

SYSTEMCS



SYSTEMCS .NET Application Framework

Innovative Software Development for CAE-Tools

Agenda

- **Why do we need a new Methodology ?**
- The SYSTEMECS .NET Application Framework
- What are the extensions to OO and CBD ?
- Summary

Why do we need a new Methodology ?

- **Challenge of Efficiency**

- Do you drive every Day a new Way to your Office ?
- Do you buy a Car which was constructed especially for you ?
Every part a unique piece ?
- What are the Costs of such a Car, what is the Delivery Time and the Quality ?

- **Software Engineering**

- What Degree of Customer specific development do you have in your Software development ?
- Do you have a common Terminology, Methodology, Tools and a Strategy for Components ?

Why do we need a new Methodology ?

- **Vision**

- Is there an Advantage to have a common Frame for a Product Development in order not start from scratch every time ?

Why do we need a new Methodology ?

- **Challenge of Flexibility**

- Do you determine the Number of necessary Sockets in your House by counting just the currently used Electronic Devices ?

- **Software Engineering**

- Do you develop your Software system based on the currently known Features ?
- How do you prepare yourself for Feature changes or new Requirements ?

- **Vision**

- Is there an Advantage to have already today the Basis for an economic Product Life Cycle ?

Why do we need a new Methodology ?

- **Challenge of Compatibility**

- Have all Sockets in your House the same Format ?
Do they have all the same Voltage ?

- **Software Engineering**

- Can you Re-Use your Software Components in other Products ?

- **Vision**

- Is there an Advantage if a Methodology and a Framework would guarantee the Re-Use of Software ?

Why do we need a new Methodology ?

- **Challenge of Strategy**

- Why speaks a Car manufacturer very often of new Generations of Engines ?


- **Software Engineering**

- What are you doing in order to protect Investments in new Software Development ?

- **Vision**

- Is there an Advantage to have already today a Methodology which supports Product Lines ?

Why do we need a new Methodology ?



Common practice

Incompatible Tool Architectures

Tools are made entirely from scratch

Tools are monolithic or proprietary extensible

(Support of One-Off development, e.g. Development of ONE Tool)

The image shows a construction site with cranes and buildings under construction, illustrating the 'Common practice' of building tools from scratch.



Improvement



.NET Application Framework

SYSTEMCS

The image shows a modern glass skyscraper, illustrating the 'Improvement' through the .NET Application Framework and SYSTEMCS.

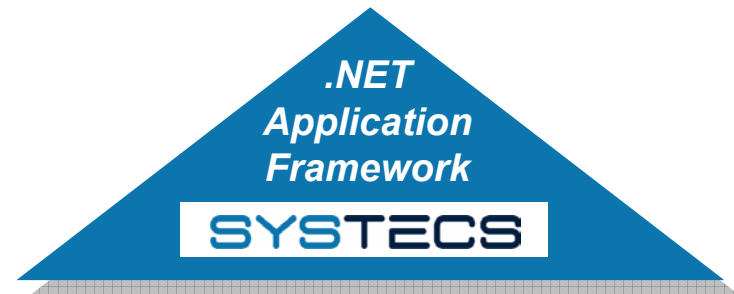
Common Practice

- **Software Architecture differ from CAE-Tool to CAE-Tool**
 - **Reason**
 - No common understanding of Software Architecture
 - Incompatible Tools and Processes
 - **Consequences**
 - Makes Re-Use impossible
 - Project instantiation is needless expensive
 - Delays Project progress

Improvement



- Introduction of a **Wizard** with integrated **Code-Generation** which supports a **standardized**
 - Terminology
 - Base Architecture
 - Methodology
 - Build
 - Tools
- This enables Re-Use of **Software Components**, fast Project setup and Project progress



