plural – are large by any standard, comprising some 10 million lines of code and costing tens to hundreds of millions of dollars.
  - The teams, spread across the products, are sizable. The largest component, for example, involves over 200 engineers.
  - The build cycles, under the product line's 1-5-9 build tempo, are four months long.

Principle: Move incrementally

Three-year transition plan



Principle:
Big Teams Become Small Teams

- Decompose the components' large teams into Scrum teams of size seven to ten.
- The teams, considered together, do the same work as they did before, but the work has also been divided into smaller bite-sized chunks that match the team size and short iteration (sprint) schedule.
- The teams are self-organized by product, mostly around areas of domain expertise or functionality, or around specific elements in the architecture.

Principle:
Big Tasks Become Small Tasks

- Planned work is organized into two-week sprints. A certain number of sprints add up to a release.
- How many sprints?
  - Recall that Lockheed Martin has adopted a four-month release cycle referred to as their “1-5-9” rhythm.
  - Thus, eight sprints constitute a release.
- Here is an elegant case where PLE and Agile work in synchronization with each other.

Principle:
Big Tasks Become Small Tasks

PL release tempo is 4 months = eight 2-week sprints (may shorten later)

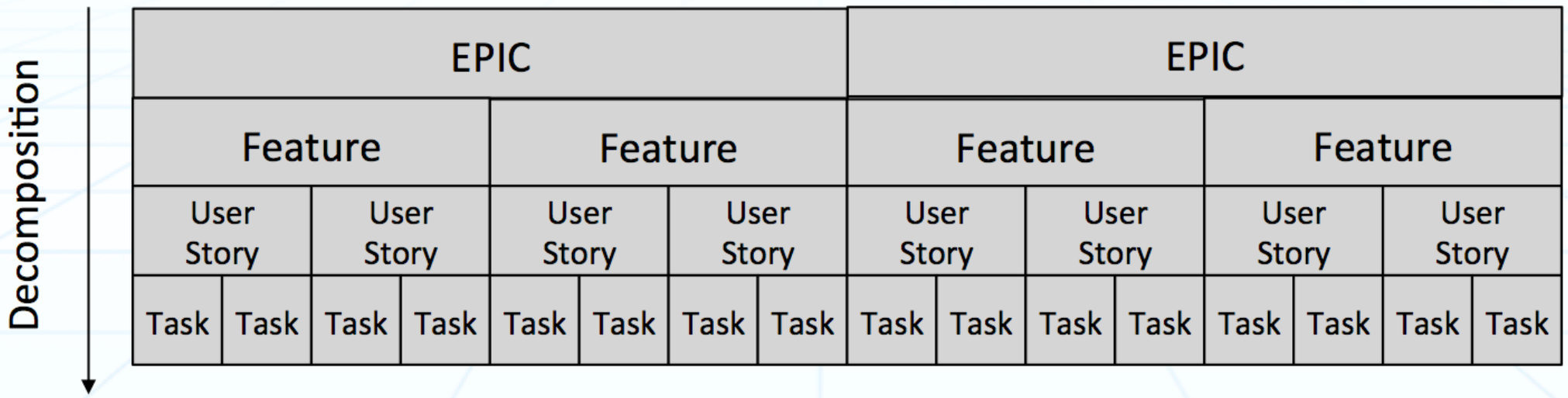
Epic – Spans Multiple Releases, Often Major Capabilities or Types of Work

Large User Story (can have sub-epics)

Feature – Completed within Release, Often tied to Program IMS(s), Often Spans Teams

User Story – Each Team owns their set, Completed within Sprint

Tasks – Generally at the Team Level, 2 to 8 worst case 16 hours



So far so good. But…

- Instead of 9 large teams, we now have ~100 small teams.
- The twelve major configurations are, at any point in time, each in a different phase of development.
  - For some, the major activity for next release is writing concept papers.
  - For others it might be writing code.
  - For still others, it could be testing.
- With 100 small teams, ensuring everyone is working towards common large-scale goals, without devoting all their time to management overhead tasks, becomes a critical concern.
- This issue is Ground Zero of where PLE meets large-scale multi-project Agile.

Enter the Scaled Agile Framework (SAFe)

- SAFe
  - a “knowledge base for implementing agile practices at enterprise scale.”
  - It defines the “individual roles, teams, activities, and artifacts necessary to scale agile from the team, to teams of teams, to the enterprise level.” *

- Atlassian’s Jira software project and issue tracking tool
  - Tracks dependencies
  - Helps plan and manage sprints
  - Allows every team member to be involved in the planning
  - Lets everyone see and be aware of task dependencies
  - Leads to more accountability, ownership, and buy-in to the team’s tasking

* Scaled Agile Framework, www.scaledagileframework.com

Enter the Scale Agile Framework (SAFe)

- Teams vote collaboratively to agree on the scope of a piece of work.
- If priorities are not being met, then the higher-level governance structures come into play to adjudicate conflicts and re-orient the work.
- A team with more resources and less schedule pressure can “farm out” their services to other teams in need.

Proof Point

- Last September: Product line's first full-up release planning event.
  - All 100 teams, representing all 800 engineers, participated in this 3-day planning exercise.
  - Produced a coordinated plan for the next eight weeks.
  - Jira provided the way to manage links among program and product work content.
  - All 800 engineers (many geographically dispersed) could see this picture unfolding in real time.

Observations so far:
Agile + PLE is paying off

- A team works under Scrum. Teams coordinate via SAFe. Both Scrum and SAFe are applied under the over-arching PLE governance structures.
- An agile approach to requirements allows a “just enough, just in time” approach with just enough detail to ensure a project’s technical and economic feasibility.
- Agile provides for more level loading and resource allocation.
- Lessons about teaming are emerging.
  - Teams should be co-located, if possible.
  - There’s no place for mediocre performers to hide on a small team.
- Each sprint culminates in a review or customer demo, which establishes trust with customer and stakeholders.

www.biglever.com