# SEPIOR
# Developing Reusable Software Components in CAM Environments

Dr. Wolfgang Daxwanger

SEKAS *GmbH*

Munich, GERMANY

daxwanger@sekas.de

# Opening Thought

The software industry is trying to do now, what the hardware industry has been doing for many years:

- Define standards for interoperability of components.
- Integrate components into packages at various levels of abstraction.

# Status Quo

- Components are already ubiquitous on the desktop (COM, DCOM, COM+, OLE, ActiveX, JavaBeans,...).
- In Computer Aided Manufacturing (CAM) environments components are used mainly on application level.
- The main part of software is **rewritten** for every automation project.
- Very low level of software reuse.
- Consequently SEKAS set up the PIE SEPIOR funded by the European Systems & Software Initiative ESSI.

# Objectives of SEPIOR

systematic introduction and application of

- OO-methodologies and Reusability

to

- Increase reusability
- Reduce development efforts
- Raise quality

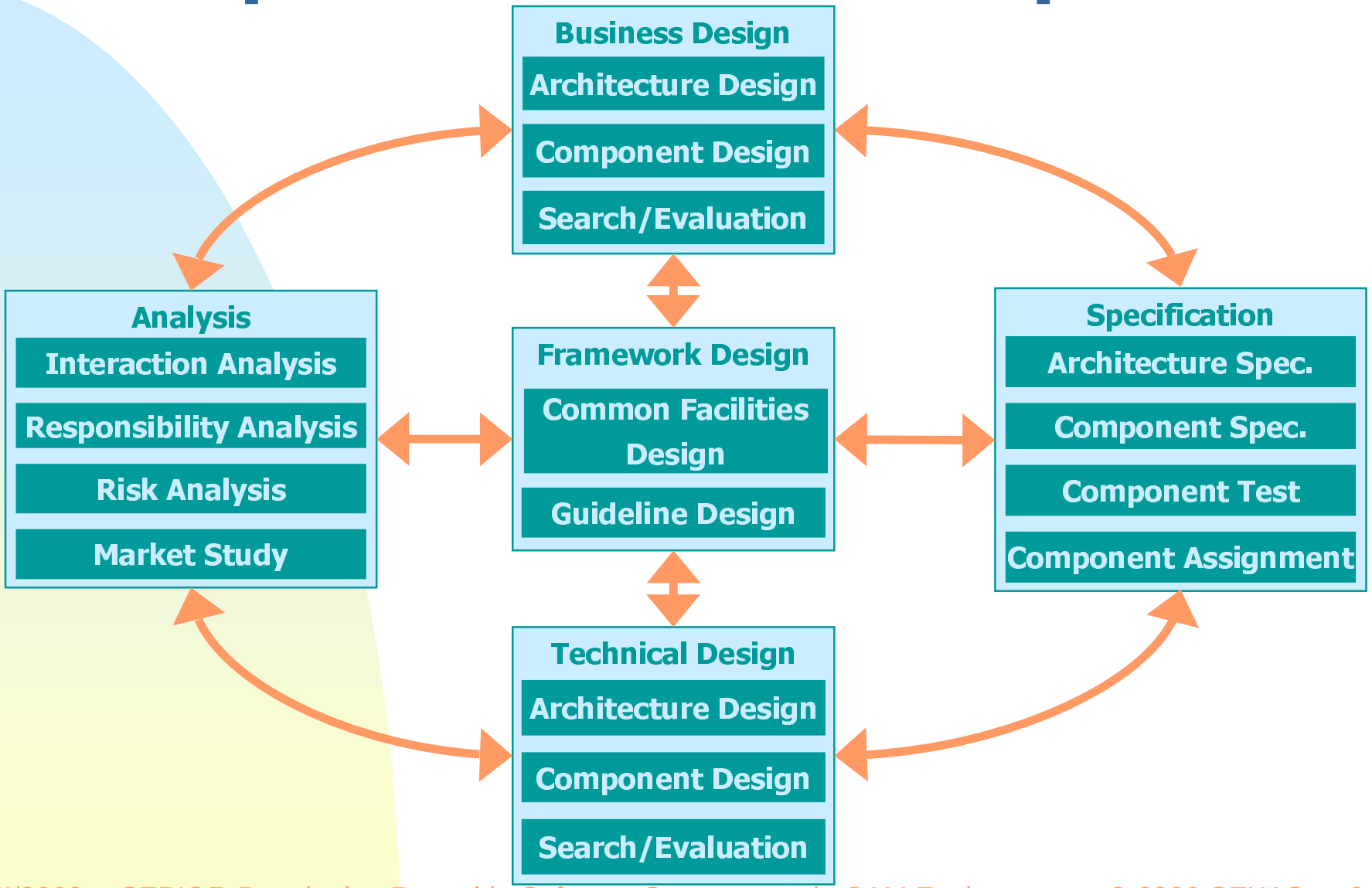
Reached through OO-based development of components