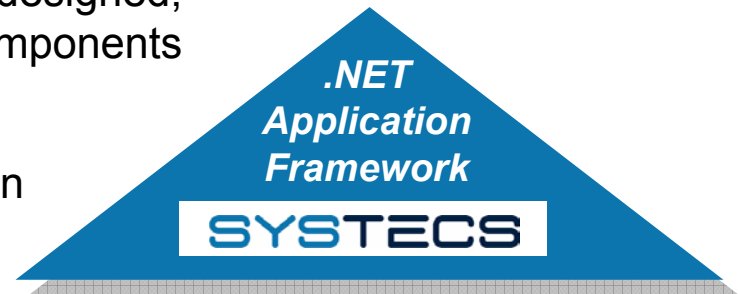
Common Practice

- **Software Designs are made from scratch for every CAE-Tool**
 - **Reason**
 - Existing Software Components are not ready for Re-Use in other Products
 - **Consequences**
 - Forces costs up
 - Increases time to market
 - Freezes valuable resources
 - Leads to incompatible products !

Improvement



- Introduction of a **Library** with re-usable Software Components
 - Library contains a powerful set of designed, tested and approved Software Components
 - Design decisions can now focus on Application relevant area
 - Application Architecture is derived from Architecture of the SYSTECS .NET Application Framework
 - Nevertheless the Software Design is free



Common Practice

- **Tools are monolithic or proprietary extensible**
 - **Reason**
 - Software is typically build with **proprietary** Architecture, Services (e.g. Logging, Resource handling), Extensions
 - No common extension mechanism is used !
 - **Consequences**
 - Maintenance of the Software is very difficult and time consuming

Improvement



- Introduction of a fully **decoupled** and **extensible Software Architecture**
 - A full extensible CAE-Tool must be composed **solely** of Components
 - Service Components
 - Graphical User Interface Components
 - Product Line Components
 - Application Components
 - Components are
 - **Connected** by a simple XML configuration
 - **Exchangeable** by changing the XML configuration on Deployment Level



... and here it is

Agenda

- Why do we need a new Methodology ?
- **The SYSTEMECS .NET Application Framework**
- What are the extensions to OO and CBD ?
- Summary

The SYSTEMECS .NET Application Framework ...

- The are three Views on a **CAE-Tool**
 - **Feature View**
 - **Software Architecture View**
 - **Deployment View**

The SYSTEMECS .NET Application Framework ...

- The Design of the **Software Architecture** of a CAE-Tool will start with the description of
 - **Tier's** or **Layers** and
 - **Patterns**
- Finally Layers and Patterns will be realized by
 - **Software Components** and its **Classes**

The SYSTECS .NET Application Framework ...

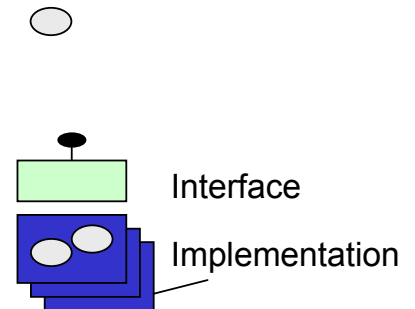
- The **Software Architecture** of a CAE-Tool consists of

Software Architecture

- **Classes**

- **Components**

- Components **consists** of Classes
- Components will be defined by an **Interface & Implementations**
 - Interface and Implementation will be clearly separated
 - A Component may have **several** implementations for the same interface



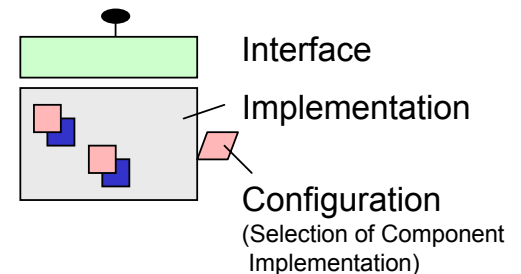
The SYSTECS .NET Application Framework ...

