

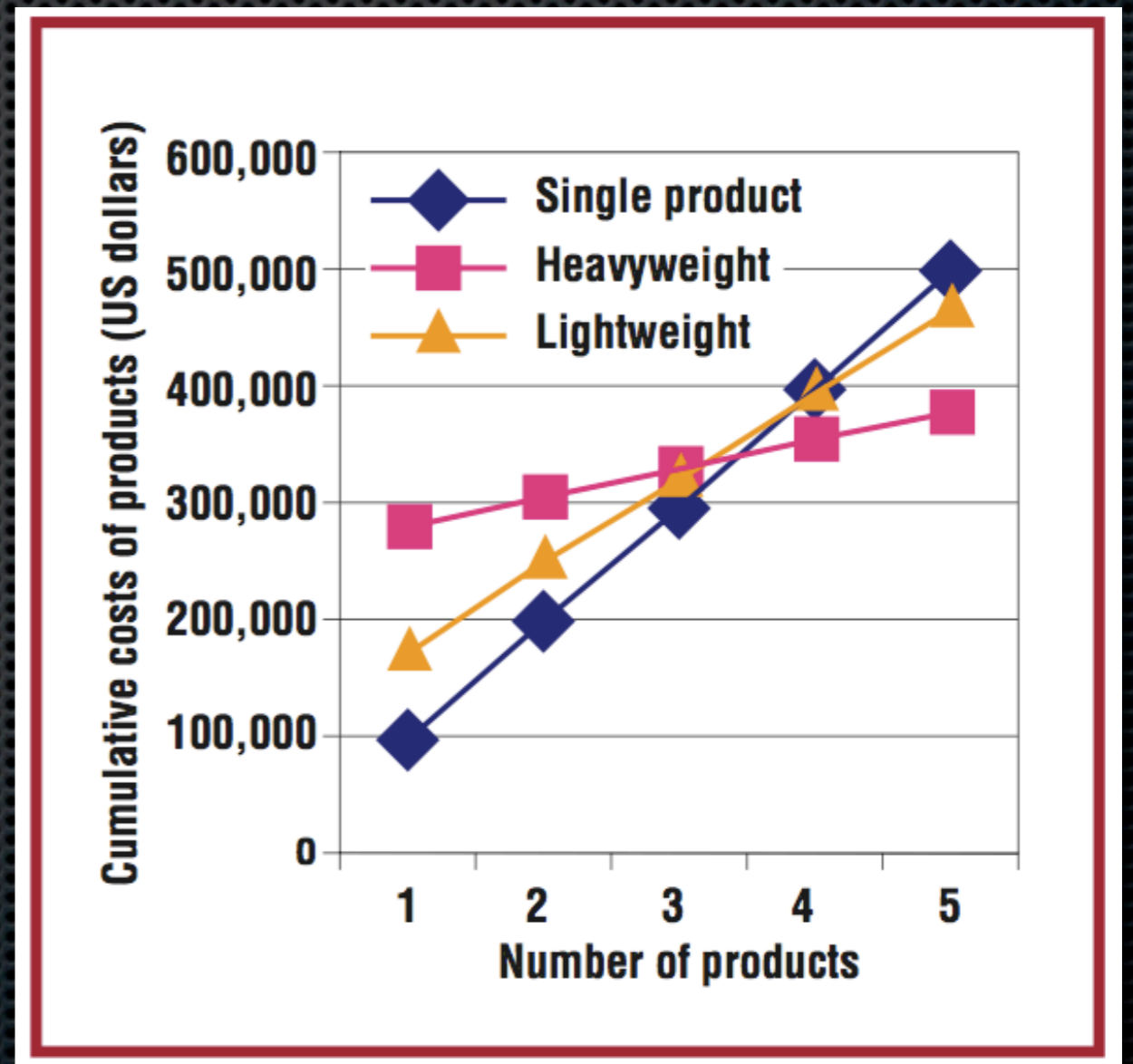
# Software Product Line Engineering

L8: Transitioning to SPL

Robert Feldt - [robert.feldt@gmail.com](mailto:robert.feldt@gmail.com)

