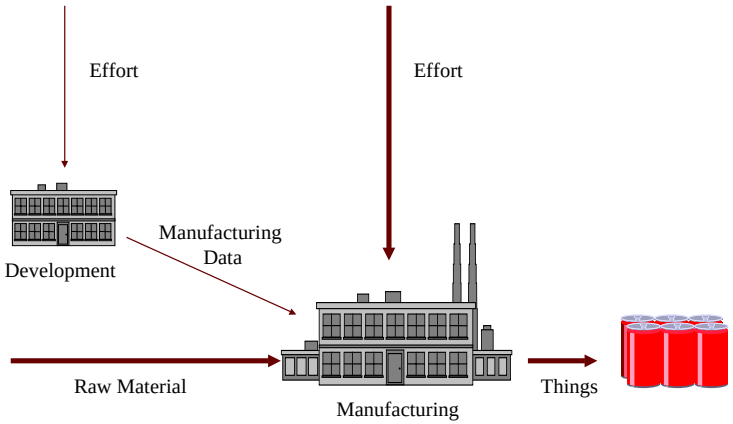
What is a Software Product?

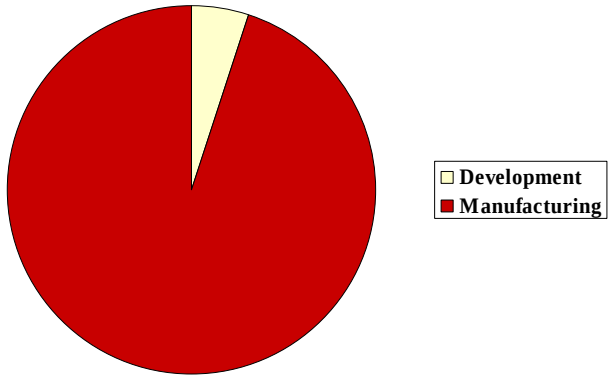
The Organisation of
Things, Files and People

Manufacturing: Production of Things

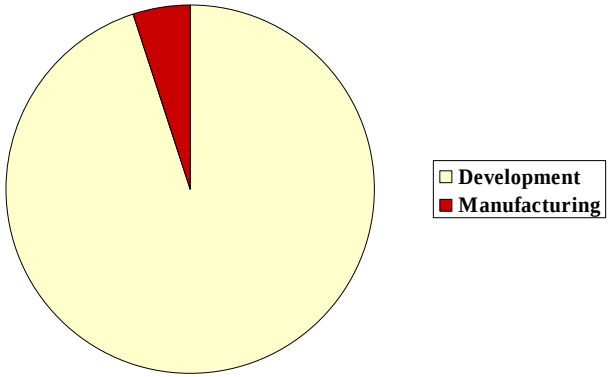
Traditional Production



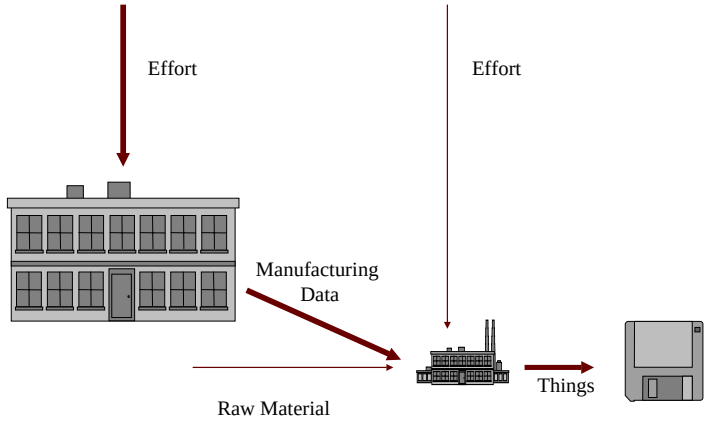
Traditional Production



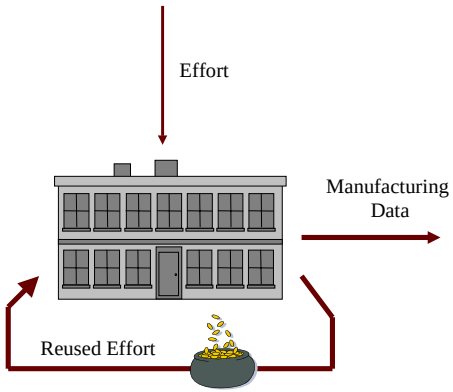
Production of Software



Production of Software



The Reusable Development Product



Types of Reuse

- Reuse within parts of a delivered product
- Reuse between releases of a delivered product
- Copy&Paste Reuse
- Reuse between different releases of different delivered products