- The **Software Architecture** of a CAE-Tool consists as well of

- **Plug-Ins**

- A Plug-In **consists** of Components
- A Plug-Ins aggregates the **Interface & Implementations** of the **Components**
 - Interface and Implementation will be clearly separated
- A Plug-In includes a **Configuration**
 - The Configuration defines the selected Implementation of the Component

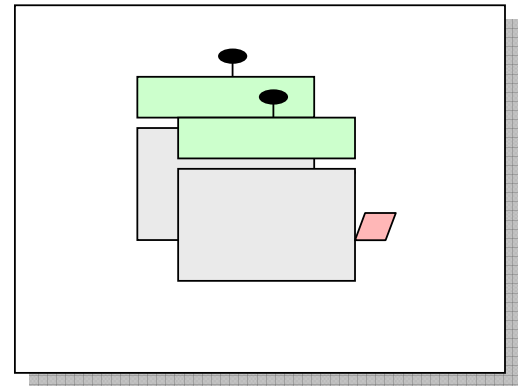
Software Architecture



The SYSTEMECS .NET Application Framework ...

- And finally ...
 - A CAE-Tool **consists** of Plug-Ins

Software Architecture



The SYSTEMCS .NET Application Framework ...

- A CAE-Tool consists on the **Deployment Level** of
 - **Assemblies** and **XML Files**
 - For both an **Interface & Implementation** of a Component and Plug-In there exists an **Assembly**
 - For each **Configuration** there is an **XML File**

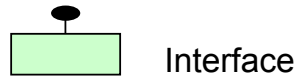
The SYSTEMECS .NET Application Framework ...

- **Software Architecture and Deployment** are now **identical** !
 - This enables **real Re-Use** on Deployment level !

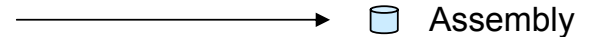
The SYSTECS .NET Application Framework ...

Software Architecture → **Deployment**

- Class



Interface

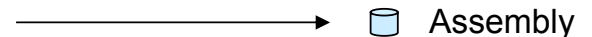


Assembly

- Component
Consists of Classes

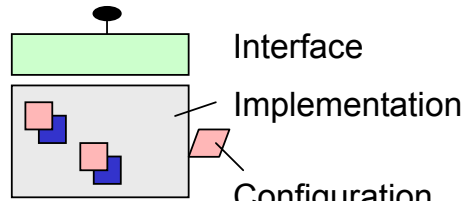


Implementation



Assembly

- Plug-In
Consists of Components

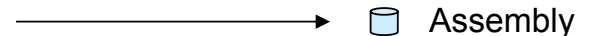


Interface

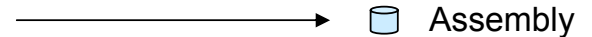
Implementation

Configuration

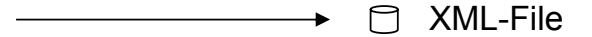
(Selection of Component Implementation)



Assembly



Assembly



XML-File

The SYSTECS .NET Application Framework ...



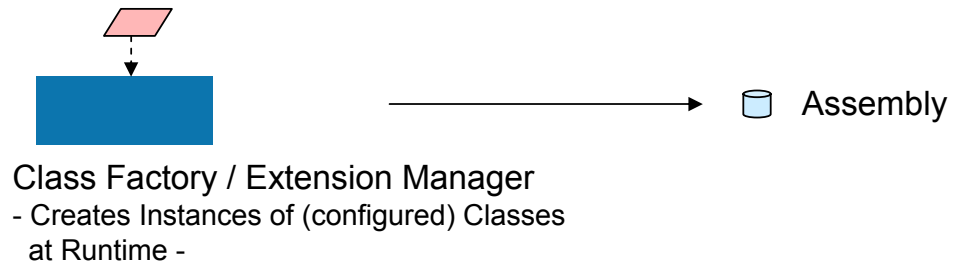
The SYSTECS .NET Application Framework ...

- A CAE-Tool must consist as well of a ...