# Transitioning/Adopting SPLs

- ✦ If we decide to adopt SPLs and transition to SPLE, HOW should we make the transition?
  - ✦ If? - PLPA, BAPO, FEF etc
  - ✦ How? - Transition strategies



# Incremental Adoption: Engenio case

- ✦ Engenio
  - ✦ High-performance storage servers
  - ✦ Customers: IBM, SGI, Cray, StorageTek, Teradata
  - ✦ Customers utilize E. core competence + wants unique features
  - ✦ Controller Firmware Dev team
    - ✦ Firmware for 82 products
    - ✦ ~1 Million LOC per product
    - ✦ 80% of code is common between products

# Incremental Adoption: Engenio case

## ✦ Challenges:

- ✦ Contractually dictated production schedules
- ✦ Business demand outpaced maintenance ability
- ✦ From sequential releases to intertwined/overlapping release cycles
- ✦ Product diversification: low-end hardware platform
- ✦ Variability through CM: 34% src files had 3-16 branches
- ✦ SPL adoption barrier: 2.5 products eq. upfront investm.
- ✦ => 900-1350 personmonths, 100 persons

# Incremental Adoption: Engenio case

- ✦ Solution: Incremental investments
  - ✦ 4 personmonth upfront investment => cumulative returns outpaced cumulative investments
  - ✦ Focus on current bottlenecks/inefficiencies
    - ✦ Excessive File Branching due to Multiple Product-focus was root causes
  - ✦ Upfront investment
    - ✦ Pilot study using SPL support tool, here BigLever Software Gears
    - ✦ 2 existing prods. remodeled => convinced mngmnt.
  - ✦ Small incremental SPL investments, no schedule disruptions

# SPL Support Tool

- ✦ Feature Modeling Language
  - ✦ model optional and varying features between products
- ✦ Product Feature Profile
  - ✦ instantiates feature model for each product
- ✦ Configurable SW Assets via Variation Points
  - ✦ language for programming variation points
  - ✦ v.p.'s configures themselves based on feature profile
- ✦ Configurator
  - ✦ compiler from (feature profile, assets) -> product
- ✦ Similar to ContinuousIntegration++ tool (à la agile)

# Incremental Adoption: Engenio case

4-stage Transition:

Infrastructure +  
core assets

Setup SPL infrastructure

Extract core assets from branches

2 products in 3300 files -> 3103 files + 51 v.p. files

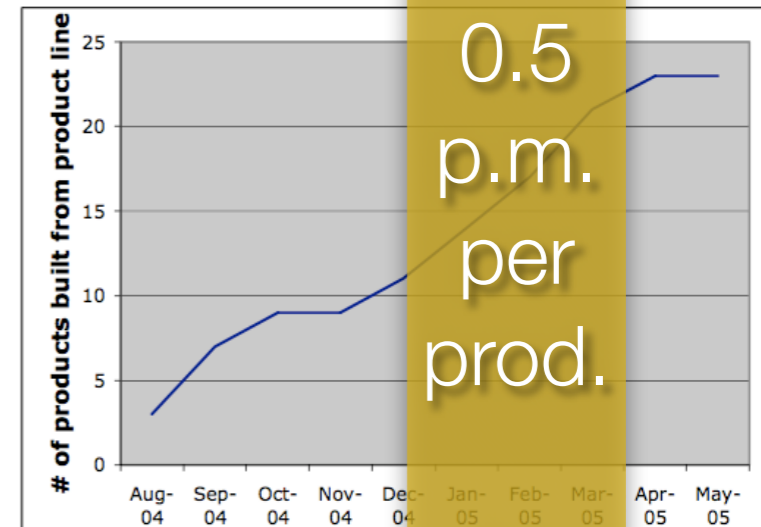
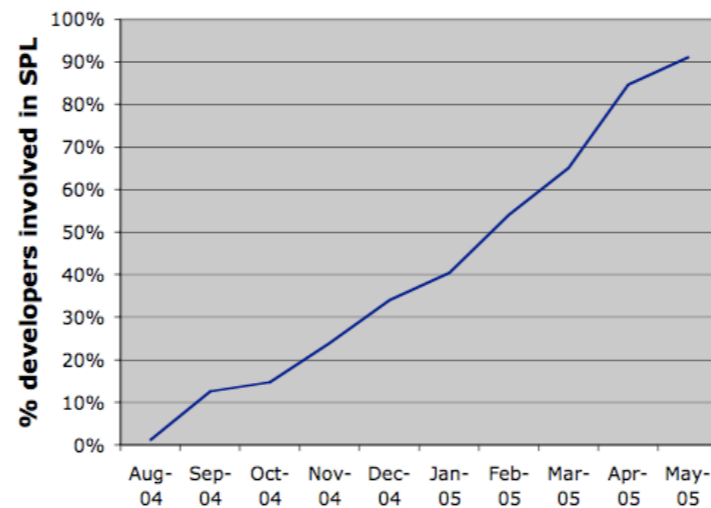
4

