# Topics

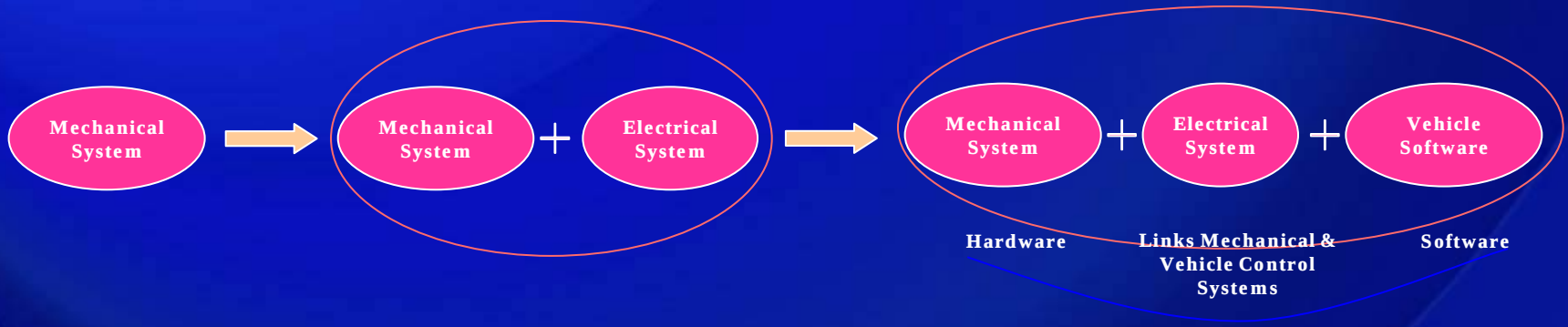
- OEM View
  - Industry Drivers
  - A Dilemma
  - Needs
  - Data Integration

# Industry Drivers

- Vehicle Complexity
  - LOC
  - System Dynamics / Interaction of Control Systems
  - Multi function ECUs
- Changing role of the Vehicle
  - Personalized to the Individual Customer
  - Internalized Health, Diagnosis, Warning, and Repair
  - Integration to the pervasive computing eco-system
- Emerging / Future Vehicle Technologies
  - Drive by Wire
  - Crash Avoidance
  - Multi and Concurrent Occupant Interactions
  - Adaptable Configurations
- Competitive Differentiation through Vehicle Electronics and Software

# Industry Drivers

- Quality and reduced cycle times
- Changing character of the vehicle from a mechanical system to a software controlled system




- Competition

# Needs

- *Requirements/Specification Management*
  - Manage discrete requirement changes and trace to delivered product(s)
  - Relate requirements within and across specifications
- *Adapt Model Based Systems Engineering Practices*
- *Verification and Validation Efficiencies*
  - Verify software function/behavior to specification
  - Quickly and completely test software functionality (positive testing)
  - Implement consistent test vectors across MIL, SIL, TCIL, and Product
  - Guard against emergence of hidden bugs (negative testing)
- *Supplier Management*
  - Ability to establish a collaborative environment supporting
    - Requirements exchange
    - Product delivery, evaluation, and acceptance

# Needs

- *Technology Management*
  - Smoothly manage upgrades in technology so that product development processes are not interrupted
  - Manage integration of new technology into the vehicle
  - Adapt new software tools
- *Security Management*
  - To apply well defined In-Vehicle authentication and authorization controls
- *Globalization*
  - Manage deployment for global operations
  - Distribution/Access Management
- *Product Maturation*
  - Capture the history of the product from inception to product
  - Traceability
  - Change control management
- *Release and Support Management*
  - Ensure software version compatibility for in-field upgrades

# Data Integration

Interoperability strategy for internal and external systems. The interoperability strategy knits together systems allowing seamless access and exchange of data and information

# Invirtech

**We are a start-up company specializing in solutions for managing the development and testing of embedded control systems.**

