DENSO
Crafting the Core

An Agile Development Method for Multiple Product Lines of Automotive Software Systems

Kengo Hayashi*
DENSO CORPORATION
kengo_hayashi@denso.co.jp

Mikio Aoyama
Nanzan University
mikio.aoyama@nifty.com

Scenario

1. Background
2. Challenges
3. Approaches
4. Twofold Iterative Process
5. Process Design and Process Assets
6. Independence Analysis of Variability
7. Application and Effectiveness
8. Discussion and Future Works
9. Conclusions

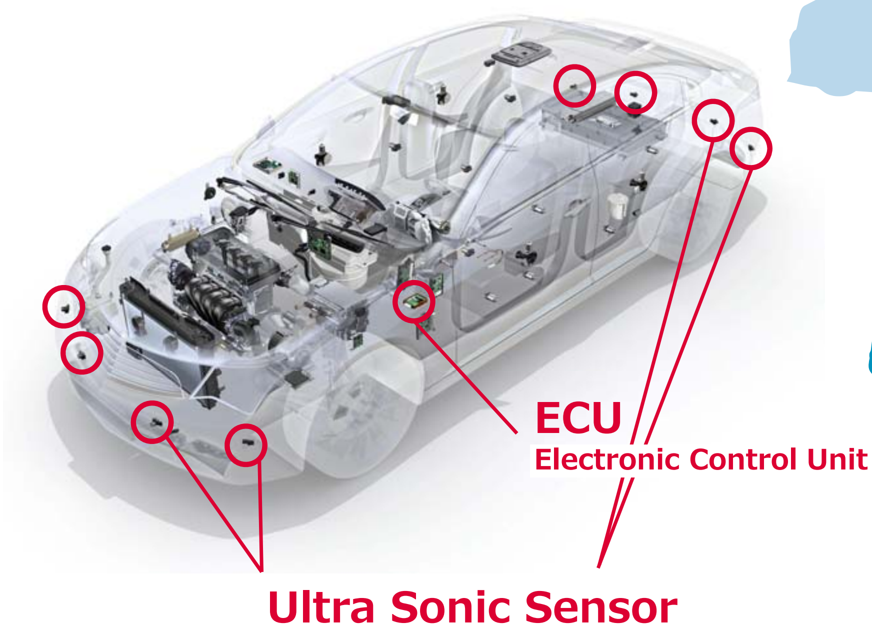
1.

