Software Process Platform 4.1

Design & Generate

VBO Services GmbH

Agenda

- Introduction of Participants
- Customer Situation
- Overview and Objective
- Advantages and USP‘s
- Overview of the *Software Process Platform 4.1* Architecture
- Code-Generator Use Approach
- What Supplies Framework 2.0 + J2EE Templates?
- References
- Further Specifications

Overview and Objective

Through **SW-ProPlatform's and ISST Fraunhofer Institut's**
applied employment of
the **Generation of Code,** tasks are solved
through unification of

Design & Generate

in a **clear linear path**
with an **iterative approach**
during the project realization:
Quality, Speed, Flexibility and Costs

Advantages and USPs of *SW-ProPlatform 4.1* (I)

- Increased productivity (90-98% of Code is automatically produced), thus fast “Time-to-market” and cost reduction
- Constant adherence to the programming defaults leads to high quality, maintenance and investment protection
- Produced code is like “hand-written” and easy to read
- Transformation of the object models into executable programs without deviations from the standards (MDA)
- Interactive and iterative voting cycles with the specialized divisions based on executable basic application
- Developer begins with his task with an application in progress (not an “empty sheet”) which was produced automatically in accordance with specialized defaults

Advantages and USPs of *SW-ProPlatform 4.1* (II)

- Independence from Programming Languages
  - Templates can be adjusted for any language and used for the generation of code
  - Templates are also available for online help, program documentation, test cases and concept specifications
- Clear separation of generator and templates
- Templates are maintained without additional script languages by simple use of input masks
- For generation of code in arbitrary programming languages no adjustment of the generator engine is necessary
- Template work and object model are stored in a repository, which provides for data consistency
- Code re-use, modular structure, **S**ervice **O**riented **A**rchitecture

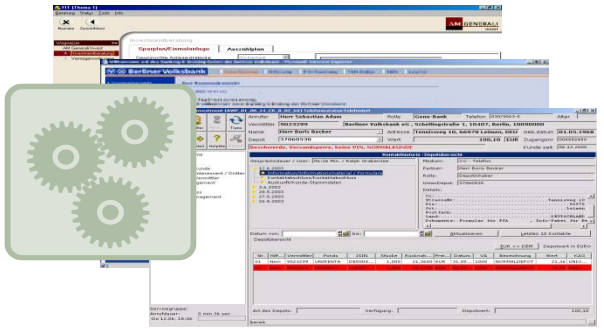
Software Process Platform: Overview of the architecture