# What is a Component?

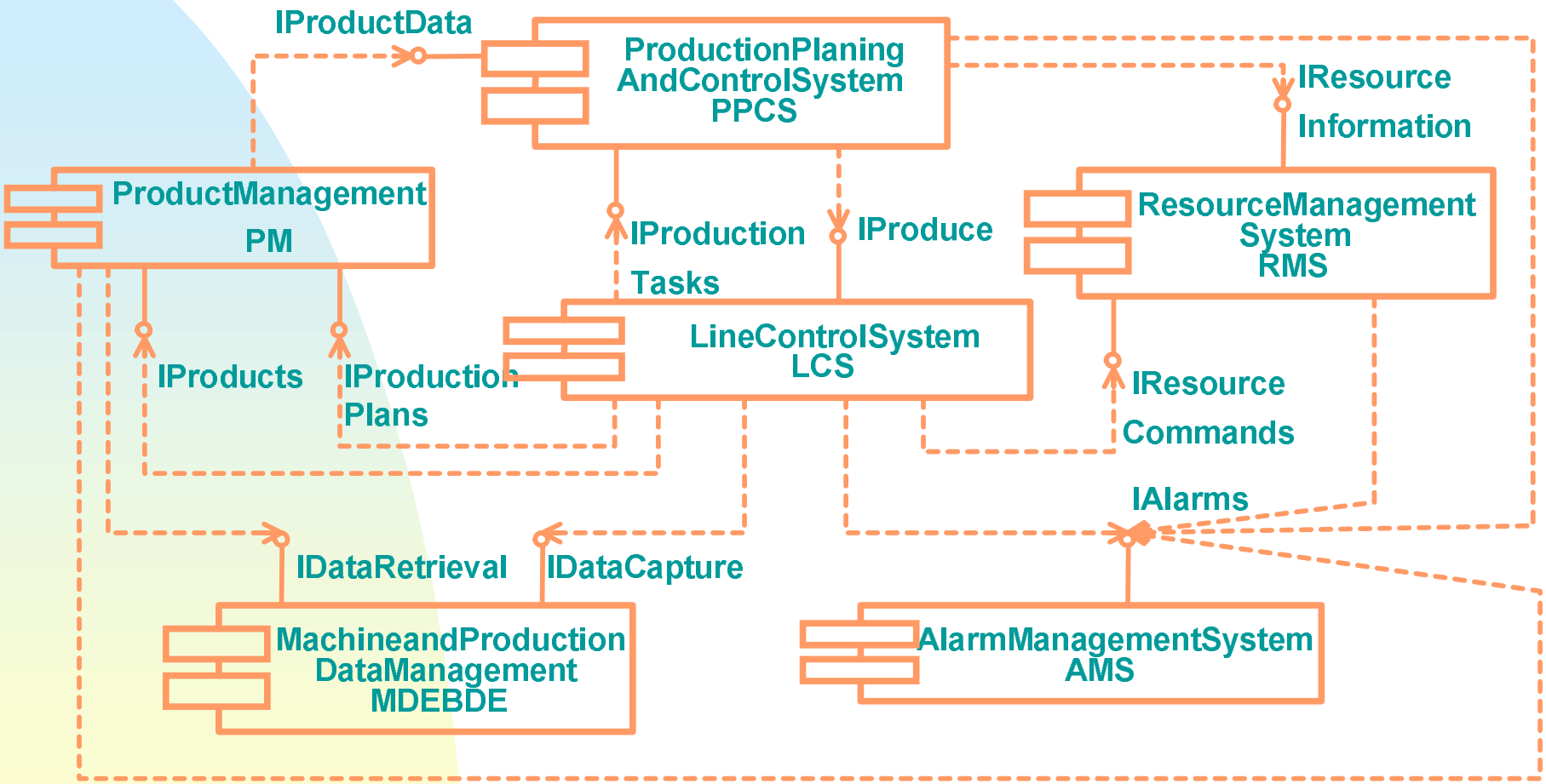
A component

- is a self-contained functional unit
- is reusable
- has precise and complete interfaces
- is distributed binary
- may be  configurable
- saves a lot of work for the developer and for the customer