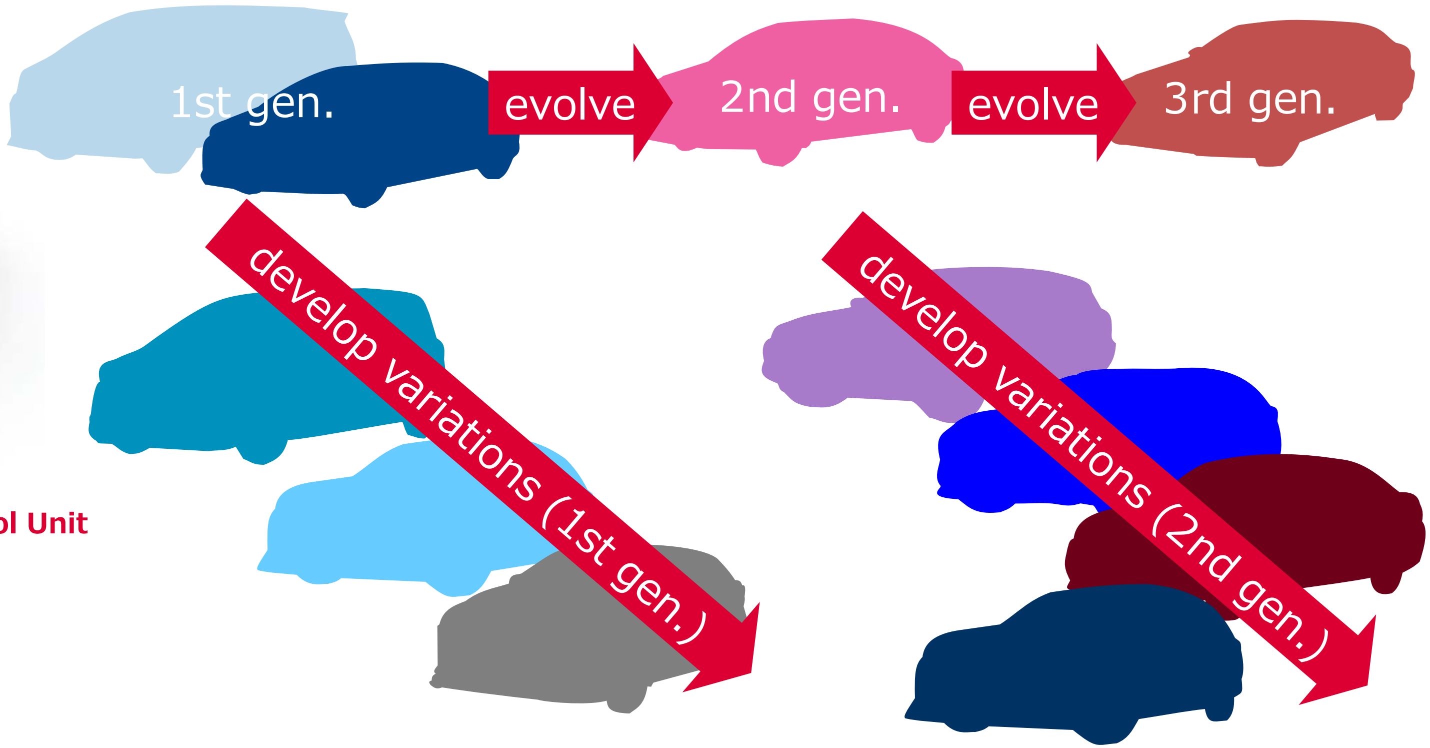
Background

DENSO
Crafting the Core

Parking Support System with Ultra Sonic Sensors

Product evolution is fast and expected to expand into many vehicle variations

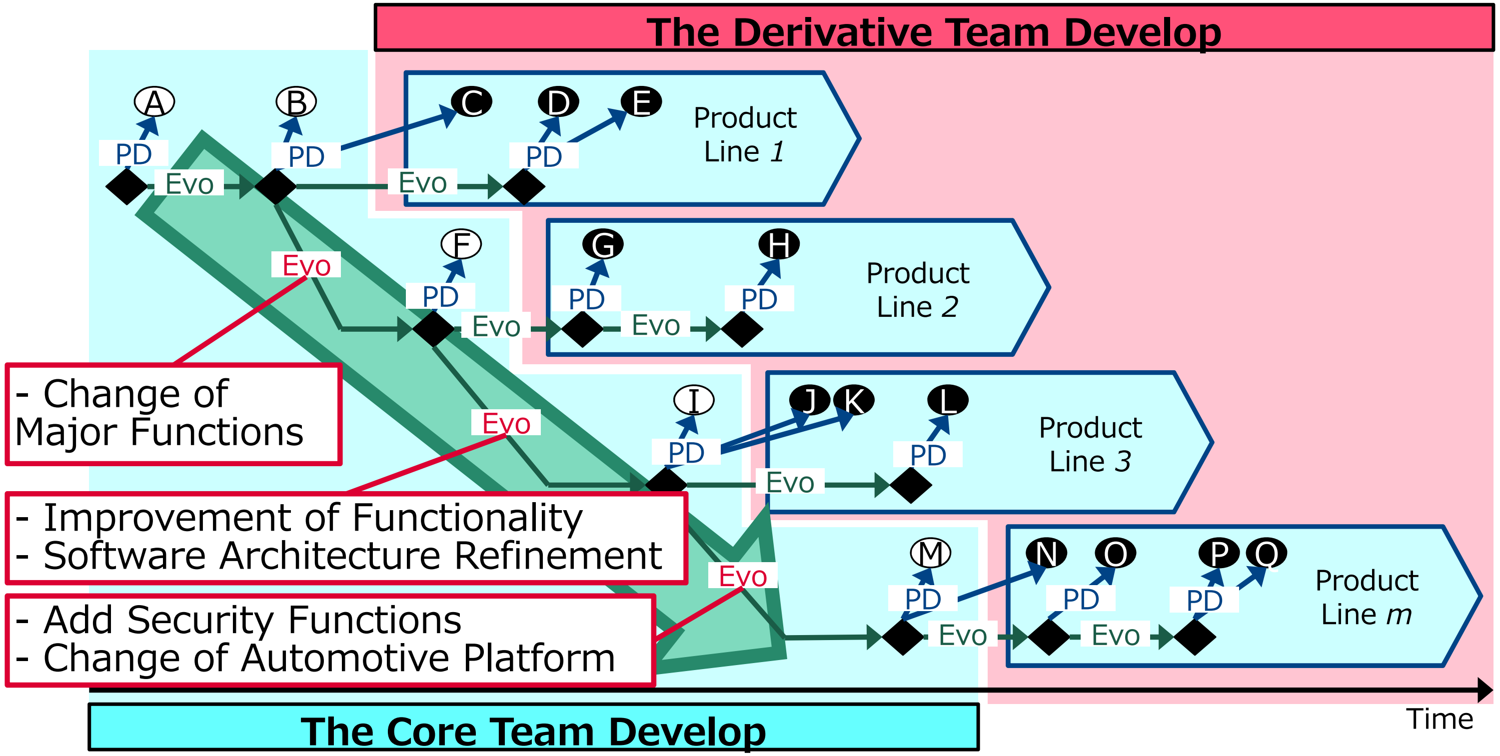


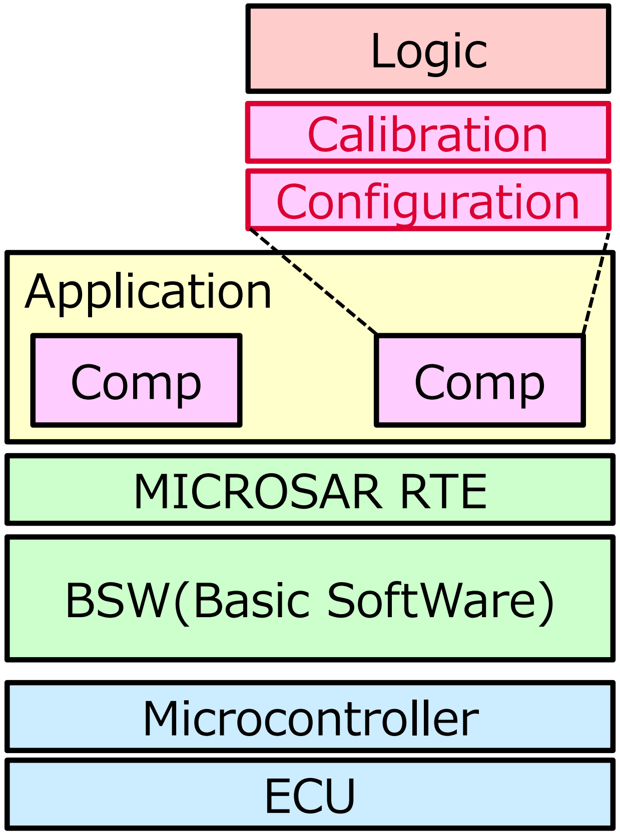


Need to deal with both variability and agile evolvability concerns

Development Organization and Software Architecture

Collaboration between two divisions in the practice of SPLE





Legend PD : Product Derivation Evo : Product Line Evolution
◆ : Versions of Core Assets ○ : Core Product ● : Derivative Product

The derivative team concurrently develops along with MPLE

2.