Framework Kernel

- The Kernel loads the Configuration (XML File) at Runtime and creates the Components and Classes

Software Architecture → **Deployment**



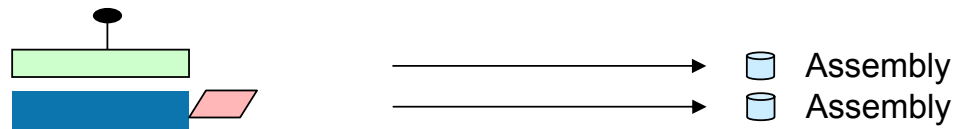
The SYSTECS .NET Application Framework ...

- A CAE-Tool must consist as well of ...

... Framework Libraries

- A Library itself is a Plug-In and contains re-usable Components
 - Common: UI, Resource, File, Property, String, ...
 - Automotive: C Parser, ASAM-MCD-2MC Reader/Writer/Editor, Test Script Engine, ...

Software Architecture \longrightarrow **Deployment**



UI Service, Resource Service
 Property Service, String Utility Service
 File Utility Service

The SYSTEMECS .NET Application Framework ...

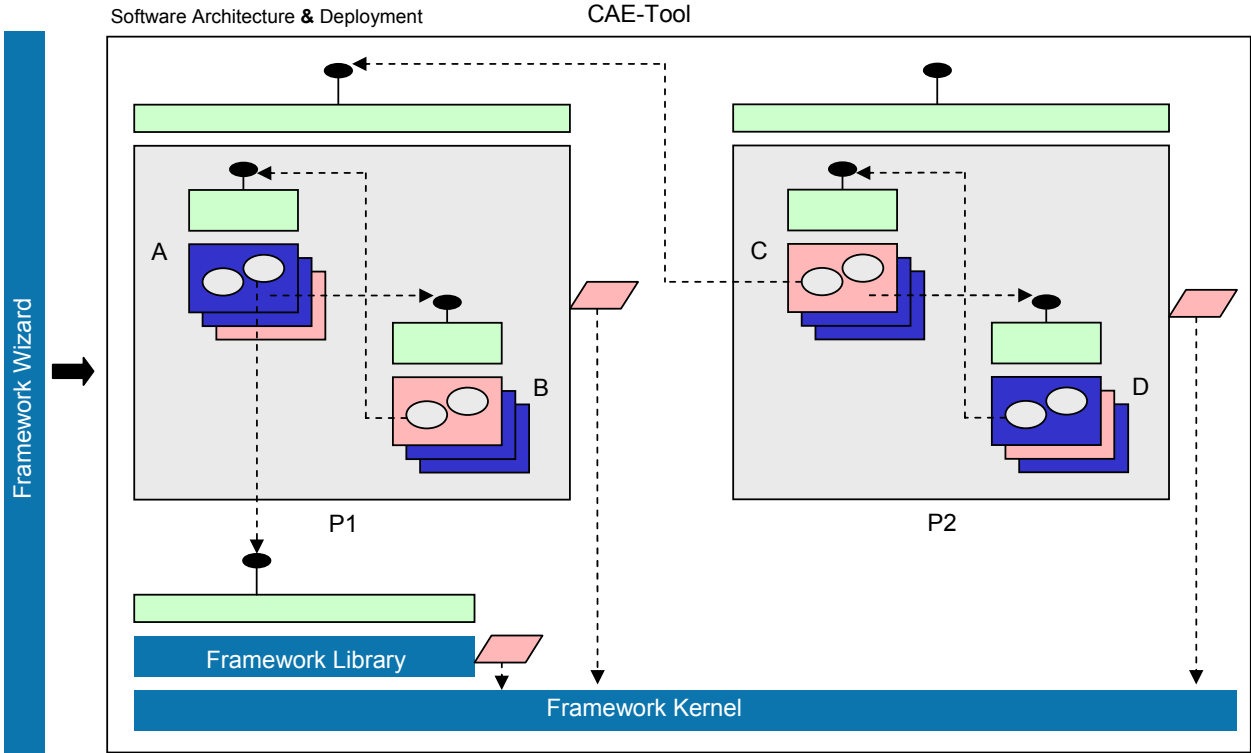
- Finally the infrastructure for the CAE-Tool generated automatically by a ...