# Process Model for Component-based Development



# Component Overview

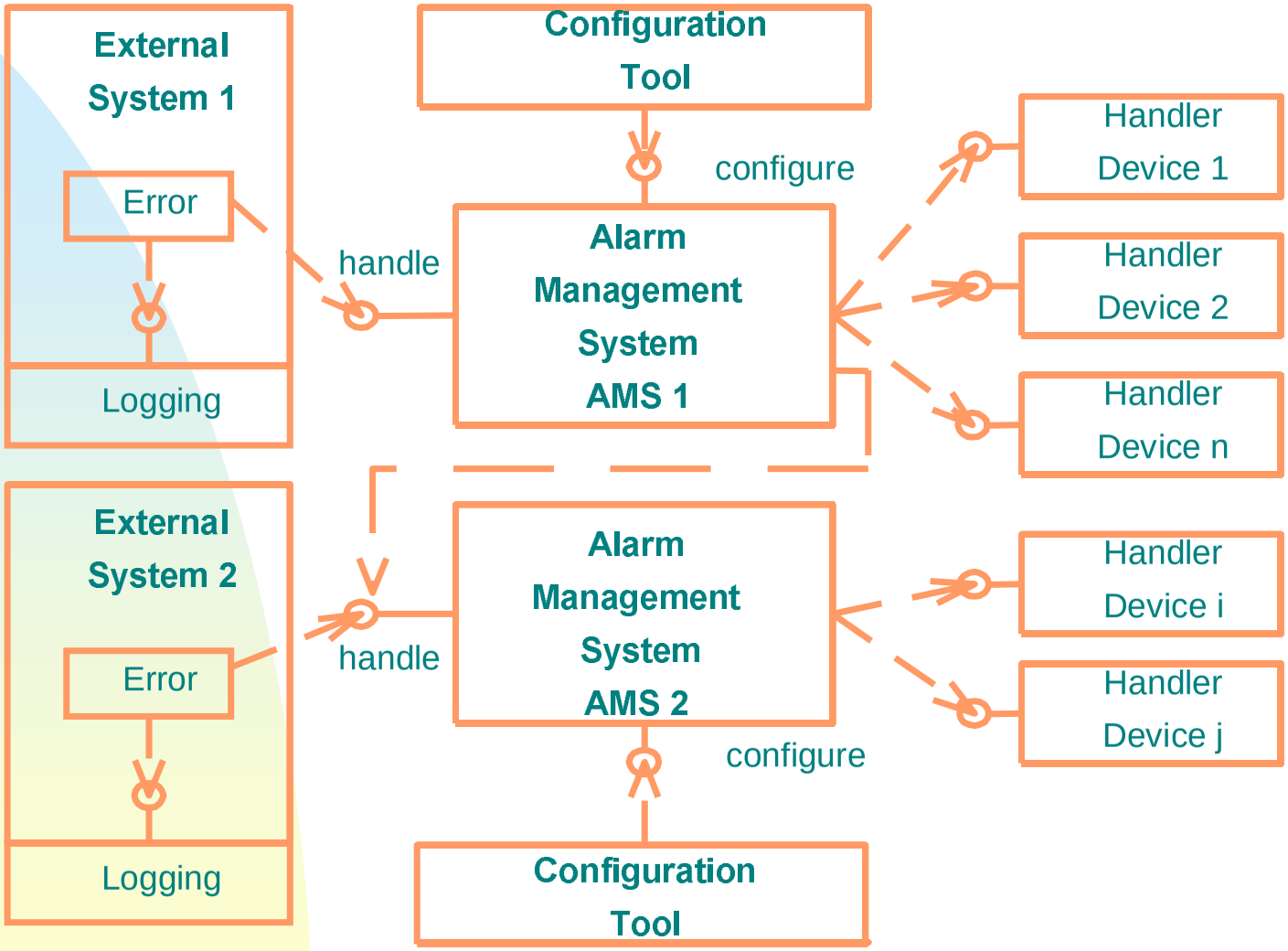


# Alarm Management System

Requirements

- No loss of error information
- Asynchronous error handling
- Freely configurable with escalation
- Platform independence
- Transparent API
- Context-sensitive error classification

# Alarm Management System



# Technical Design Aspects

- CORBA instead of COM/DCOM, because
  - implementation for all relevant systems exist.
  - CORBA is well suited for medium to large size components.
  - CAM is a traditional domain of CORBA.

- JAVA as development platform
  - shorter development time
  - superior CORBA integration compared to C++
  - distribution is an integral part of JAVA philosophy (WORA, serialization, ...)