p.m.

Team  
organization

Transition teams from branching

Dev. Processes



0.5

p.m.

per  
prod.

21

prod.

Validation + Q.A.

Refactor core to optimize  
commonality and variation points

# Incremental Adoption: Engenio case

4-stage Transition:

Infrastructure +  
core assets



Team  
organization



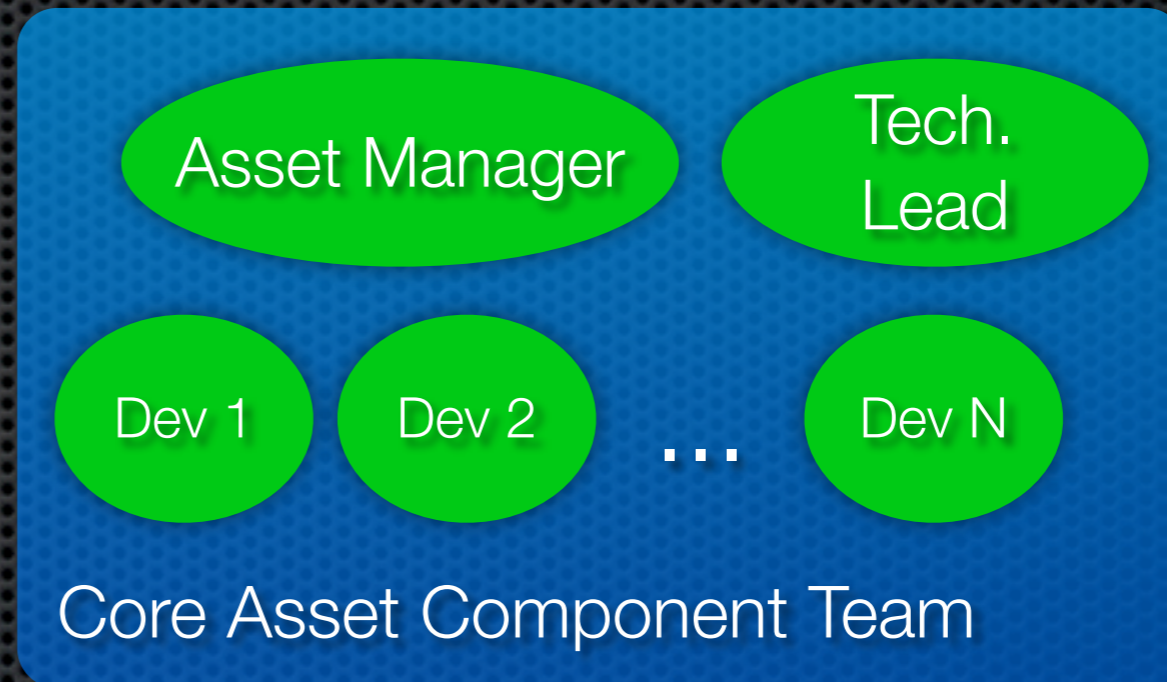
Dev. Processes



Validation + Q.A.