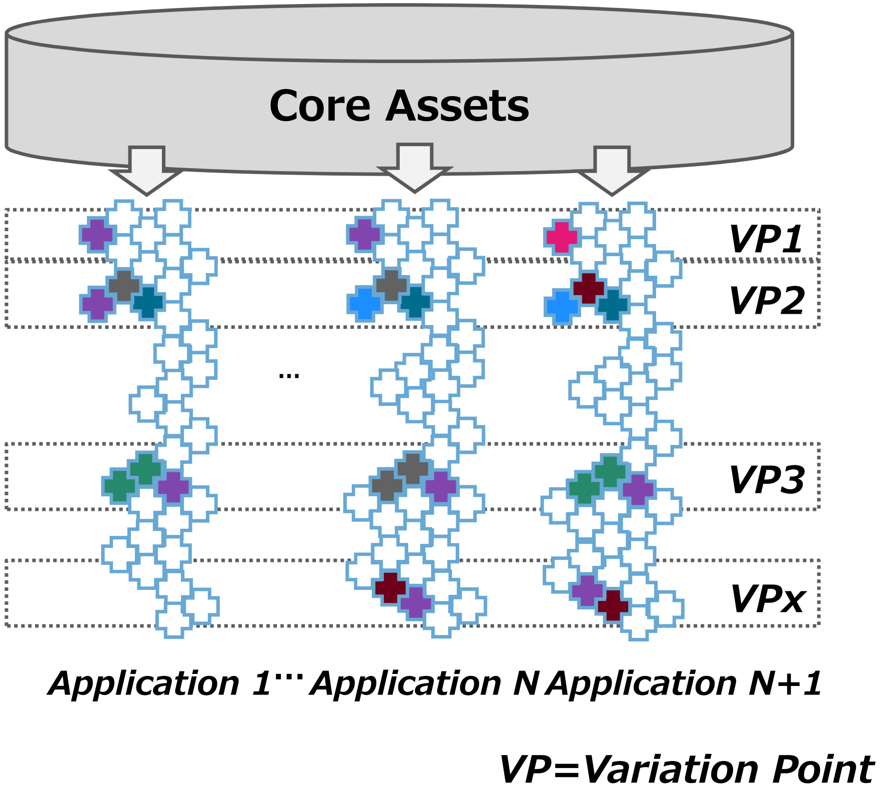
Challenges

DENSO
Crafting the Core

SPLE (Software Product Line Engineering)

SPLE deals with diversity by separating development into:

- **Domain engineering**
- **Application engineering**



In practice

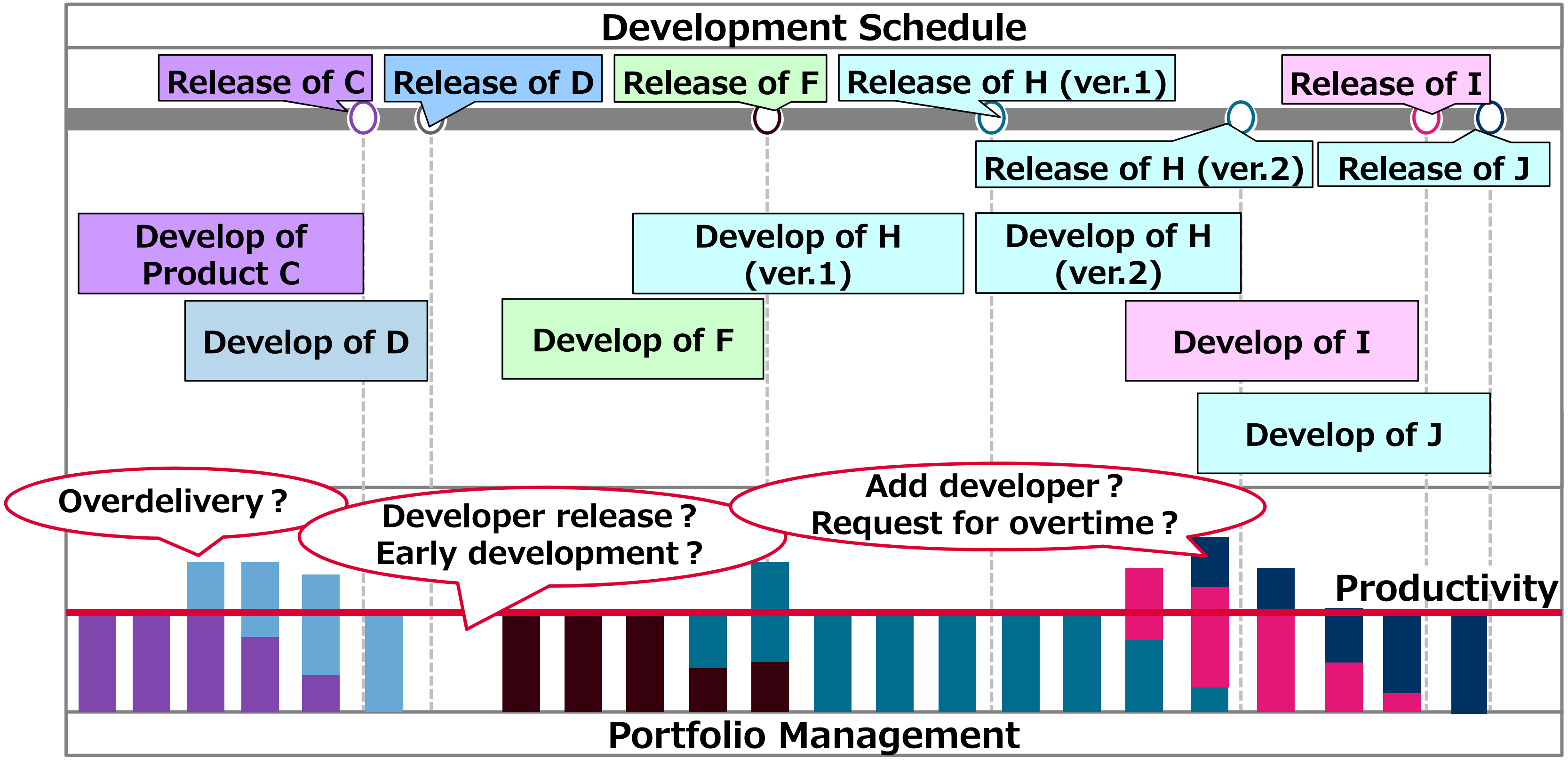
Some issues to be solved for SPLE

- **Incomplete architecture**
- **Evolutionary change over multiple generations**
- **Lack of test automation**