Framework Wizard

- Automatic Generation of Make File, Directories, Templates for Classes, Components and Plug-Ins

The SYSTECS .NET Application Framework ...

- An Example ...



Agenda

- Why do we need a new Methodology ?
- The SYSTEMECS .NET Application Framework
- **What are the extensions to OO and CBD ?**
- Summary

Limitations of Object Orientation (OO)



- **Software is produced from scratch using OO Technologies**
 - Real Re-Use of Software-Components is only possible on **Deployment Level**
 - With OO Technology typically many Classes are mapped to one Assembly
 - Re-Use of Classes on Deployment level is therefore not possible
 - Mapping one Class to one Assembly on the other hand is not practicable since it leads to very inefficient development
 - Therefore **OO failed its promise of Re-Use**

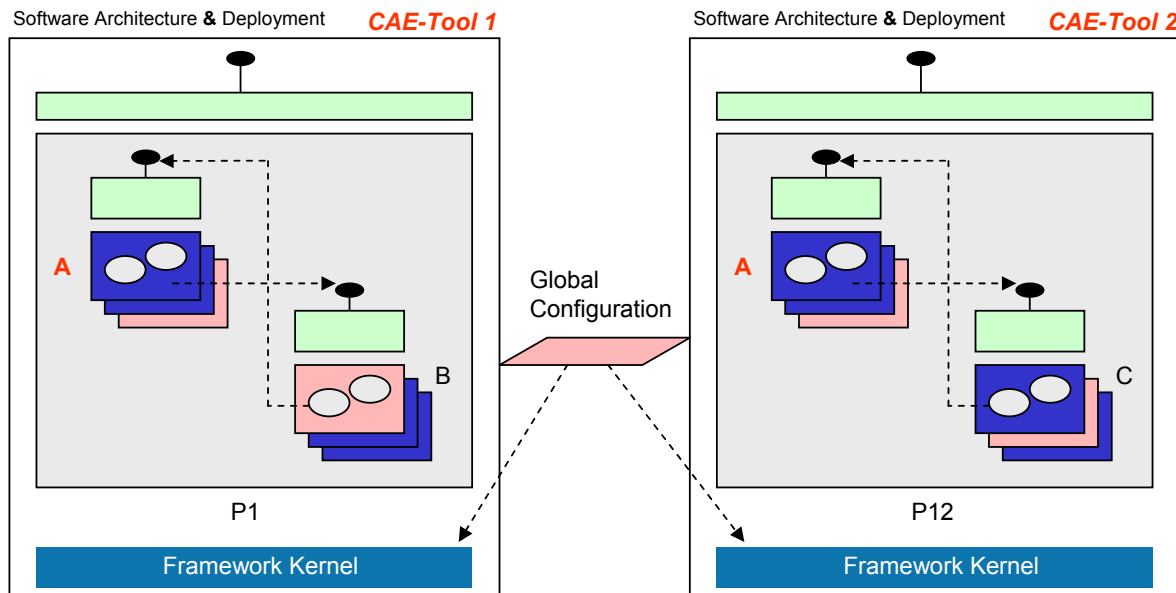
Improvement



- **Direct Relationship between Feature, Architecture and Deployment**
 - Software Components and Plug-Ins are mapped to Assemblies
 - Configurations are mapped to XML- Files
 - This enables Re-Use of Software Components and Plug-Ins on Deployment Level

Limitations of Component Based Development (CBD)

- **CBD – Supports global Configurations only**
 - Software Components are here unique on a PC
 - Example: CAE-Tool 1 and 2 **must** use the same Implementation of Component A