Software Process Platform 4.1 is a tool-supported linear path,

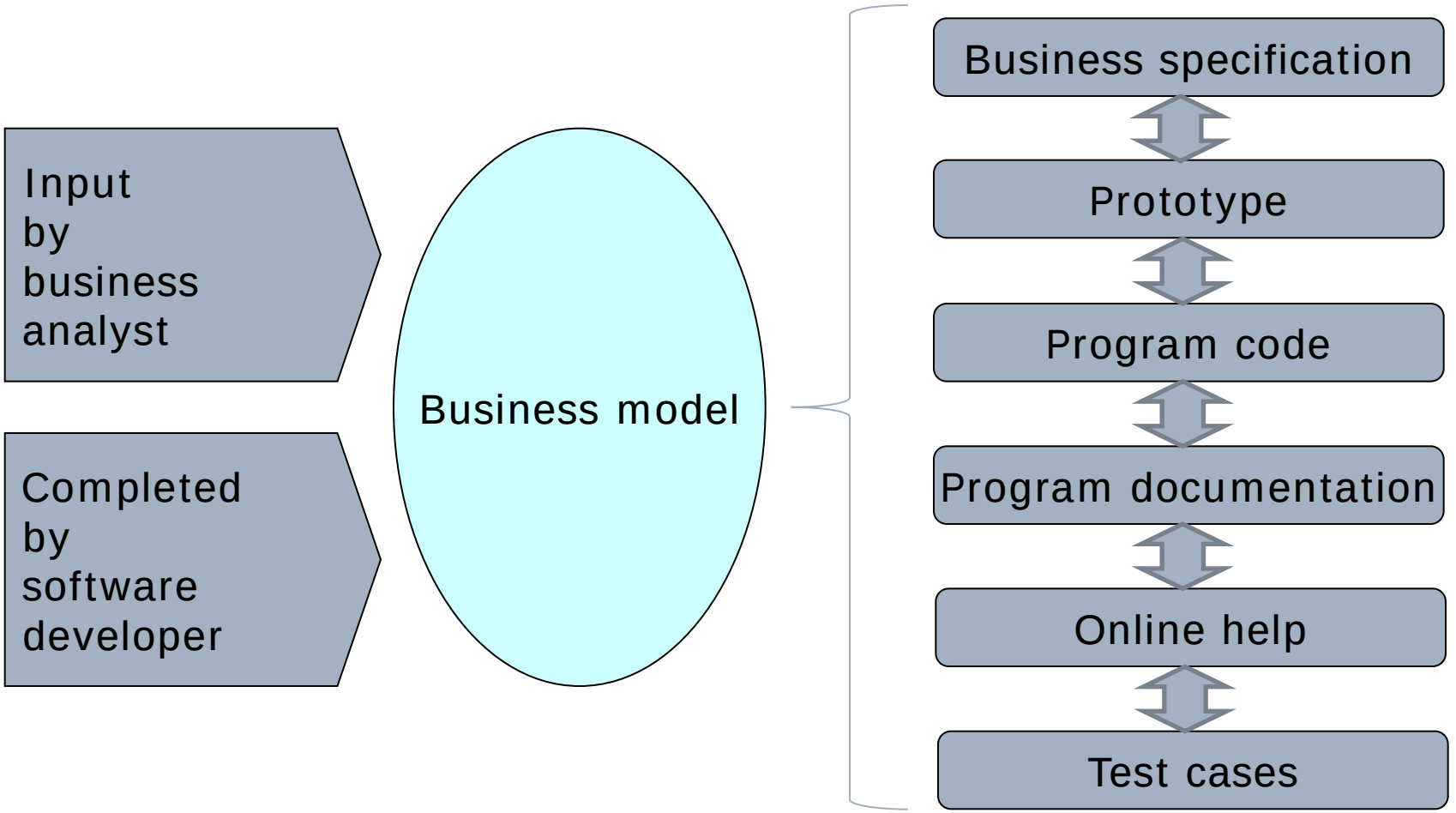
that on push of a button

- on the basis of a structural drawing (business process definition) and
- its available construction units (Templates and Framework)