From product teams to core asset component teams

Assets grouped by service layers in architecture



Planning to define teams

Educating team members



# Incremental Adoption: Engenio case

4-stage Transition:

Infrastructure +  
core assets



Team  
organization



Dev. Processes



Validation + Q.A.

From Release centric to SW Product Family centric

Assemble Process Task Force

Add mapping step from feature reqs to asset reqs

Same time: Better respond to changing customer reqs

# Incremental Adoption: Engenio case

4-stage Transition:

Infrastructure +  
core assets



Team  
organization



Dev. Processes



Validation + Q.A.

Iterative Feature req validation

Shift responsibility of certification groups

# Incremental Adoption: Engenio case

- ✦ Investments:

- ✦ 4 personmonths upfront
- ✦ 12 personmonths total

- ✦ Outcomes:

- ✦ 23 products of 1MLOC each, and 135 developers shifted to SPL
- ✦ Increased quality and productivity
- ✦ After first 3 transition stages, they expanded from 23 to 52 products in 5 months
- ✦ By incrementally showing benefit, easier to convince people to actually change, harder for detractors

# Four Transition Strategies

- ❖ Big Bang introduction
  - Overall cost lower
  - Plan guides work
  - Core assets earlier
  - Harder to undo
- ❖ SPLE for new, key products “at once”
  - Higher costs/time
  - Stops other dev
- ❖ Incremental introduction
  - Change/stop unless good
  - Limited costs/time
  - Start small then expand in increments
    - Current dev continues
    - Rework/change
  - Expand: Organisational scope || Investments
    - More time
- ❖ Tactical approach
  - Focus on urgent needs
  - Start w. small team
  - Wrong direction
  - Partial adoption, driven by technical problems
    - Low start cost
    - Lack support
- ❖ Pilot project strategy
  - Current dev continues
  - Limited costs/time
  - More time
  - Develop new product partly via SPL
    - Change/stop unless good
    - Rework
  - First SPL product || Extension of related, existing prods ||  
Toy product || Prototyping

# Successful SPL Adoption

## ✦ Decide

- ✦ 1. Define Business Strategy and Vision
- ✦ 2. Learn about SPLE
- ✦ 3. Perform a risk analysis in company context

## ✦ Prepare

- ✦ 4. Identify stakeholders & Gain support for new ways of working
- ✦ 5. Set concrete goals for the transition & Create stakeholder business cases
- ✦ 6. Scope the PL to determine boundaries and content
- ✦ 7. Evaluate orgs status and ability to adopt new ways
- ✦ 8. Plan the transition, create adoption plan

## ✦ Transition

- ✦ 9. Launch and institutionalize

# Business Strategy & Vision (Decide)

## ✦ Strategy

- ✦ How is the world changing? How will it affect the company?
- ✦ Customer needs that we cannot support? New markets or segments? etc.

## ✦ Vision statement

- ✦ Example, DNV (Norwegian Marine Securities company), long-term vision for vessel information services market:

- ✦ *“To establish a common information repository containing or referring all information accumulated for an object, e.g. a vessel, throughout its life-cycle. This should enable the transfer of information on the object between all involved actors, including feedback of experiences accumulated during the object’s life-cycle, for efficient delivery of high quality services and for continuous learning and improvement.”*

WHAT!

NOT How!

# Risk analysis (Decide)

- ✦ Many potential risks to consider! Some:
  - ✦ Number of PL products lower than expected
  - ✦ Reuseable assets not accepted or used => waste
  - ✦ Customer satisfaction requires maximum control and flexibility
  - ✦ Domain and SPL thinking not accepted in org
  - ✦ Maintenance of existing products hinders SPL adoption
  - ✦ Strategies focus on “made to order” (bespoke/contract dev)
  - ✦ Too strong influence from existing customers
  - ✦ Cultural differences between departments hinders org restructuring DNV: Align
  - ✦ Formal (Big Bang) approach leads to too much org change Siemens: Only testing!