The cost of application engineering done not become 0

MPLE (Multiple Software Product Line Engineering)

The derivative team develops multiple products concurrently



Portfolio management becomes more complicated, and risk increases

Issues in Portfolio Management of MPLE

Portfolio	Issues
Developer Resources	It is painful to increase or decrease developers since domain skills are necessary
Core Assets	It is confusing to modify the same asset at the same time
Development Environment	A limitation of the resource of test environments causes a bottleneck of testing

Realize effective portfolio management for stable development

3.

Approaches

DENSO
Crafting the Core

Overview

Issues

Developer Resources

It is painful to increase or decrease developers since domain skills are necessary

Core Assets

It is confusing to modify the same asset at the same time

Environments

A limitation of the resource of test environments causes a bottleneck of testing

Approach

Scrum with multi projects

Grasp more accurate on
- Time-box
- Story point

Incrementally into subdivisions
- Feature Oriented

Solution

Twofold Iterative Process Structure

Support

- Process Design
- Process Assets
- Dependency Analysis of Variability

Introduce Scrum's framework to enhance portfolio management

4.

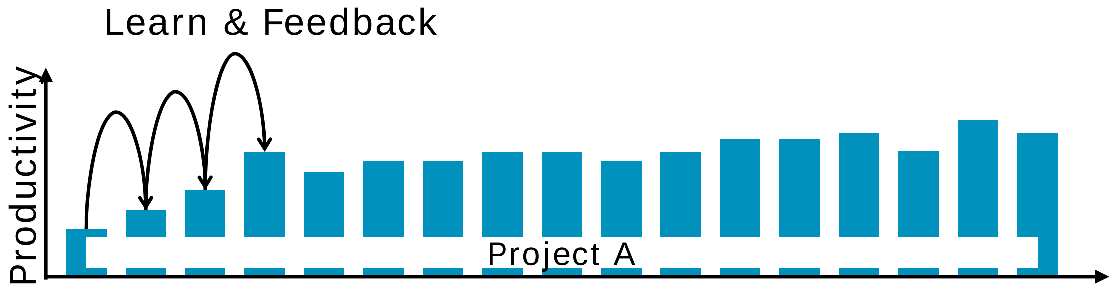
Twofold Iterative Process Structure

DENSO
Crafting the Core

Twofold Iterative Process Structure

<Conventional Agile Method>

Learn productivity and feedback to the plan with single iteration loop

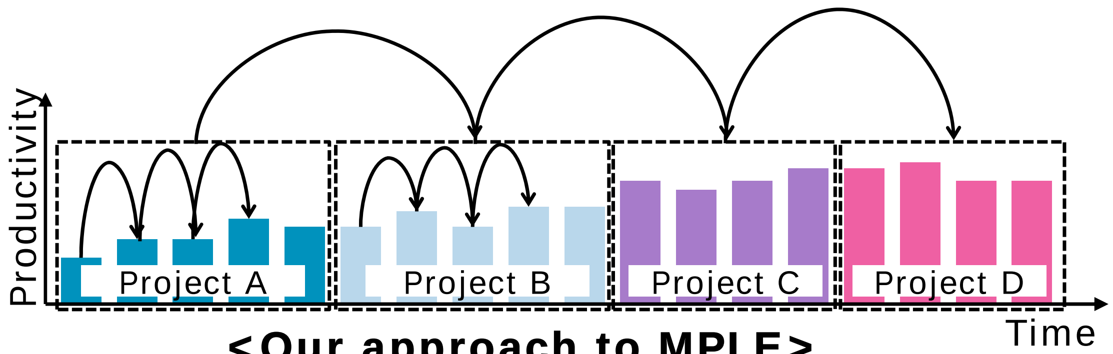


<Conventional agile management method>

<Twofold Approach>

Minor iteration loop within a project

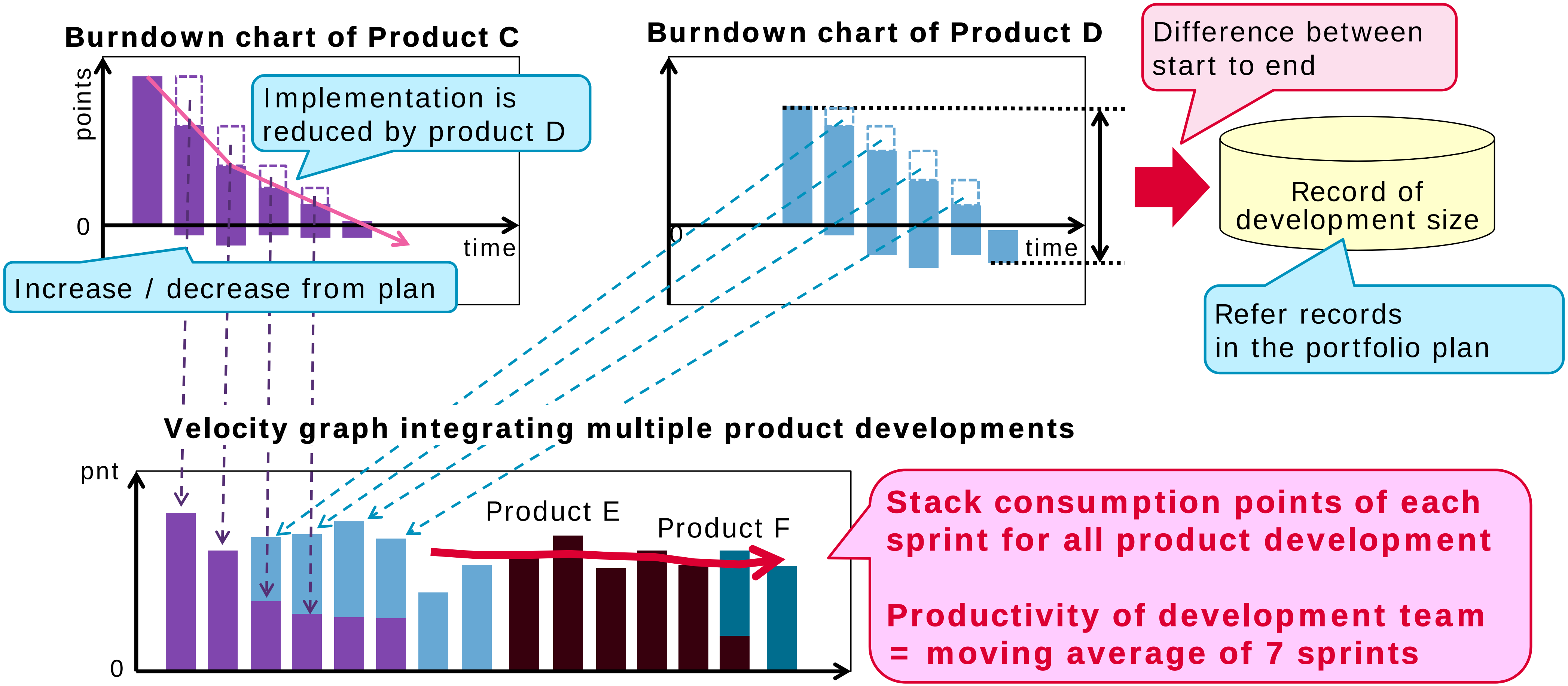
Major iteration loop across the multiple projects



<Our approach to MPLE>

Two fold feedback within and across iterative multiple projects

Monitoring Development Size and Productivity



Integrated measured velocity = planning guidelines

5.

Process Design and Process Assets

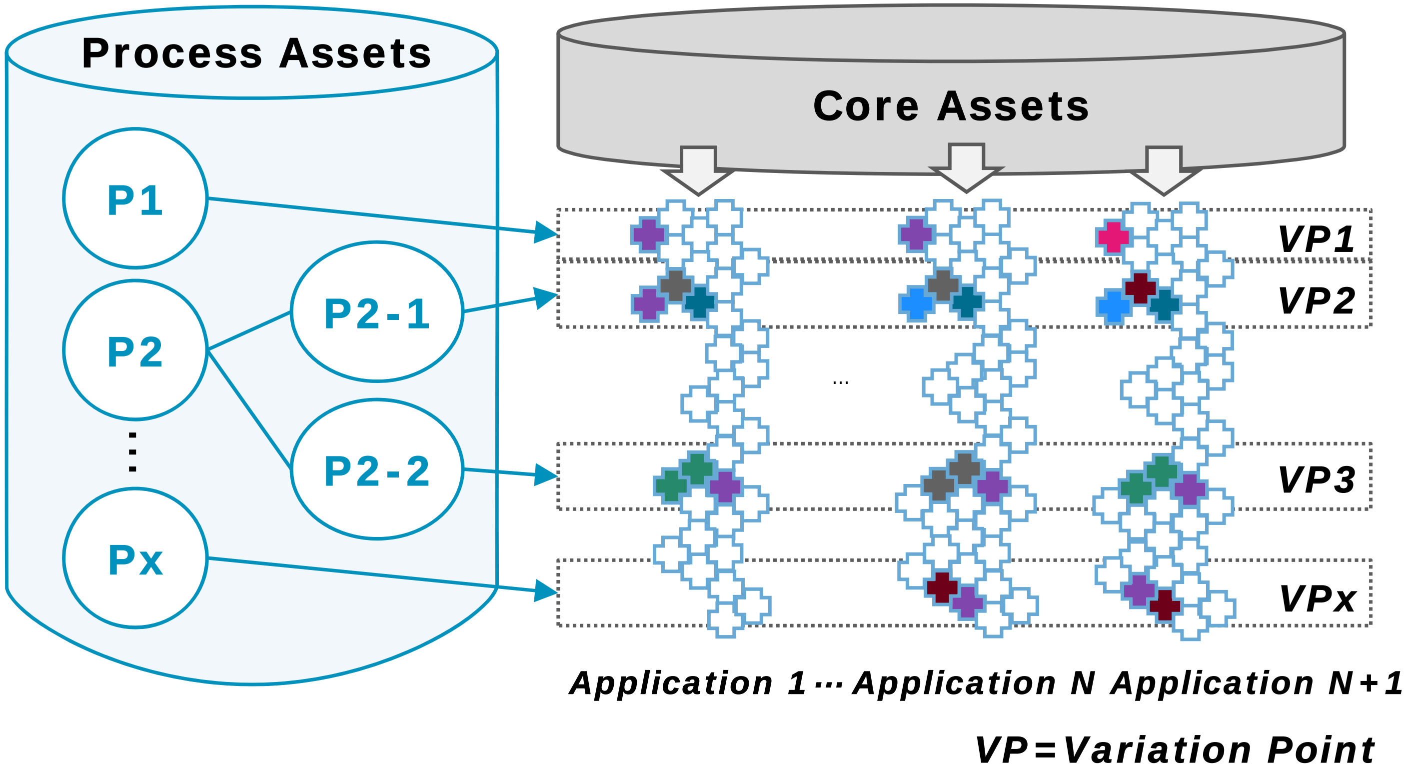
DENSO
Crafting the Core

Process Design and Process Assets

The process can be iteratively reused over the engineering of multiple applications

Design and Reuse Processes as Process Assets:

- Tailoring result
- Work procedure
- Configuration of artifact



Designed Process maintains the learning effect across the projects

6.