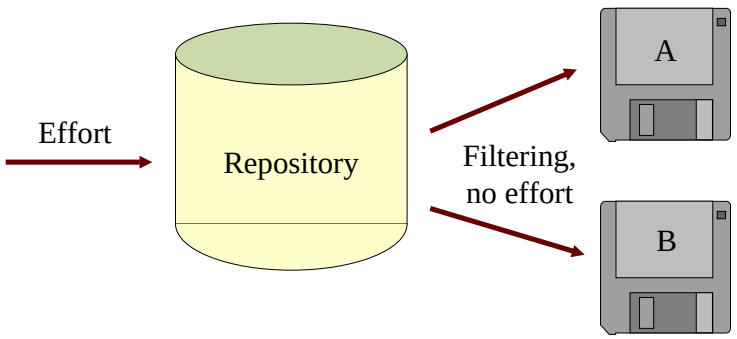
Non-traceable Reuse
(~Pfleeger:Compositional Reuse)

- Use need not to be documented
- Generic existing components
- Platform functionality
- Changes must be backward compatible
- Supplier and user may be in different organizations
- Quality is extremely important

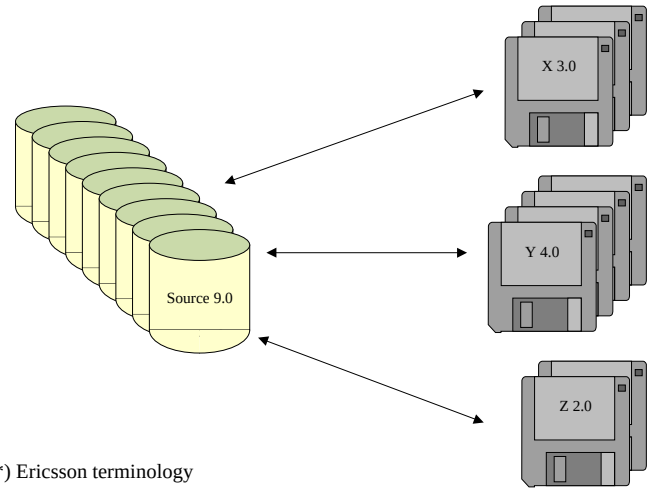
Traceable Reuse
(~Pfleeger:Generative Reuse)

- Explicit and documented use
- Components useful only in a certain context
- Application functionality
- All using functionality may be updated to allow non-backward compatible changes
- Supplier and user must be in the same organization
- Normal quality requirements

Production for Reuse

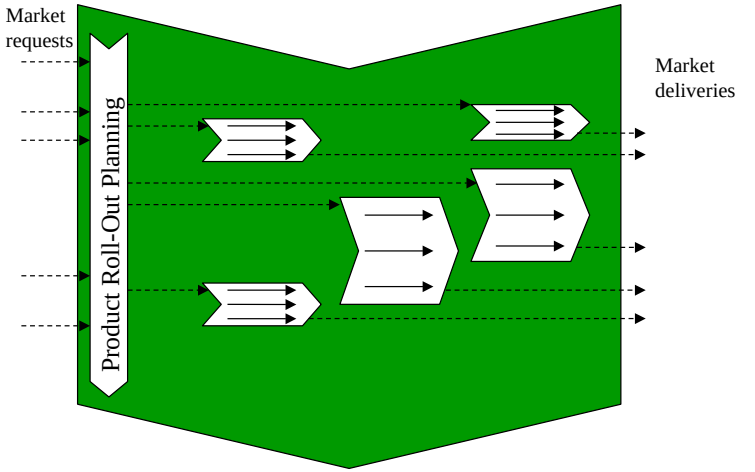


Source Systems* and Delivery Systems*

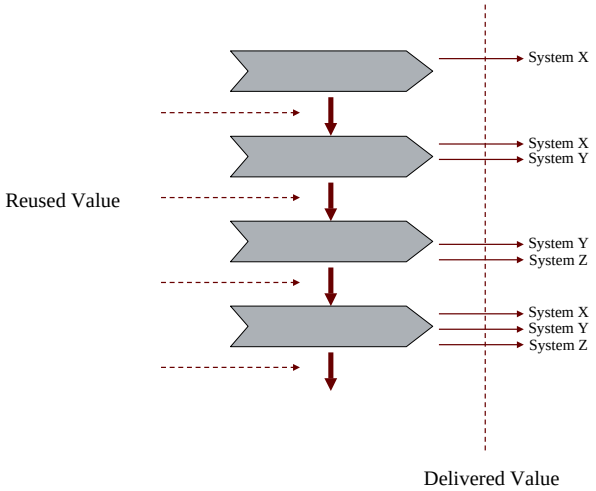


*) Ericsson terminology

Life Cycle



Evolving the Source System



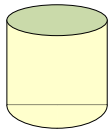
Value = Volume * Quality

- Delivered Quality
  - As perceived by the customer
- Reused Quality
  - As perceived by the future developer
- Maintained Quality
  - As perceived by maintenance responsible