Telvent: Current risks drove SPL adoption

# Pulse-Eco Risk analysis (Decide)

- ✦ Consider benefits and risks for dimensions:
  - ✦ Domain Maturity - sufficient domain understanding?
  - ✦ Stability - requirements/market change speed?
  - ✦ Resource constraints - Money, Time, Experts/Knowledge
  - ✦ Organisational constraints - Cultural differences etc.
  - ✦ Market potential - internal (assets used?) + external (enough customers?)
  - ✦ Sufficient Commonality & Systematic Variability?
  - ✦ Coupling & Cohesion - higher coupling => harder to reuse
  - ✦ Existing assets?



# Gaining support (Prepare)

- ✦ Ongoing “sale” activity to stakeholders: dev, marketing, management
- ✦ Example, Bosch:
  - ✦ Workshops to inform, learn from feedback & gain trust
  - ✦ Enthusiastic middle managers enlisted as “ambassadors”
  - ✦ Vision statement behind the change for better communication:
    - ✦ *Software is built from a common architecture and a set of components using a product line approach, so that high quality individually tailored products can be built easily and predictably, using as few hardware resources as possible, thereby reducing overall development costs.*

# Gaining support (Prepare)

- ✦ Example, DNV:
  - ✦ Mock-ups to illustrate next gen products and their development
- ✦ Example, Philips Medical:
  - ✦ Community of architects from different departments
    - ✦ Discuss sw, platform, ideas, problems, roadmaps etc
    - ✦ Also decisions
    - ✦ Meetings, emails, intranet forums, teleconf continuously
  - ✦ Became “Champions” and permeated company
  - ✦ Over time there was less need for them => newcomers didn't see the SPL point

# Stakeholder Business Case (Prepare)

- ✦ Stakeholder: Product Manager
- ✦ Current state: Variation supported with file branching. Redundant work between similar customer products.
- ✦ Stakeholder goals:
  - ✦ Increase revenue, profit, market coverage, quality, time-to-market
- ✦ SPL Goal Achievement metrics:
  - ✦ Connects goals to SPL. How does SPL help reach goal? Metrics that compare to current situation/single-system dev? Connect to costs?
  - ✦ Decrease TTM <- Fewer file branches <- Feature models + V.P. (metric: Average branches per file, # of feature models, # V.P.)
- ✦ Deliverables, Resources, Workload

# Evaluating Organisation (Prepare)

- ✦ To better understand which actions to take
- ✦ PLPA, FEF/BAPO, CMMI
- ✦ SEI's SPL Practice Framework
  - ✦ 29 practice areas in 3 categories:
    - ✦ SE: Arch def, Arch eval, Component dev, Mining existing assets
    - ✦ Technical Mngmnt: Config Mngmnt, Measurements, Scoping
    - ✦ Organisational Mngmnt: Business case, Funding, ...
  - ✦ PLTP = PL Technical Probe
    - ✦ Strengths and weaknesses in SPLP Framework

# Launching SPLE (Transition)

- Example, market maker
  - Hired new dev that started SPL dev
  - Close integration with rest, Existing assets to use
  - Firm time deadline to focus
- Exampen, Phillips Consumer Electronics
  - 3 years to set up
  - Two lead products on SPL: high visibility, low risk
  - Then roll out to other products

# Launching SPL (Transition)

- ✦ Example, Bosch
  - ✦ Need for 2 SPLs + single-system dev
  - ✦ Stepwise transition:
    - ✦ small pilot projects in R&D collected experience
    - ✦ new business unit with people from all depts
    - ✦ champions in middle management

W. Hetrick, C. Krueger, J. Moore, “Incremental Return on Incremental Investment: Engenio’s Transition to Software Product Line Practice”, Dynamic Languages Symposium, Portland, Oregon, 2006, pp. 798-804.

J.D. McGregor, S. Jarrad, L.M. Northrop, and K. Pohl; “Initiating Software Product Lines”, IEEE Software, vol. 19, no. 4, July 2002, pp.24–27.

K. Schmid, “An assessment approach to analyzing benefits and risks of product lines”, In The Twenty-Fifth Annual International Computer Software and Applications Conference (Compsac’01), pages 525–530, 2001.