## Objectives:

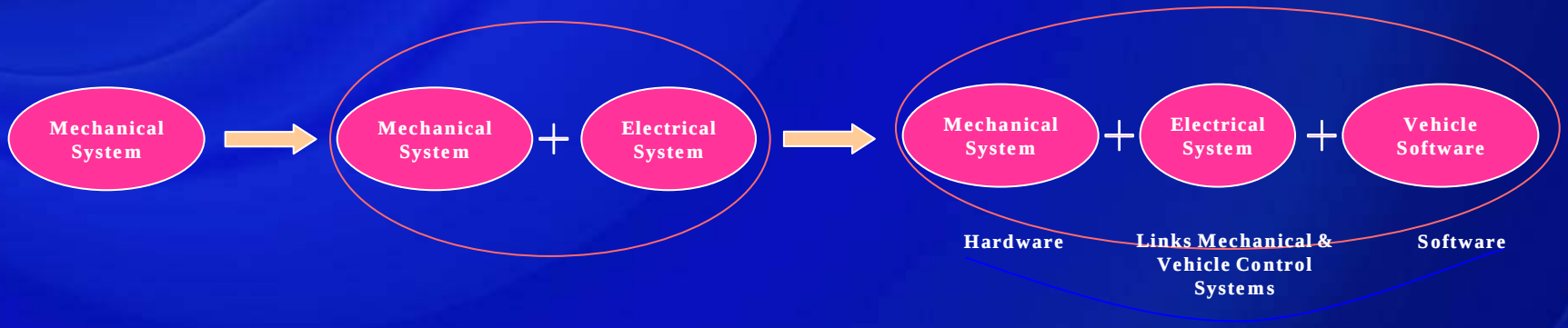
- **Integrate best-of-breed technical tools into an automated, database-driven solution.**
- **Integrate with corporate PDM/PLM systems.**
- **Provide consulting on testing of open/closed-loop real-time systems.**

# Topics

- Observations
- Solutions(?)
- ECSW Development Process
  - Virtual testing
- Data Synergy
- Conclusions

# Observations

- Vehicle is a software system




- Complexity, complexity, complexity
- Low level and high level systems all competing to control the same resources
- Protect for bizarre/unexpected behavior (Thunderbird, Taurus examples)

