

Kevin Thompson
Ph.D. Consulting

Modernizing Hardware Product Development

**Agile approaches for improving delivery
predictability in complex engineering organizations**

Executive Summary

Hardware Product Development Is Systematically Failing

90%

of launches
delayed by
late-stage
design changes

11 of 12

Ericsson
hardware
projects
studied were
late

30%+

of engineering
time
consumed by
rework

2x

schedule
growth
possible
in complex
systems

We help hardware organizations reduce late-stage surprises and improve delivery predictability using Agile approaches adapted specifically for hardware development.

Based on peer-reviewed engineering-management and product-development research.

Leadership and Expertise



Kevin Thompson, Ph.D.
Agile Transformation Consultant

Hardware organizations do not fail because engineering teams lack talent. They fail because complexity, uncertainty, and integration risk overwhelm traditional management systems.

Kevin Thompson, Ph.D. (Physics, Princeton) has

- Scientific and engineering background in aerospace at NASA and LLNL
- Led Agile consulting and transformation initiatives across more than 100 engineering organizations worldwide.
- Pioneered Agile approaches for hardware-development organizations

Specialization Areas

- Hardware product-development systems
- Cross-functional engineering coordination
- Agile transformation for complex physical products
- Delivery predictability and risk reduction
- Product-development workflow modernization

The Structural Problems in Hardware Product Development

Late Discovery of Problems

- Integration issues surface too late
- Customer feedback arrives late
- Risks become visible only after escalation

Excessive Cost of Change

- Late-stage design changes are expensive
- Rework consumes engineering time
- Validation cycles are slow and costly

Coordination Complexity

- Hardware and software teams move at different speeds
- Cross-functional dependencies create friction
- Organizational silos delay decisions

Slow Learning Cycles

- Teams spend months before validating assumptions
- Sequential workflows delay feedback
- Planning assumptions persist too long

Traditional hardware-development systems optimize for execution efficiency rather than rapid learning and adaptation.

Why Traditional Product-Development Management Breaks Down in Hardware

Traditional Project Management

- Assumes plans can be predicted upfront
- Optimizes for milestone compliance
- Encourages late integration
- Hides uncertainty until late stages

Typical Outcomes

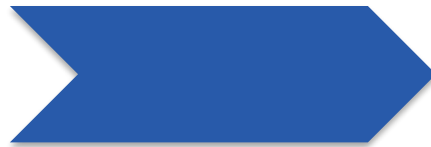
- Expensive late-stage design changes
- Cross-functional misalignment
- Schedule slips discovered too late

LOW UNCERTAINTY

Traditional planning works well

HIGH UNCERTAINTY

Agile approaches outperform predictive planning



11 of 12 Ericsson hardware-development projects studied were late

Agile Hardware Development

- Assumes uncertainty is normal
- Uses rapid validation and iteration
- Surfaces integration issues early
- Enables continuous prioritization

Typical Outcomes

- Earlier visibility into technical risk
- Reduced rework and churn
- More adaptive delivery planning

Hardware development is dominated by uncertainty, integration risk, and expensive late-stage learning.

What Causes Hardware Delays: Ericsson AB

Project	Timedelay [%]
A	17,1
B	80,2
C	89,4
D	27,9
E	235,7
F	36,2
G	24,1
H	16,9
I	33,5
J	72,2
K	25,9
<i>Average</i>	<i>59,9</i>
<i>Median</i>	<i>33,5</i>

11 of 12 projects studied were late

Biggest Factors behind delays:

1. Immature technology from vendors (undue optimism)

2. Inconsistencies in how work is done from place to place

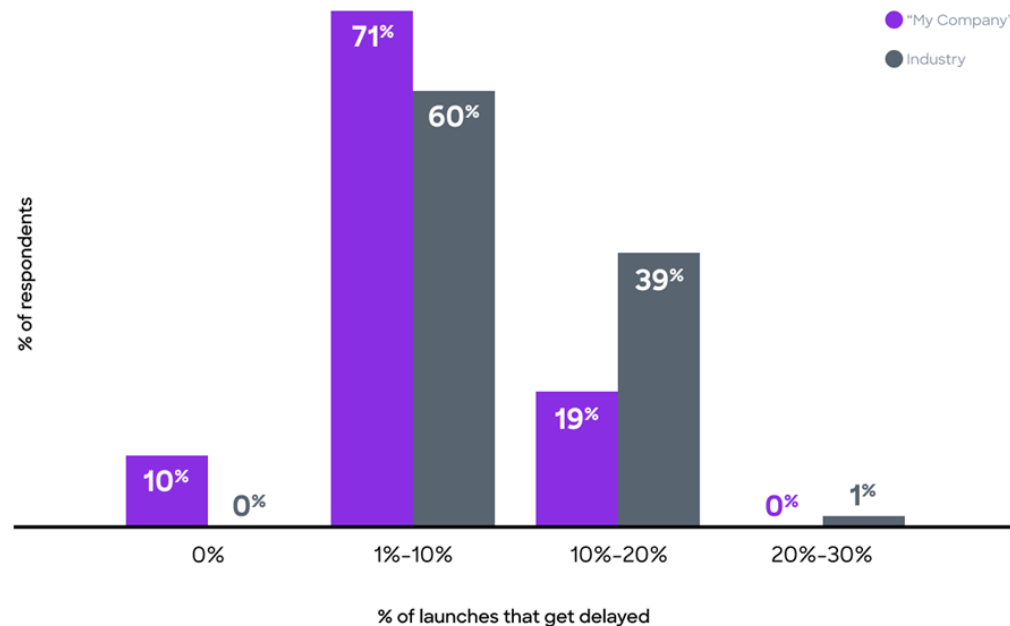
3. Cross-functional communication failures

Ericsson AB makes telecommunications and networking equipment. This study examined 11 late projects in depth, to identify factors responsible for delays in completion.

