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

<table>
<thead>
<tr>
<th>Platform</th>
</tr>
</thead>
<tbody>
<tr>
<td>Common components</td>
</tr>
<tr>
<td>Standard design</td>
</tr>
<tr>
<td>Framework</td>
</tr>
<tr>
<td>Family architecture</td>
</tr>
<tr>
<td>Generic aspects, functions, or features</td>
</tr>
<tr>
<td>Reuse</td>
</tr>
<tr>
<td>Products (in project environment)</td>
</tr>
</tbody>
</table>
Claimed Advantages of Generic Developments

- Reduced time to market
- Reduced cost per function
- Improved quality
- Improved reliability
- Improved predictability
- Easier diversity management
- Increases uniformity
- Employees only have to understand one base system
- Larger purchasing power
- Means to consolidate knowledge
- Increase added value
- Enables parallel developments of multiple products
- “Free” feature propagation

- building on shared components
- build every function only once
- maturing realization
- modularity
- less learning
- economy of scale
- not reinventing existing functionality
- product-to-product or project-to-project
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

<table>
<thead>
<tr>
<th>Homogeneous Domain</th>
<th>Cath lab</th>
<th>MRI</th>
<th>Television</th>
<th>Waferstepper</th>
</tr>
</thead>
</table>

| 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

Product Families and Generic Aspects
version: 2.3
August 21, 2020
Granularity of generic developments shown in 2 dimensions

- Delegated integration
- Shared integration

System components and subsystems illustrated with specific examples:
- CV: Generator
- MIP: Flat detector
- EVM: CCD

Integration levels:
- Actual integration level
- Intended integration level
Modified Process Decomposition

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

Customer

Material

Needs & Feedback

Technical Product Doc.

Product-Related Processes

Support

Customer-Oriented Process

Product Creation Process

Shared Assets Creation Process

People, Process, and Technology Management Process

Customer

Budget

Product roadmap

Business Drivers

Generic assets

Technology Process

People Process

Needs & Feedback

Technical Product Doc.

Product-Related Processes

Support

Customer

Budget

Product roadmap

Business Drivers

Generic assets

Technology Process

People Process

Needs & Feedback

Technical Product Doc.

Product-Related Processes

Support

Customer

Budget

Product roadmap

Business Drivers

Generic assets

Technology Process

People Process

Needs & Feedback

Technical Product Doc.

Product-Related Processes

Support

Customer

Budget

Product roadmap

Business Drivers

Generic assets

Technology Process

People Process

Needs & Feedback

Technical Product Doc.

Product-Related Processes

Support
Financial Viewpoint on Process Decomposition

Product Families and Generic Aspects
10  Gerrit Muller

version: 2.3
August 21, 2020
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
Propagation Delay Platform Feature to Market

Product integration

product feature 1

product feature 2

Release

Product integration

test

feature 1

feature 2

Release

Product Families and Generic Aspects

version: 2.3
August 21, 2020
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

- **customer**
  - supplying business
  - policy and planning
  - customer oriented process (sales, service, production)
  - Product Creation Process
  - create generic components
  - people and technology management process
- **lead customer**
  - direct feedback too specific?
- **carrier product**
  - product feedback product specific?
- **platform**
  - feedback problem too generic
- **technology push**
  - no feedback

Product Families and Generic Aspects
version: 2.3
August 21, 2020
GDmodels