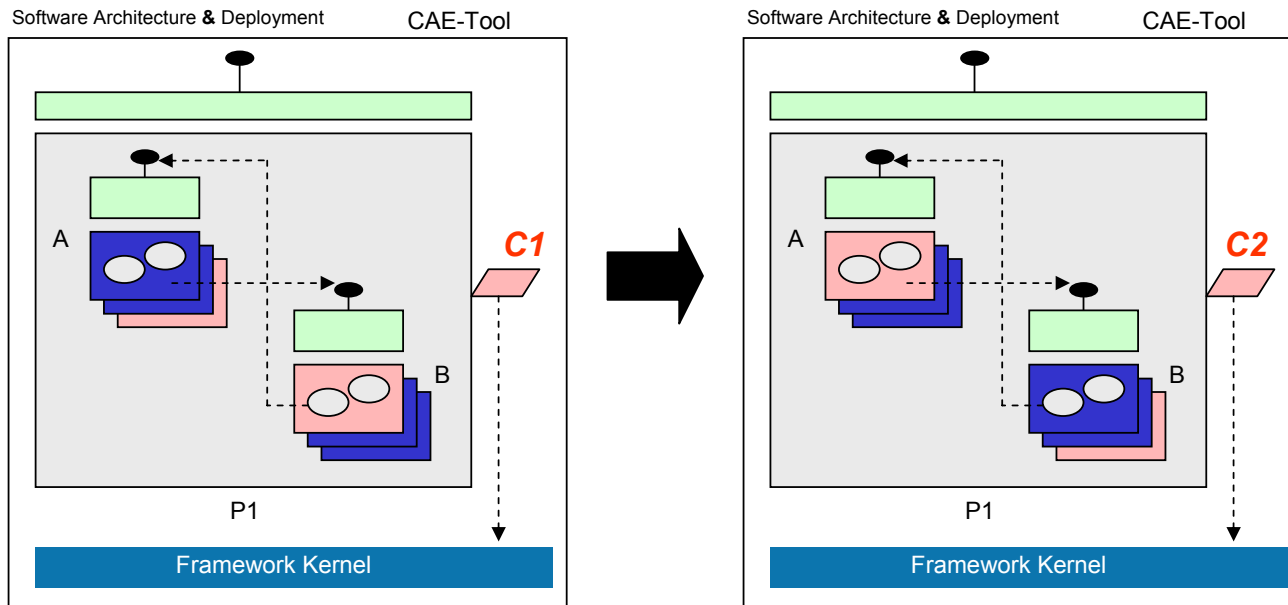


Improvement



- **Systematic Re-Use through local Configurations**

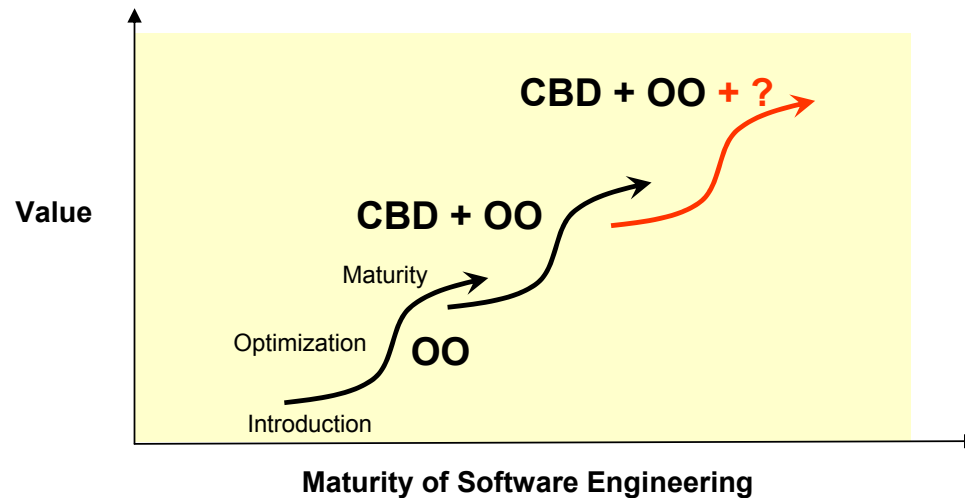
- Configuration on deployment level
- Extension or exchange of Components by changing the Configuration