What Causes Hardware Delays: CoLab Report

- 90% of all launch delays were due to late-stage design changes, after design was thought to have been finalized

Percentage of all launches delayed by late stage design changes



- Labor-intensive investigation to identify risks omitted during design stage
 - False economy

90% of companies delay product launches © 2025 CoLab Software

What Actually Changes in an Agile Hardware Organization

Traditional Hardware Development

- Long sequential handoffs
- Integration happens late
- Teams optimize locally
- Problems discovered near launch
- Fixed plans despite uncertainty



Agile Hardware Practices

- Cross-functional teams
- Frequent integration and validation
- Continuous prioritization
- Smaller development batches
- Rapid feedback loops
- Visible work and risk



Organizational Outcomes

- Earlier discovery of problems
- Reduced rework and churn
- Better schedule visibility
- Faster decision-making
- Improved coordination
- More adaptive delivery

Agile does not eliminate uncertainty in hardware development. It changes when uncertainty becomes visible.

How We Help Hardware Organizations Improve Product Delivery

Reduce Late-Stage Surprises

Challenges

- Late integration risk
- Expensive late-stage changes

What We Do

- Earlier validation loops
- Agile hardware operating models

Outcomes

- ✓ Earlier visibility into technical risk
- ✓ More predictable delivery

Improve Cross-Functional Coordination

Challenges

- Engineering silos
- Coordination breakdowns

What We Do

- Cross-functional Agile implementation
- Leadership alignment

Outcomes

- ✓ Stronger cross-functional execution
- ✓ Faster organizational decision making

Modernize Product Development Management

Challenges

- Poor visibility into delivery risk
- Rigid planning systems

What We Do

- Adaptive planning systems
- Agile program management

Outcomes

- ✓ More adaptive prioritization
- ✓ Improved execution transparency

Build Internal Agile Capability

Challenges

- Inconsistent Agile practices
- Lack of internal expertise

What We Do

- Executive workshops
- Team coaching

Outcomes

- ✓ Sustainable organizational capability
- ✓ Consistent cross-functional execution

Why Hardware Development Requires a Different Agile Approach

Generic Agile Consulting

Optimized for software environments

Assumes low-cost iteration

Limited understanding of:

- Physical integration
- Manufacturing constraints
- Long validation cycles
- Hardware/software synchronization
- Late-stage cost of change

Typical Result

✗ Agile rituals without operational improvement

✗ Software-oriented mindset is a fundamental mismatch to hardware development



Hardware-Specific Agile Transformation

Designed for complex physical-product environments

Accounts for:

- Integration risk
- Costly late-stage changes
- Supply-chain realities
- Long hardware iteration cycles
- Cross-functional engineering dependencies

Typical Result

✓ Earlier risk discovery and improved delivery predictability

Hardware development is not “software development with longer lead times.” It is a fundamentally different operating environment, with a much higher cost of change.

Engagement Models

Executive Workshops

- Leadership alignment and planning sessions

Organizational Assessments

- Identify workflow, delivery-risk issues, recommend solutions

Team Enablement

- Training as appropriate for different groups
- Guidance throughout transformation process



Achieved better cross-functional alignment

- Marisa Richardson, Senior Manager: "There's an increased level of expertise in the overall product by the whole team because they work so closely developing and reviewing sprint deliverables, as opposed to working in functional silos that may not be in sync."



Improved accountability and delivery

- Paige Fordice, Portfolio Manager: "Overall, accountability has increased, we're meeting our quarterly demo requirements, and we're meeting all our key project deadlines."



Created a predictable software-release cadence

- John Sadler, VP and General Manager: "For the first time in this Division's recent history, we're about to deliver a high-quality product with the feature set we said we would, on time.... That's dramatic."

Where to Start

1. Executive Discovery Conversation

Understand current product-development challenges

- Delivery predictability
- Cross-functional friction
- Late-stage surprises
- Integration and scaling issues

Goal

Identify the highest-leverage improvement opportunities



2. Organizational Assessment

Evaluate the current development system

- Workflow and coordination
- Planning and prioritization
- Risk visibility
- Leadership alignment

Goal

Expose systemic constraints and improvement targets



3. Transformation Roadmap

Build a practical transition strategy

- Pilot initiatives
- Leadership alignment
- Team enablement
- Scaling approach

Goal

Create an adaptive roadmap tailored to the organization

The goal is not to “implement Agile.” The goal is to improve how complex hardware products are developed.