Reuse:

The Art of Organizing
Produced Information

Storing Produced Information:



Data bases

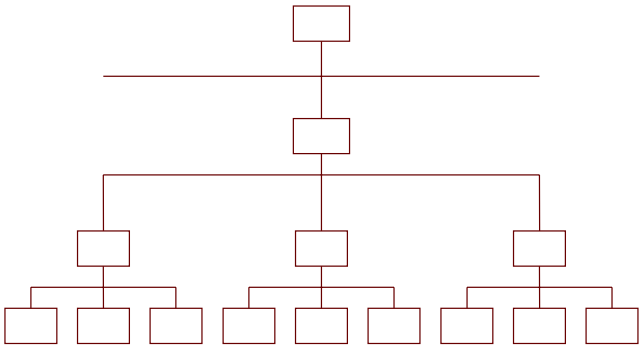
Files and documents

Peoples minds

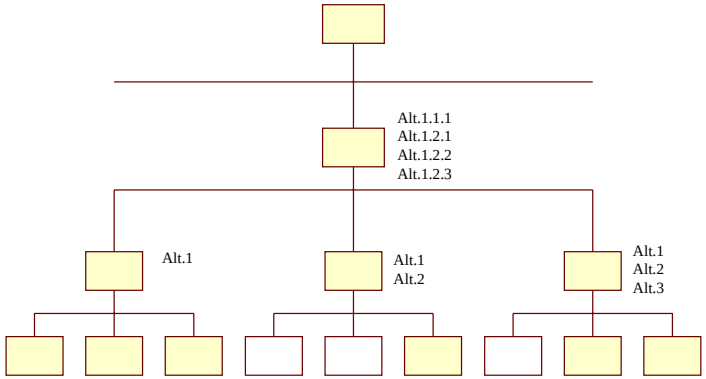
Organizing People:

Misplacing people means deterioration of Reused Value

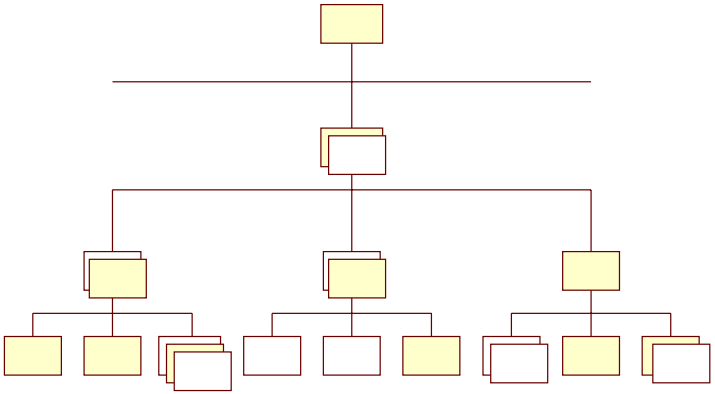
Simple Folder-Type Hierarchy



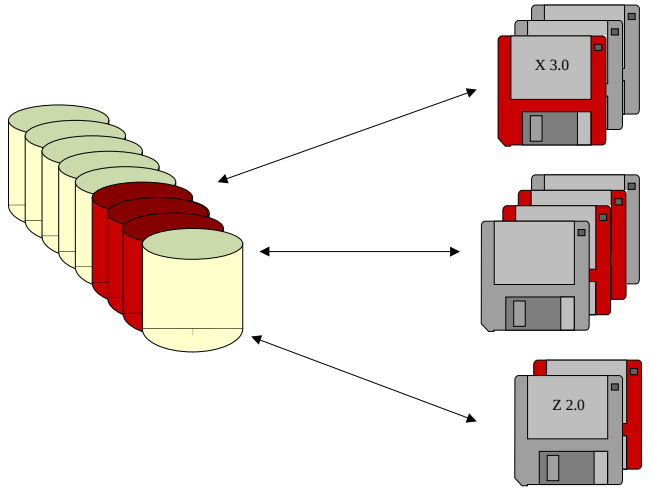
Manufacturing by Filtering



Versioning and Filtering



Product Tracking



Fault Tracking

- Which Source Systems contain the fault?
- Which Delivery System releases are produced from the faulty Source system?
- Which Delivery Systems include the faulty functionality

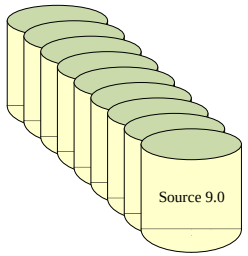
Feature Tracking

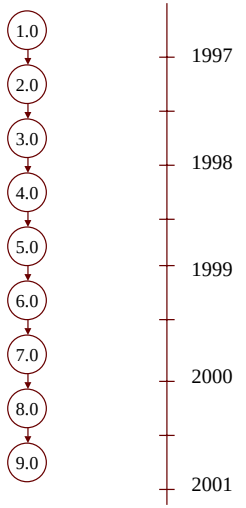
- In which release of a system will a certain feature be available?
- Which functionality has been delivered to a certain customer?
- Have all promised functionality been delivered?
- Which versions of different products are compatible?
- ...