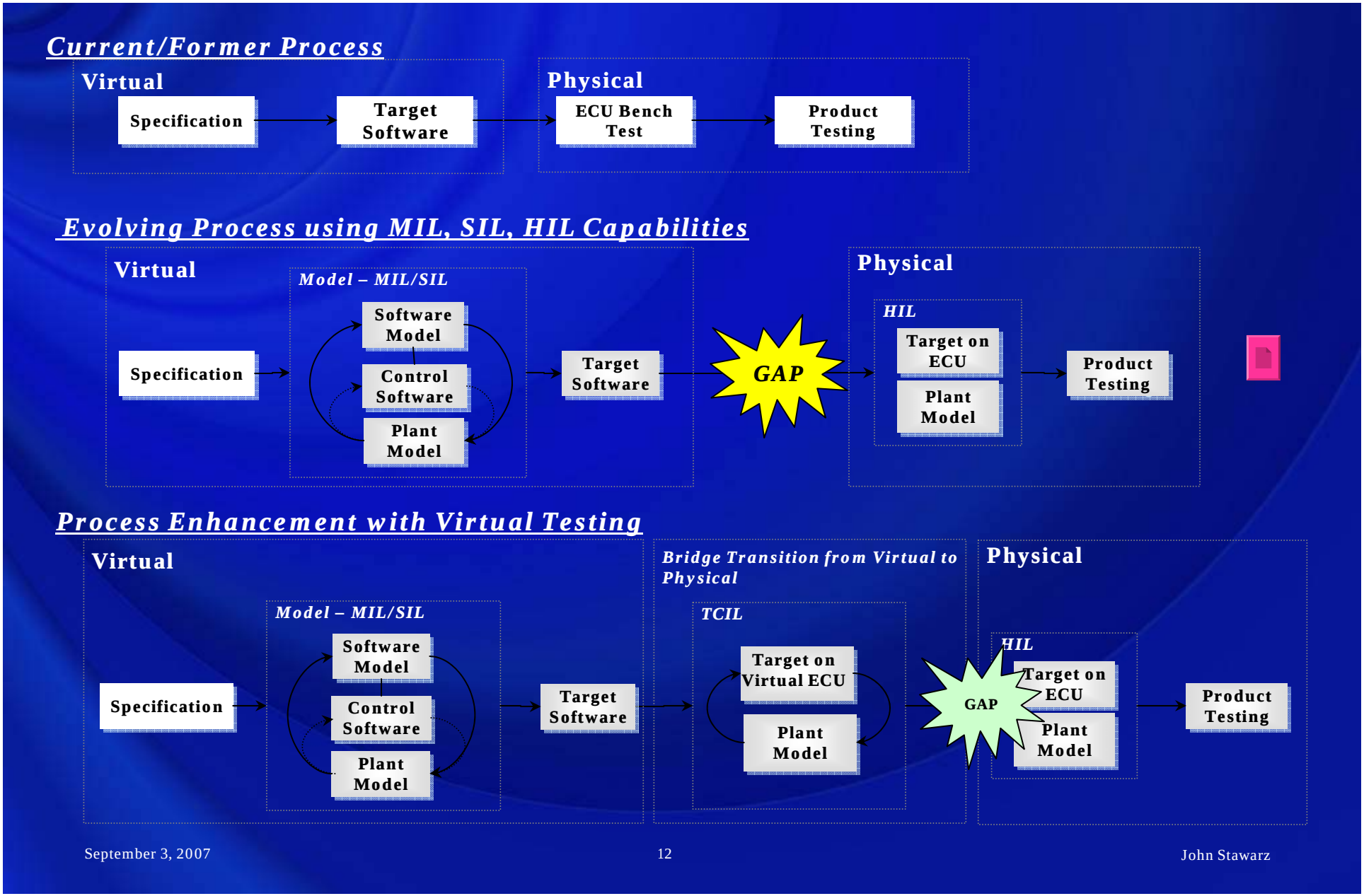
# Observations

- Need to make validation more efficient
  - Too time consuming
  - Initiated late in process
  - Costly (use of hardware, prototypes, etc.)
- Ensure robust, error free(?) software delivery within current/future timing constraints
- Key engineering areas of concern
  - Requirements/Specification Management
  - Model Based Design Engineering
  - Validation Processes
  - In Production Support