Agenda

- Why do we need a new Methodology ?
- The SYSTEMECS .NET Application Framework
- What are the extensions to OO and CBD ?
- **Summary**

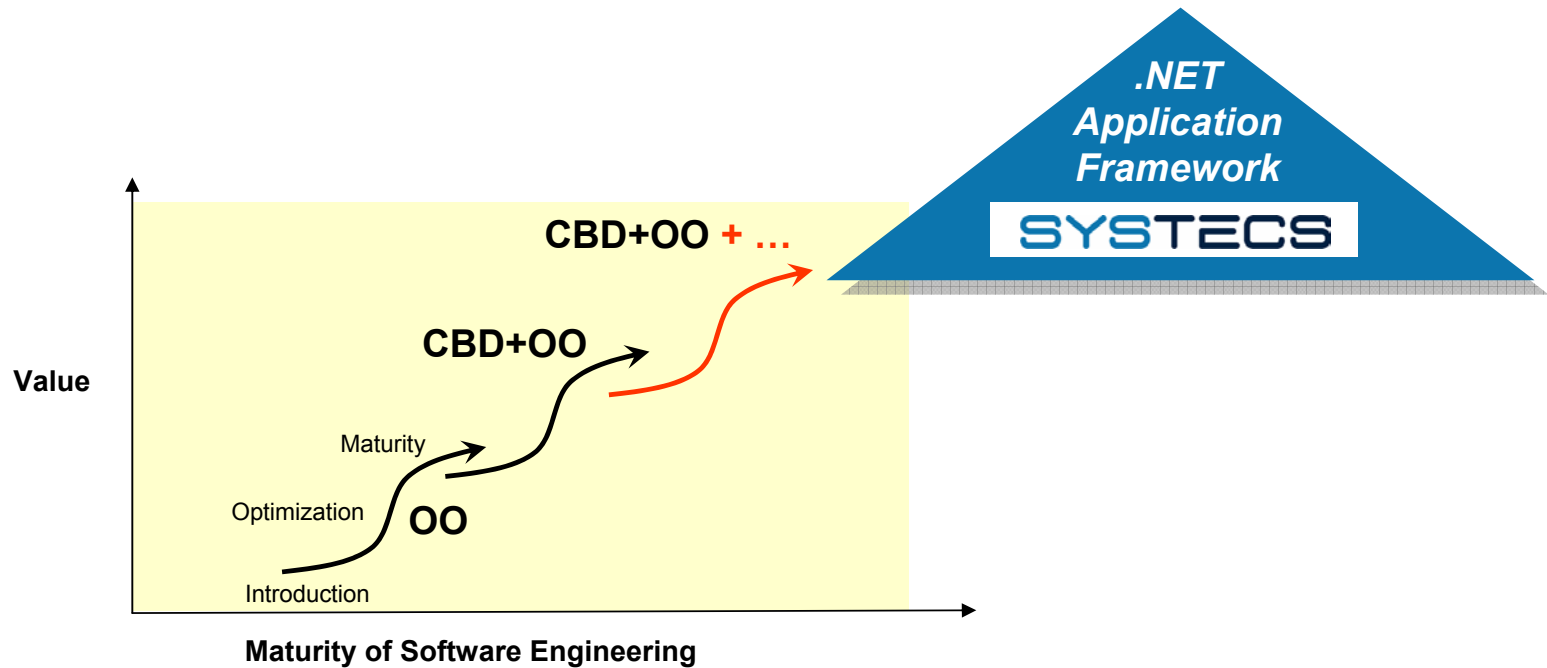
Summary



- Software development technology evolves primarily through disruptive changes called paradigm shifts
- The current Paradigms **O**bject **O**rientation and **C**omponent **B**ased **D**evelopment have reached a plateau
- **A paradigm shift is needed !**