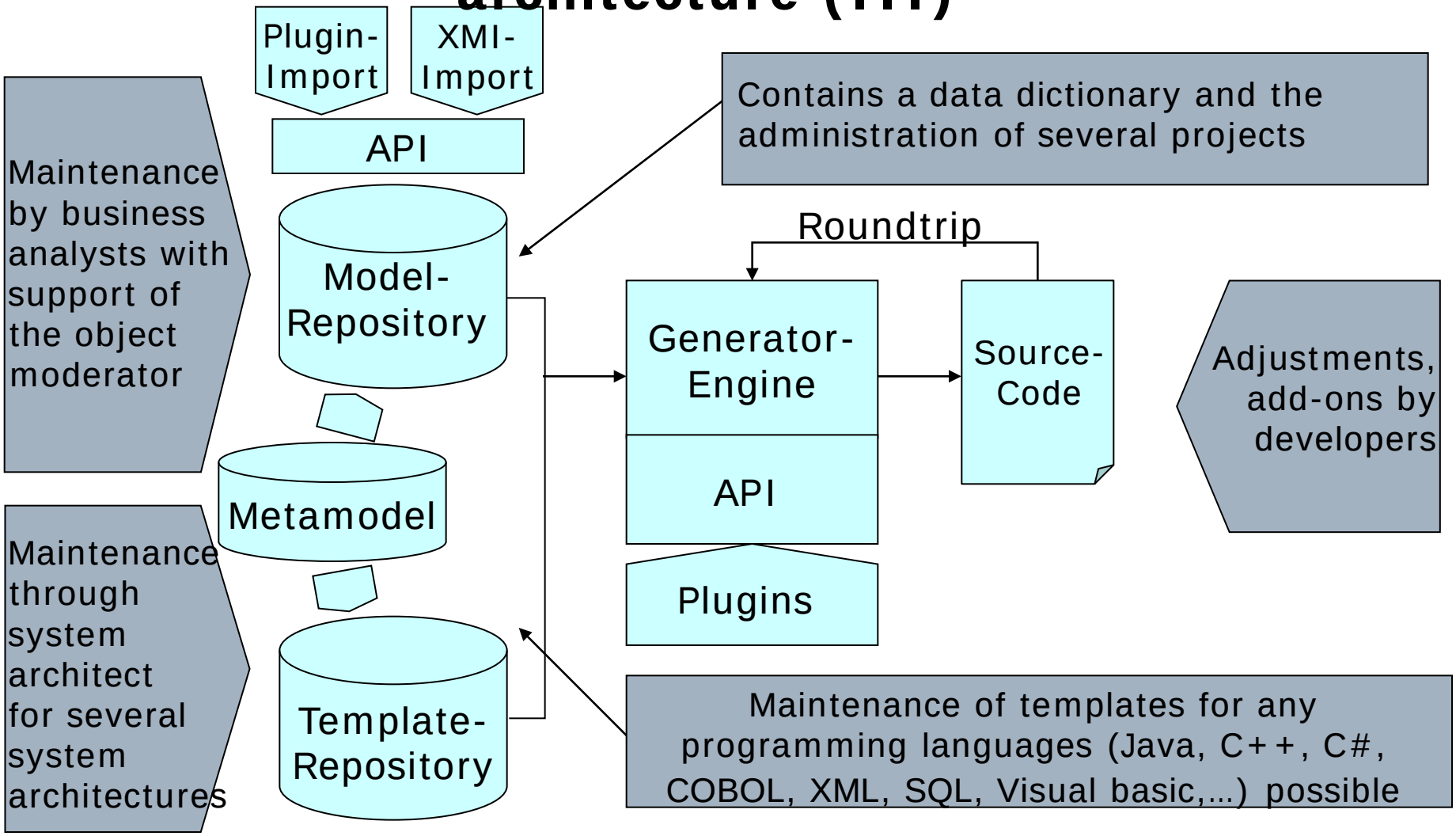
produces the application!