Dependency Analysis of Variability

DENSO
Crafting the Core

Dependency Analysis of Variability

Dividing method and order constraints of development determined by dependency of variation points

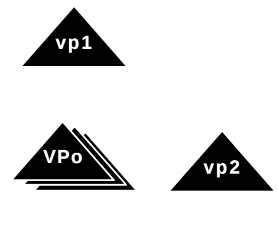
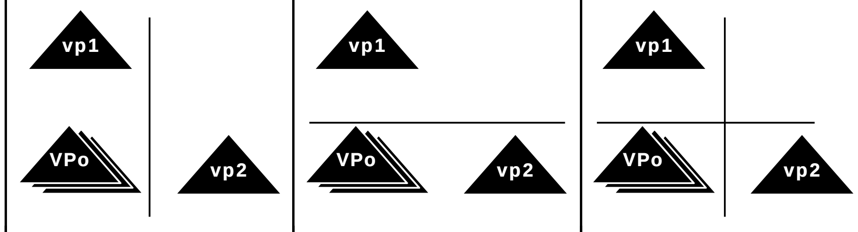
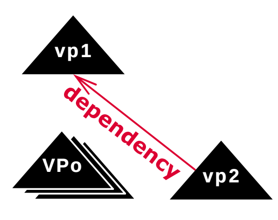
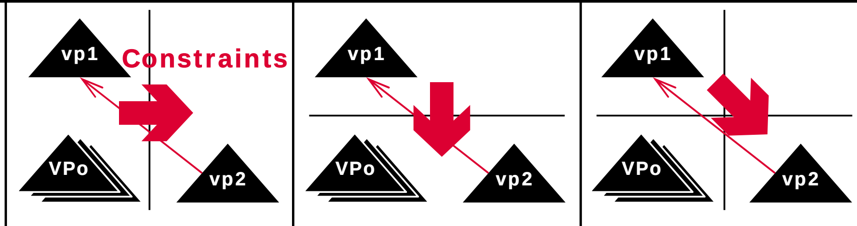
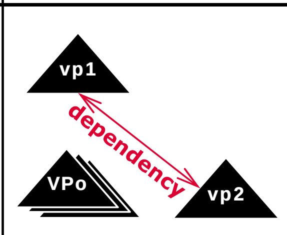
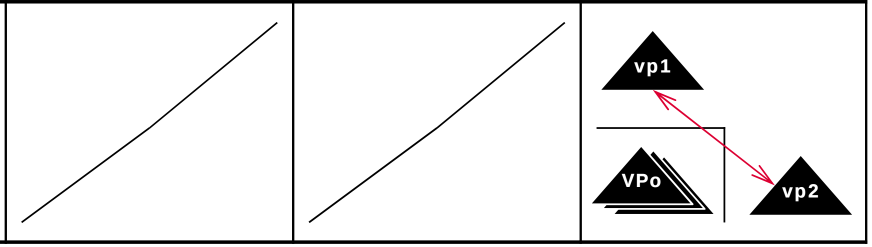
Analyze the structure of variability

↓

Analyze the dependency of the set of variation points

↓

Identify the order of divided development unit

Dependency	Variability structure	Constraints and dividing method
None		
vp2 depends on vp1		
vp1 and vp2 are inter-dependent		

Realize incremental development with less regression testing cost

7.

Application and Effectiveness

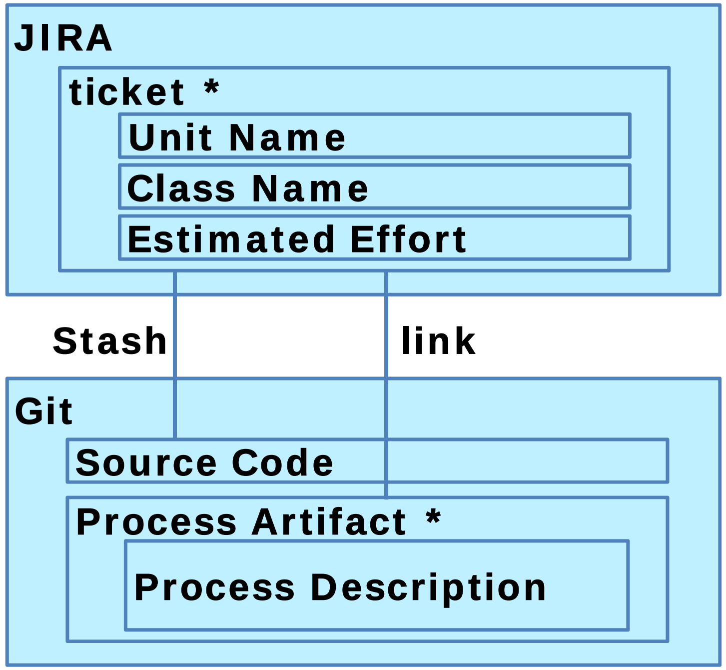
DENSO
Crafting the Core

Application

The presenter as the leader of the development team

Development duration	10 months
A unit of sprint	2 weeks
Total number of sprints	22 sprints
Total number of projects	11
Size of unit (KLOC)	1 - 20