Time To Market (TTM)

The time from when a product is wanted by the market until it is available for purchase.

Sequential Development



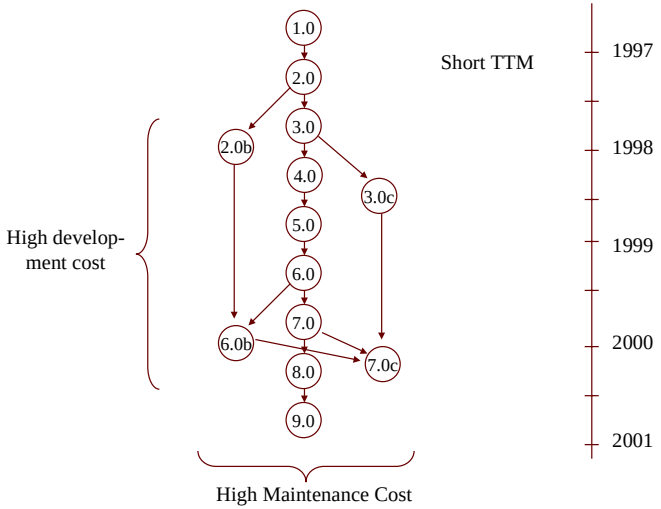


Low Development Cost

Low Maintenance cost

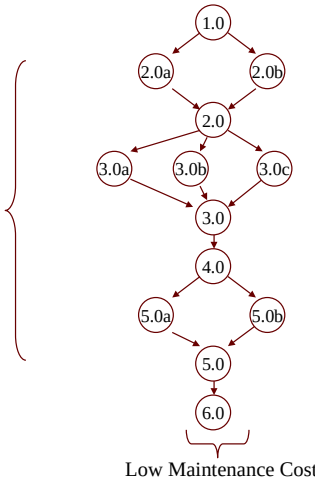
High TTM

Non Sequential development



Short TTM

Parallel Development



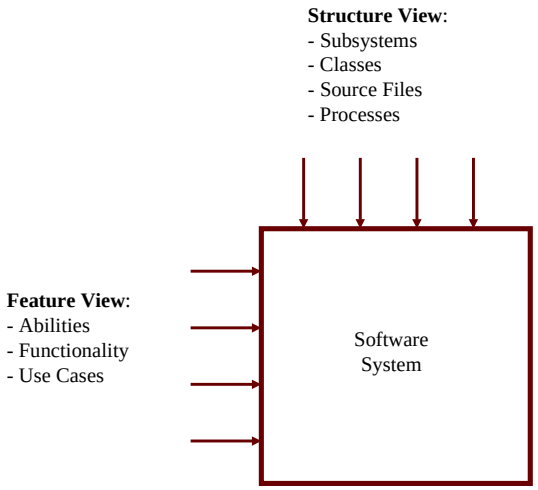
Short TTM

TTM Approaches: Examples

- Micro Project
- Branch/Merge
- Distributed Feature/Centralized Structure

Note: These are still unproven techniques

Orthogonal views



The Micro Project Approach

- Each development cycle is a very short, but complete project.
- Sequential development
- Few features developed in each cycle.
- Lead time < 3 months.
- Feature roll out planning for each cycle.
- Overlapping development => Product release every 4 weeks possible.
- People keep their roles between projects.
- Significant architectural impacts not possible.
- Works only for mature products.

The Branching/Merging Approach

- Parallel projects make uncoordinated impacts to different “branches” of a part of the product.
- Projects are feature oriented.
- After development is finished, branches are “merged” with support from tools.
- Merging is a separate design activity that may be more costly than feature development.
- Requires that features don’t have strong structural overlap.
- More suitable for administrative than real-time applications.
- Significant architectural impact not possible.

Distributed Feature/ Centralized Structure Approach

- Parallel feature oriented projects work independently in separate functional areas.
- Feature oriented projects assign all structural impacts to a common structure oriented project.
- Structure oriented project coordinates architecture and approves all impacts.
- People in structure oriented project keeps their roles between projects.