# Impacts of Componentware

The introduction of Componentware has impacts on the following areas in the company

- Technical (CASE tool support, testing,..)
- Business (benefit ↔ effort)
- Skills (adoption of new methods and techniques)
- Organizational (process model)
- Cultural (incentives)

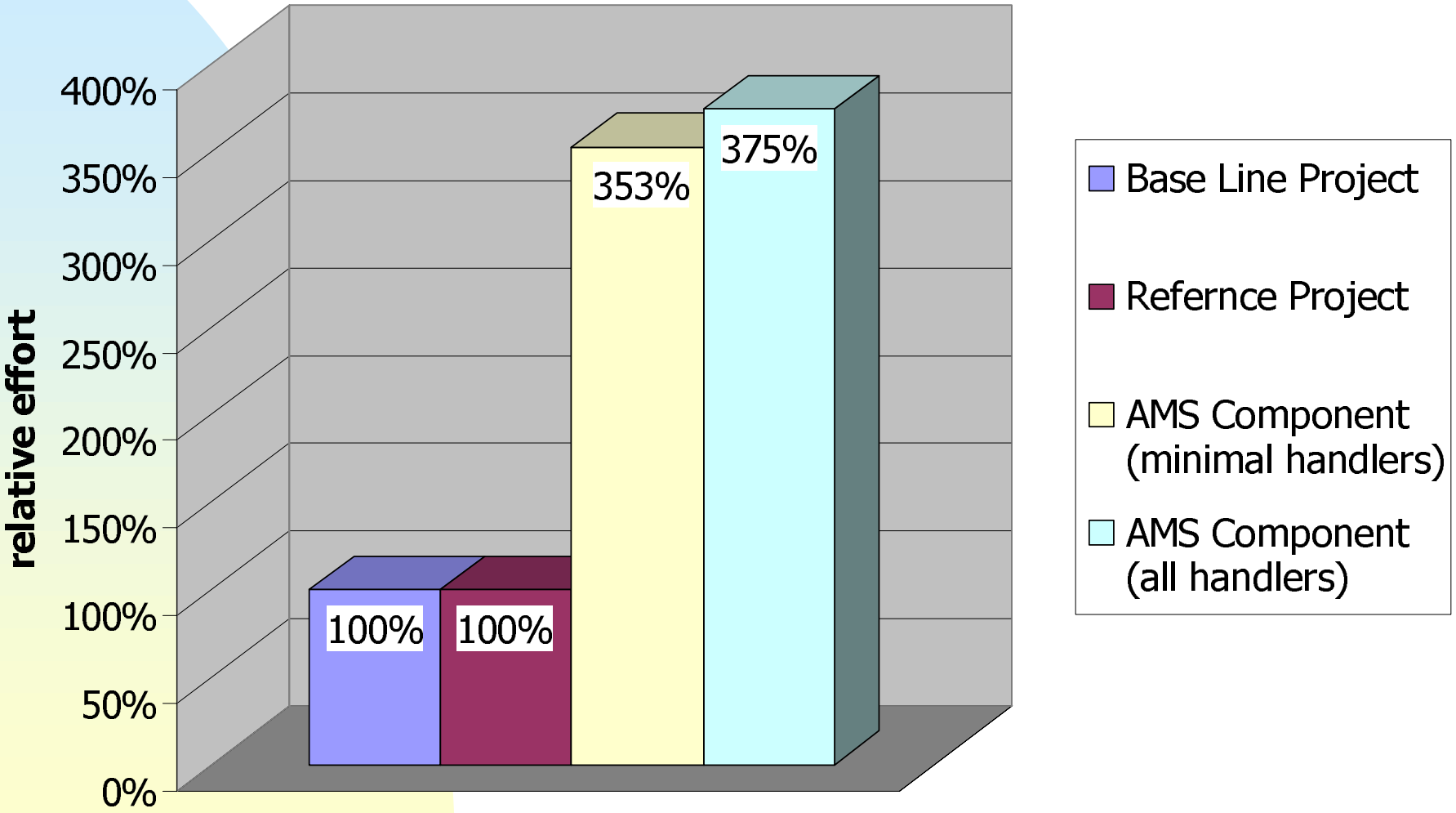
# Experiences: Human Factors

- Prolong training and coaching until adoption of methods is reached!
- Restrict size of discussion-teams to 3-5 members!
- Chose a CASE-Tool supporting all implementation phases from design to code generation with
  - UML
  - code synchronization
  - integrated documentation
- Dare to iterate!

