Abstract

Most products fit in a larger family of products. The members of such a product family share a lot of functionality and features. It is attractive to share implementations, designs et cetera between those members to increase the efficiency of the entire company.

In practice many difficulties pop up when product developments become coupled, due to the partial developments which are shared. This article discusses the advantages and disadvantages of a family approach based on shared developments and provides some methods to increase the chance on success.
Typical Examples of Generic Developments

- Platform
- Common components
- Standard design
- Framework
- Family architecture
- Generic aspects, functions, or features
- Reuse
- Products (in project environment)
### Claimed Advantages of Generic Developments

<table>
<thead>
<tr>
<th>Advantage</th>
<th>Benefit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reduced time to market</td>
<td>building on shared components</td>
</tr>
<tr>
<td>Reduced cost per function</td>
<td>build every function only once</td>
</tr>
<tr>
<td>Improved quality</td>
<td>maturing realization</td>
</tr>
<tr>
<td>Improved reliability</td>
<td></td>
</tr>
<tr>
<td>Improved predictability</td>
<td></td>
</tr>
<tr>
<td>Easier diversity management</td>
<td>modularity</td>
</tr>
<tr>
<td>Increases uniformity</td>
<td>less learning</td>
</tr>
<tr>
<td>Employees only have to understand one base system</td>
<td></td>
</tr>
<tr>
<td>Larger purchasing power</td>
<td>economy of scale</td>
</tr>
<tr>
<td>Means to consolidate knowledge</td>
<td></td>
</tr>
<tr>
<td>Increase added value</td>
<td>not reinventing existing functionality</td>
</tr>
<tr>
<td>Enables parallel developments of multiple products</td>
<td></td>
</tr>
<tr>
<td>“Free” feature propagation</td>
<td>product-to-product or project-to-project</td>
</tr>
</tbody>
</table>

Product Families and Generic Aspects

3 Gerrit Muller

version: 2.3
March 7, 2019
GDclaims
Experiences with reuse, from counterproductive to effective

bad
- longer time to market
- high investments
- lots of maintenance
- poor quality
- poor reliability
- diversity is opposed
- lot of know how required
- predictable too late
- dependability
- knowledge dilution
- lack of market focus
- interference
- but integration required

good
- reduced time to market
- reduced investment
- reduced (shared) maintenance cost
- improved quality
- improved reliability
- easier diversity management
- understanding of one base system
- improved predictability
- larger purchasing power
- means to consolidate knowledge
- increase added value
- enables parallel developments
- free feature propagation
Successful examples of reuse

- **Homogeneous domain**: cath lab, MRI, television, waferstepper
- **Hardware dominated**: car, airplane, shaver, television
- **Limited scope**: audio codec, compression library, streaming library
Limits of successful reuse

struggle with integration/convergence with other domains

TV: digital networks and media
cath lab: US imaging, MRI

poor/slow response on paradigm shifts

TV: LCD screens
cath lab: image based acquisition control

software maintenance, configurations, integration, release

MRI: integration and test
wafersteppers: number of configurations

how to innovate?
Drivers for Generic Developments

Customer value
- application adaptability
- availability variations
- new features originating from different products
- timely availability
- reliability

Internal benefits
- asset creation
- increase economy of scale

Extrovert driver

Introvert driver
- availability of accumulated feature set
- design for configurability
- shared architectural framework
- quality increase
- predictability
- availability integrated base product
- maturity
Granularity of generic developments shown in 2 dimensions

- **Delegated integration**
- **Shared integration**

- **System**
- **Platform**
- **Subsystem**
- **Module**
- **Component**

- **CV**
- **MIP**
- **EVM**

- **Generator**
- **Flat detector**
- **CCD**

- **Actual integration level**
- **Intended integration level**

- **EV**
- **R/F**
Modified Process Decomposition

- Customer-Oriented Process
  - Policy and Planning Process
  - Customer-Related Processes
  - Product Creation Process
  - Shared Assets Creation Process
  - People, Process, and Technology Management Process

Customer Roadmap
- Business Drivers
- Technology, Process, and People Roadmaps
- Product Roadmap
- Budgets

Product needs and feedback
- Customer needs
- Technical Product Doc.
- Product-related processes

Support
- Info
- Order
- Product
- $\$

Material
- Business
- Technical
- People

Product needs & feedback
- Budget, Plan
- Technical Product
- Product-related processes

Shared Assets Creation Process
- Generic assets
- Technology, Process, and People roadmaps

People, Process, and Technology Management Process
- People Technology Process
- Technology, Process, and People roadmaps
- Needs & Feedback
- Product roadmap

Product Families and Generic Aspects

version: 2.3
March 7, 2019
GDPprocessDecompositionFamily
Financial Viewpoint on Process Decomposition

Product Families and Generic Aspects

version: 2.3
March 7, 2019
GDprocessDecompositionFamilyByValue
Value and Feedback Flow

Policy and Planning Process

Customer-Oriented Process

Product Creation Process

Shared Assets Creation Process

People, Process, and Technology Management Process

Customer roadmap

Business drivers

Product roadmap

Product needs and feedback

Product creation process

Product needs and feedback

Material

Presales

Sales

Logistics

Production

Service

Support

Technical product documentation

Technical product roadmap

Product-related processes

People Technology Process

Budgets

Product roadmap

Business processes

Technology roadmap

Technology, Process, and People roadmaps

Budgets

Shared Assets Creation Process

People, Process, and Technology Management Process

Product Families and Generic Aspects

11 Gerrit Muller

version: 2.3

March 7, 2019
Modified Operational Organization PCP

Product Families and Generic Aspects

version: 2.3
March 7, 2019
GDoperationalOrganization
Propagation Delay Platform Feature to Market

Product Families and Generic Aspects

version: 2.3
March 7, 2019
GDpropagationDelay
Sources of Failure in Generic Developments

**Technical**
- Too generic
- Innovation stops (stable interfaces)
- Vulnerability

**Process/People/Organization**
- Forced cooperation
- Time platform feature to market
- Unrealistic expectations
- Distance platform developer to customer
- No marketing ownership
- Bureaucratic process (no flexibility)
- New employees, knowledge dilution
- Underestimation of platform support
- Overstretching of product scope
- Nonmanagement, organizational scope increase
- Underestimation of integration
- Component/platform determines business policy
- Subcritical investment
Models for Generic Development

Product Families and Generic Aspects

15  Gerrit Muller

version: 2.3  March 7, 2019  GDmodels