Development Period and Target Number of Projects

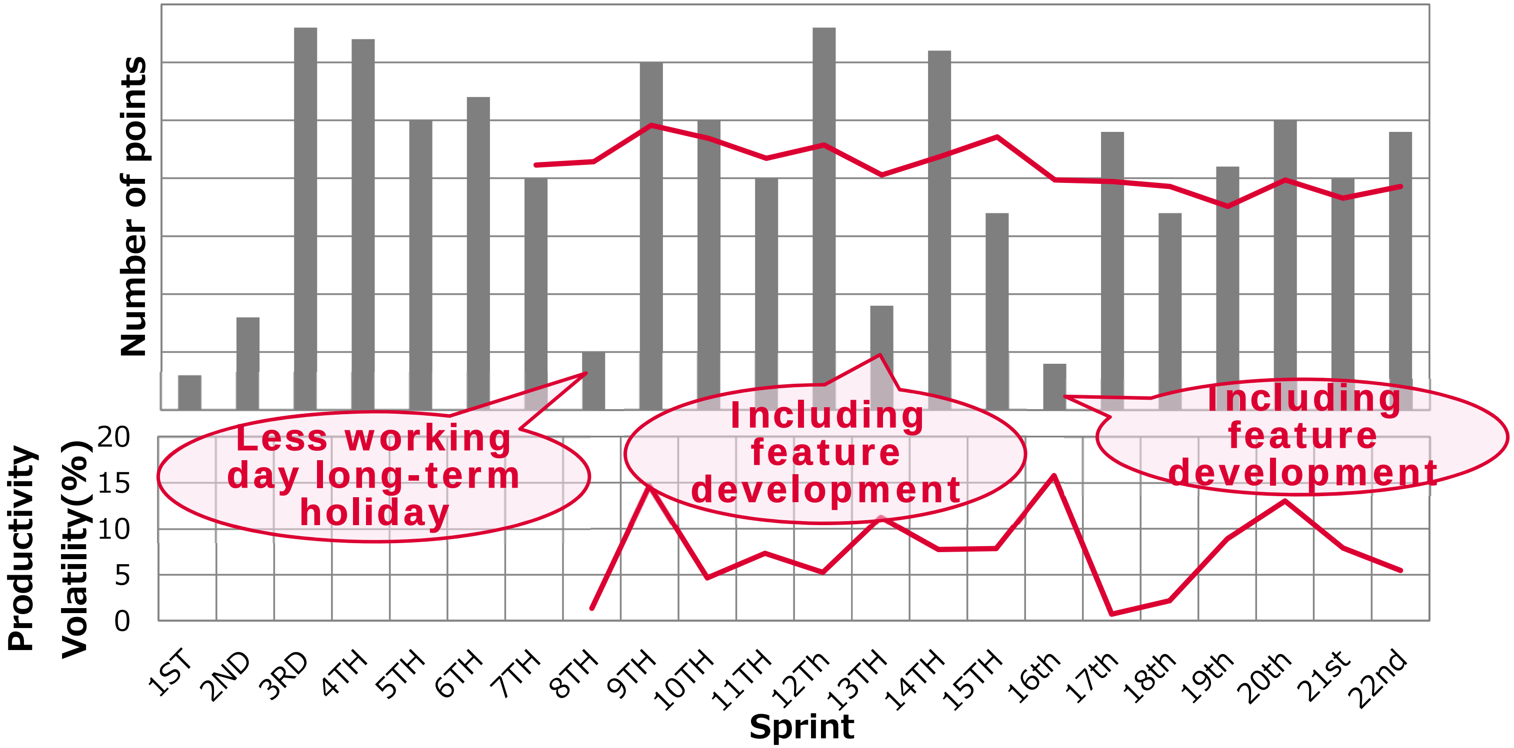


Development Environment with Process Assets

Statistics obtained from the actual projects

Stability of the Development

Measure the velocity (moving average of 7 sprints)
productivity is predictable = development is stable

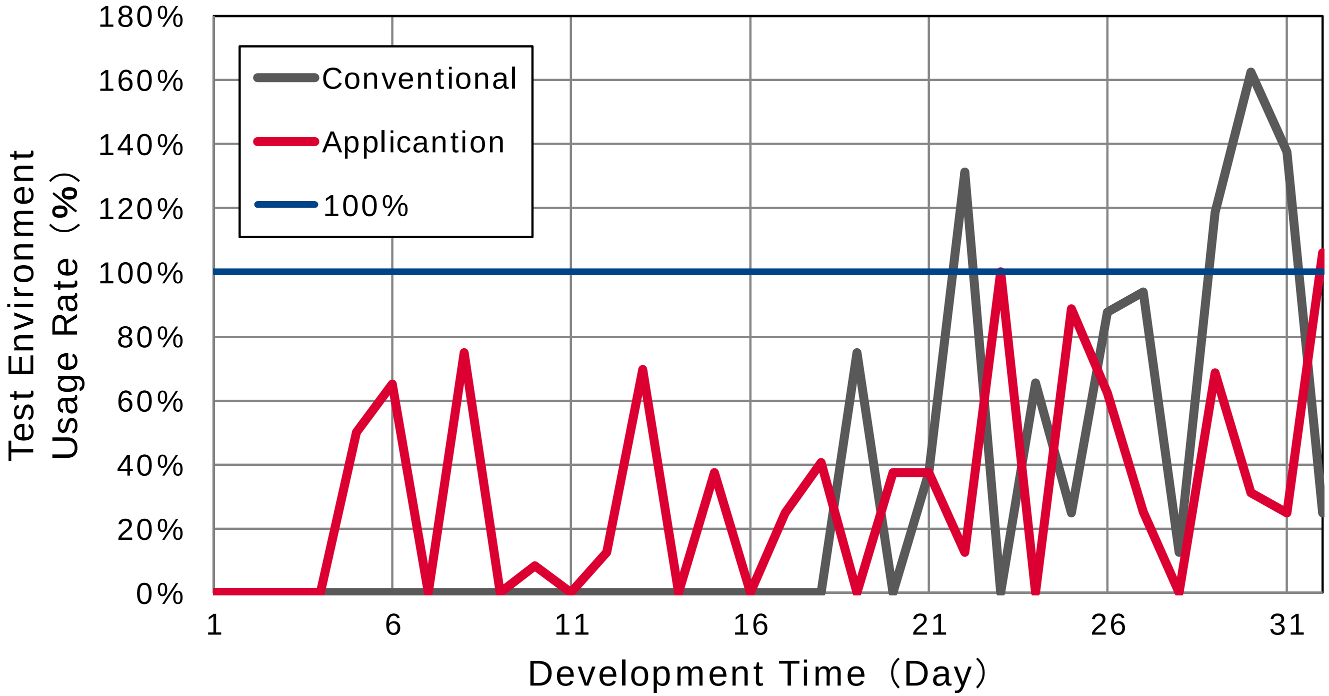


The development is highly stable if items have iterativeness

Leveling the Test Effort and Usage of Test Environment

Comparison of test environment usage rate with similar scale development

	Conventional	Application
Test days (Day)	12	20
Test hour (H)	77.75	78.32
Average (%)	80.99	48.95
Max (%)	162.50	106.25