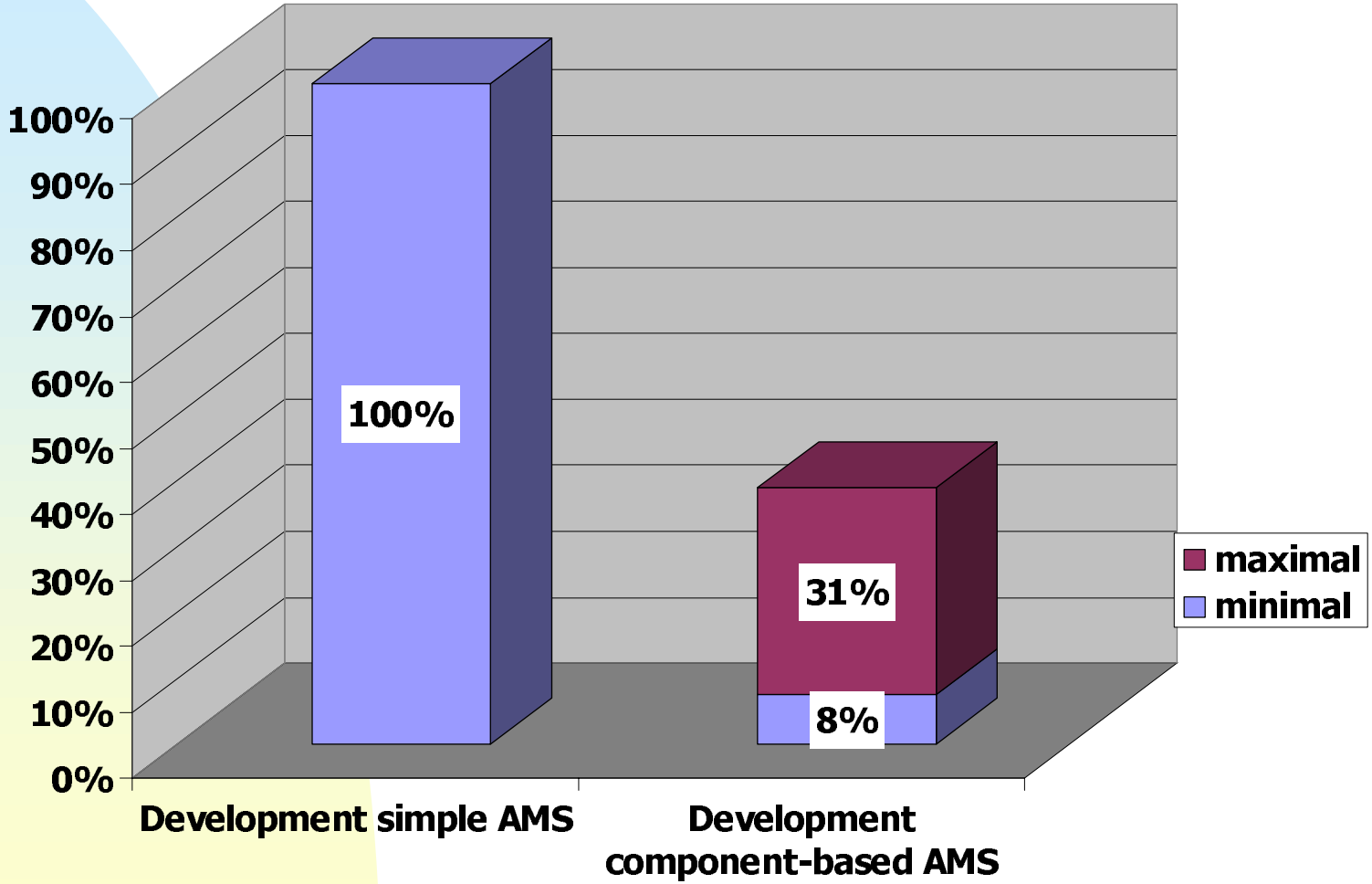
# Result Measurement

- Re-use impact is expressed by
  - saved effort in development
    - goal: reduce effort by 20%
    - measured by comparison of
      - effort for functionality in reference projects
      - development effort for component
      - preliminary effort for pilot project using component
  - saved effort in maintenance
    - goal: reduce effort by 50%
    - measured by extrapolation of maintenance within pilot project

# Result Measurement -Development Effort-



# Result Measurement -Development Effort vs. Reuse effort-



# Conclusions

- The introduction of components results in a
  - flexible construction of automation systems
  - reusing high quality software
  - with a high level of maturity and reliability.
- A solid framework makes it easy to
  - add
  - delete
  - substitute
  - extend

  } functionality in a CAM-system.