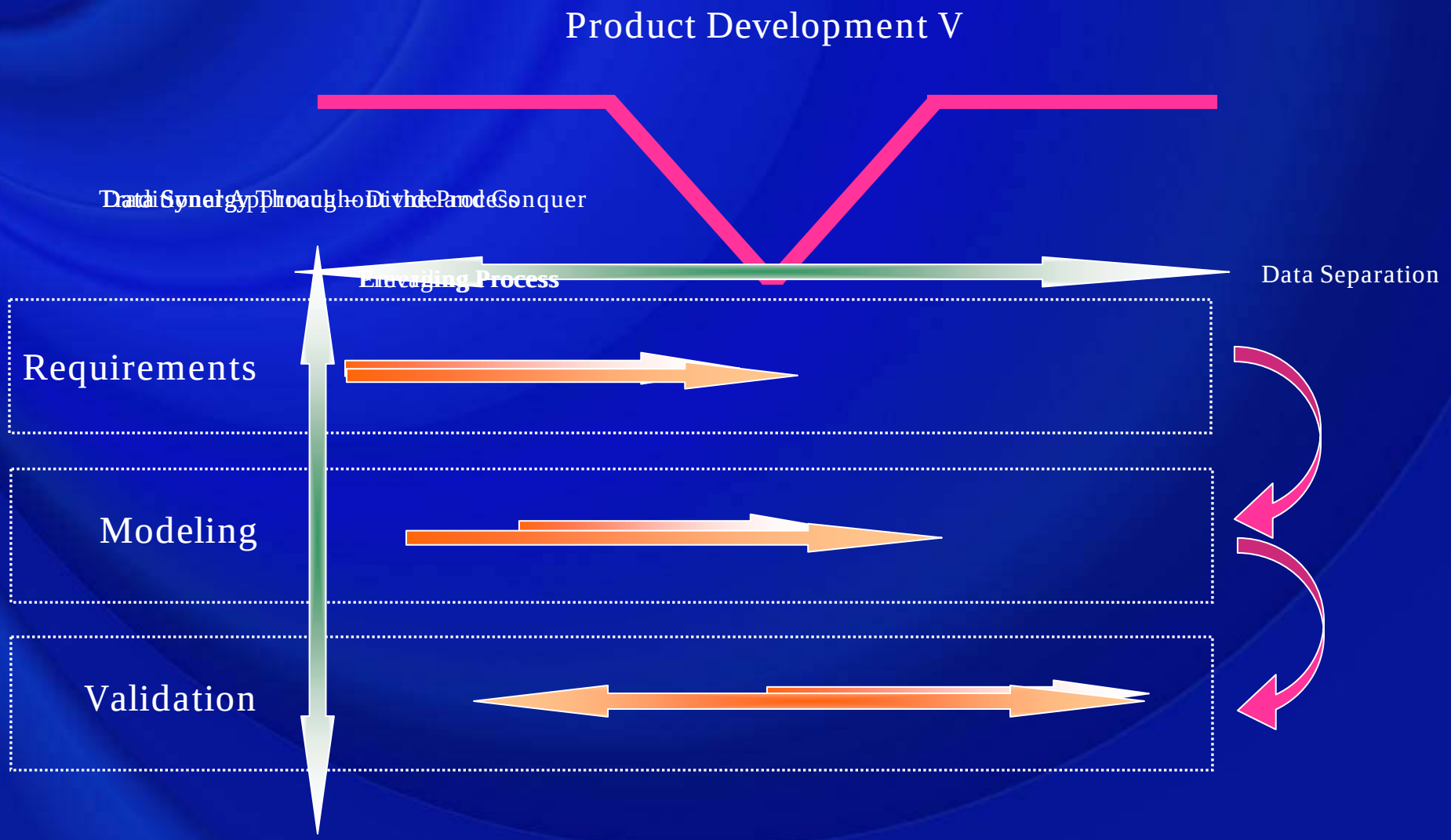
# Solutions Ideas

- Process Enhancement
  - Apply Virtual Testing
  - Data driven validation – test assembly, configuration
  - Re-use test vectors/procedures across MIL, SIL, HIL
- Data Synergy
  - Re-evaluate what it means to capture and use data throughout the process

# ECSW Development Process



# Data Synergy



# Conclusions

- A solution should span the process and bridge the virtual to physical gap
- Need to efficiently iterate test cycles in order to confirm software behavior
  - Positive Testing and Negative Testing
  - Guard against future emergence of hidden software bugs
- Ensure data transition consistency and re-use as the control system morphs from a set of requirements, to a set of models, to real hardware and software
- Broaden testing scope to include interactive and investigative analysis of alternative architectural designs

# Virtual vs. Physical

## Virtual

- Advantages
  - Lower Cost
  - Time, easily supports changes
  - Code generation?
- Disadvantages
  - It still is a model
    - Variability
    - Time to build model
  - Need plant and function models

## Physical

- Advantages
  - It's the real thing, target code, target hardware
- Disadvantages
  - Higher Cost
  - Validation operation
    - Need prototype products
    - (De)Instrument
    - Calibration & Maintenance
    - Scheduling
    - People
    - Environment
  - Testing completeness (positive / negative testing)