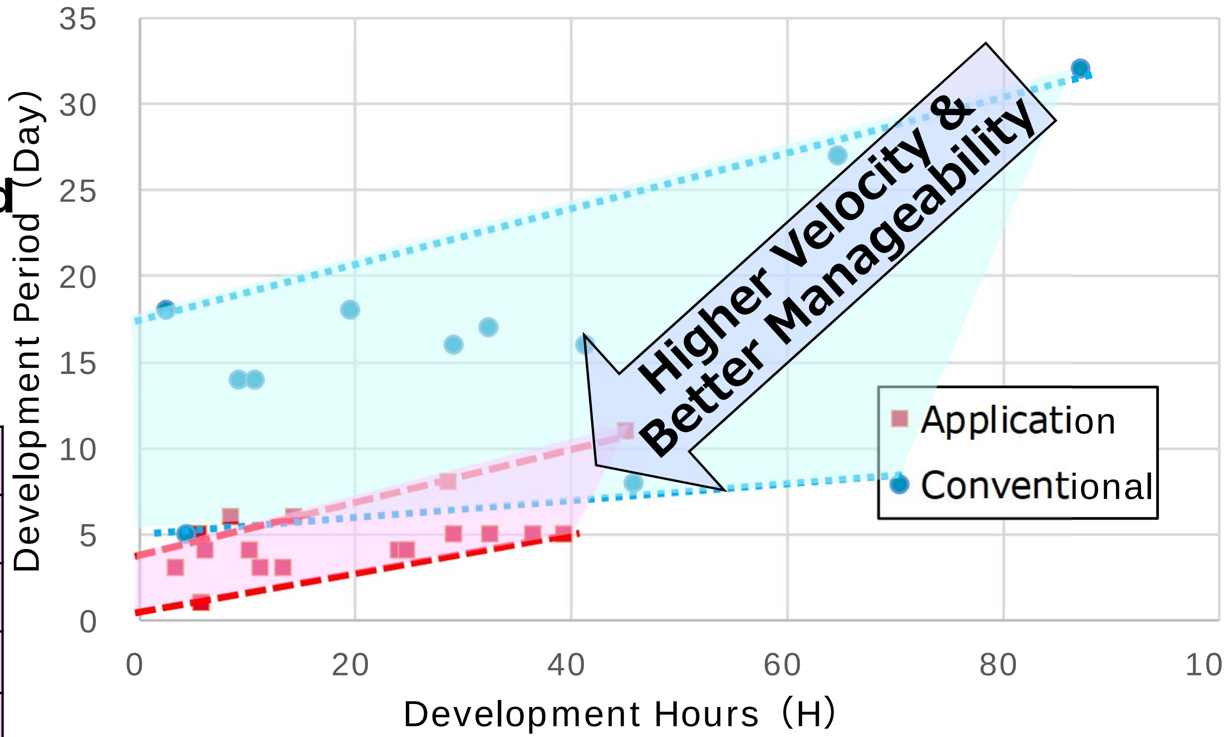
Lowered usage rate and leveling the peak load

Higher Velocity and Better Manageability of Value Stream

- Distribution of development period vs development effort for each development items
- Lower development time/period indicates higher velocity
- Lower SD indicates better manageability

		Conventional	Application
Item Count		11	17
Dev. Time (H)	Average	31.53	19.99
	SD	25.49	13.02
Dev. Period (Day)	Average	16.82	4.82
	SD	7.21	2.15

SD: Standard Deviation



Reduced variations and improved velocity of value stream

8.

Discussion and Future Works

DENSO
Crafting the Core

Discussion and Future Works

Q1. Has the portfolio management been strengthened?

A1. Yes. Stable productivity was obtained, the development scale was able to be grasped, and it become possible to keep updating the executable plan.

Q2. Is this approach the best?

A2. No, but Better. Automatic testing and a more ideal configuration system can realize simpler development.

Q3. Do not apply agile development for domain engineering?

A3. Domain engineering is easier to apply. However, it is necessary to care the architecture because the architecture is easy to erosion.

Q4. Is further improvement possible?

A4. Yes. In the future, we plan to develop an architecture accommodating concurrent development with domain engineering.

9.

Conclusions

DENSO
Crafting the Core

Conclusions

Goal
- **Improvement of manageability in concurrent product development on MPLE**

Solutions
- **Twofold Iterative Process Structure**
- **Process Design and Process Assets**
- **Dependency Analysis of Variability**

Benefits
- **The development is highly stable if items have iterativeness**
- **Lowered usage rate and leveling the peak load**
- **Reduced variations and improved velocity of value stream**

Future Works
- **Develop an architecture accommodating concurrent development with domain engineering**

About the Speakers

Mr. Kengo Hayashi is a architect and project manager of Advanced Safety Engineering Div., DENSO CORPORATION, Kariya, Japan.
He has engaged in the development of car navigation software systems and advanced sensing software system.
He is pursuing the doctoral program at Aoyama Laboratory, in the graduate school of software engineering, Nanzan University.
His research interests include software management, software product line engineering, and agile development.

Dr. Mikio Aoyama is a professor of Dep. of Software Engineering, Nanzan University, Nagoya, Japan since 2001.
His research interests include requirements engineering, software architecture, and machine learning for the applications in cloud/edge computing and automotive systems. Prior to joining academia, Dr. Aoyama has 15 years of experience in the development of large-scale real-time distributed systems at Fujitsu Limited.
His paper “Agile Software Process and Its Experience” presented at ICSE ’98 is one of the earliest work on agile.

DENSO

Crafting the Core