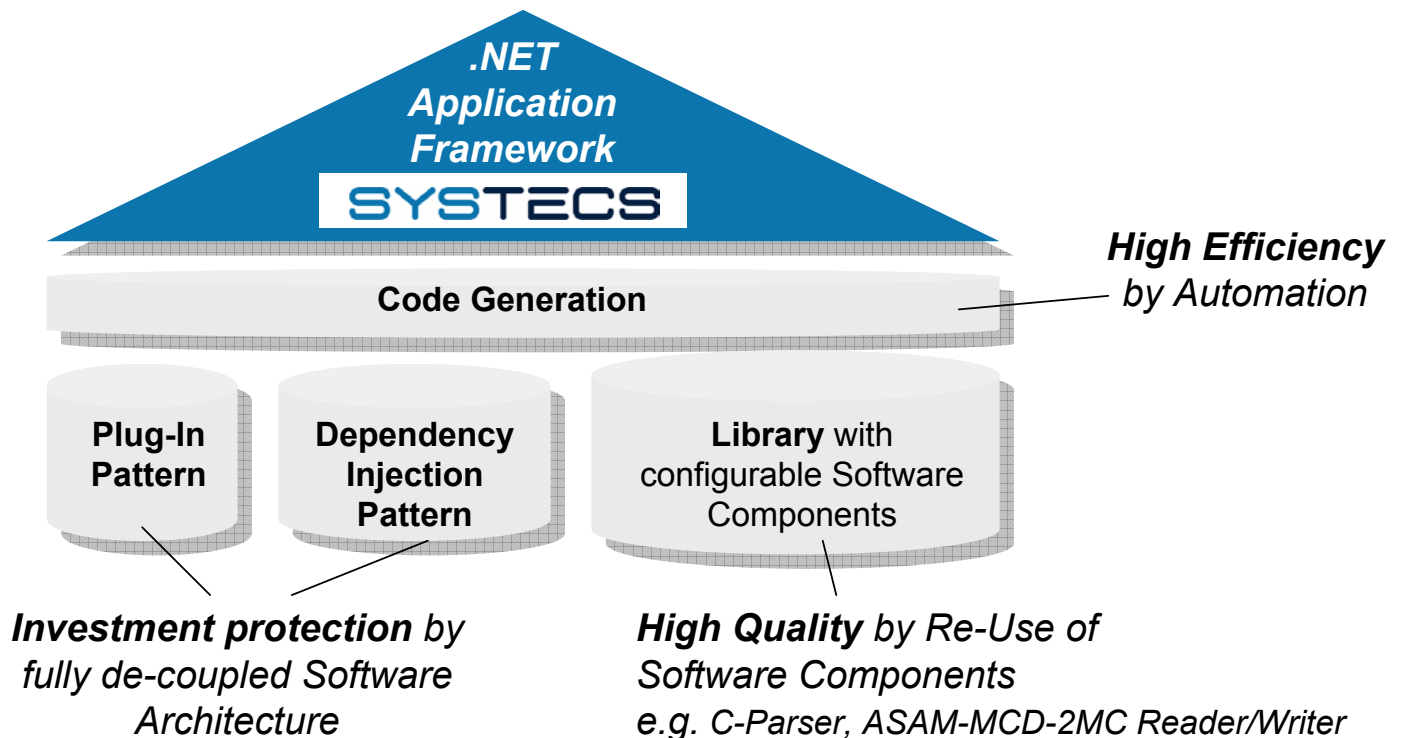
Summary

- SYSTEMECS supports the needed paradigm shift ...



Summary

- ... with the following Technologies



Contact

- Please feel free to contact us:

SYSTEMCS Informationssysteme GmbH

Kernerstr. 4

D 70771 Leinfelden-Echterdingen

Phone +49- 711- 16082 - 10

Fax +49- 711- 16082 - 8

sales@systemcs.com

www.systemcs.com