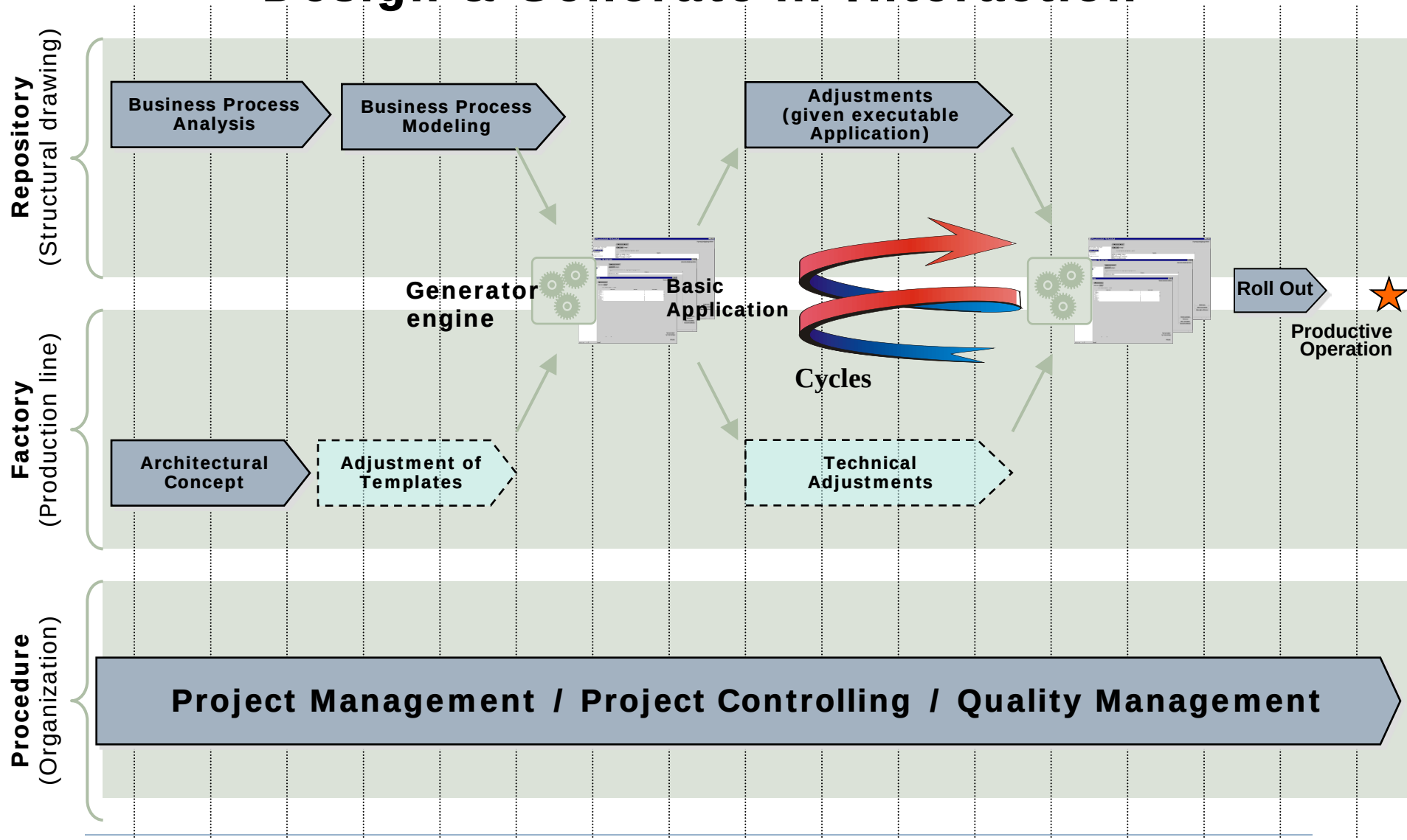
Software Process Platform: Overview of the architecture (II)



Software Process Platform: Overview of the architecture (III)



Design & Generate in Interaction



Functional Range of Business Frameworks (I)

Generally

- From high scaling enterprise applications to small installations with same architectural structure
- High re-use ability of code
- Description of general functions:
  - Multi-company management
  - Multi-language
  - Multi-user ability, locking mechanism
  - Administration of user, roles, rights, and departments
  - Reporting system/ reports
  - Process oriented operation, workflow support, reminder function
  - Online help
  - Administration and notice of messages (error, warning, info)

Functional Range of Business Frameworks (II)

Architectures

- Swing-Client ➔ Persistence (DB)
- Swing-Client ➔ JEE App. Server ➔ Persistence (DB)
- HTML-Server / Servlet Container ➔ Persistence (DB)
- HTML-Server / Servlet Ct. ➔ JEE App. Server ➔ DB

Generation

all specialized concepts, program documentations (JavaDoc), applications, descriptors, .ini-data, masks, database Structures etc.

Performance

Optimization for:

- Database accessing
- Mask Structure
- Client-Server-Communication

Functional Range of Business Frameworks (III)

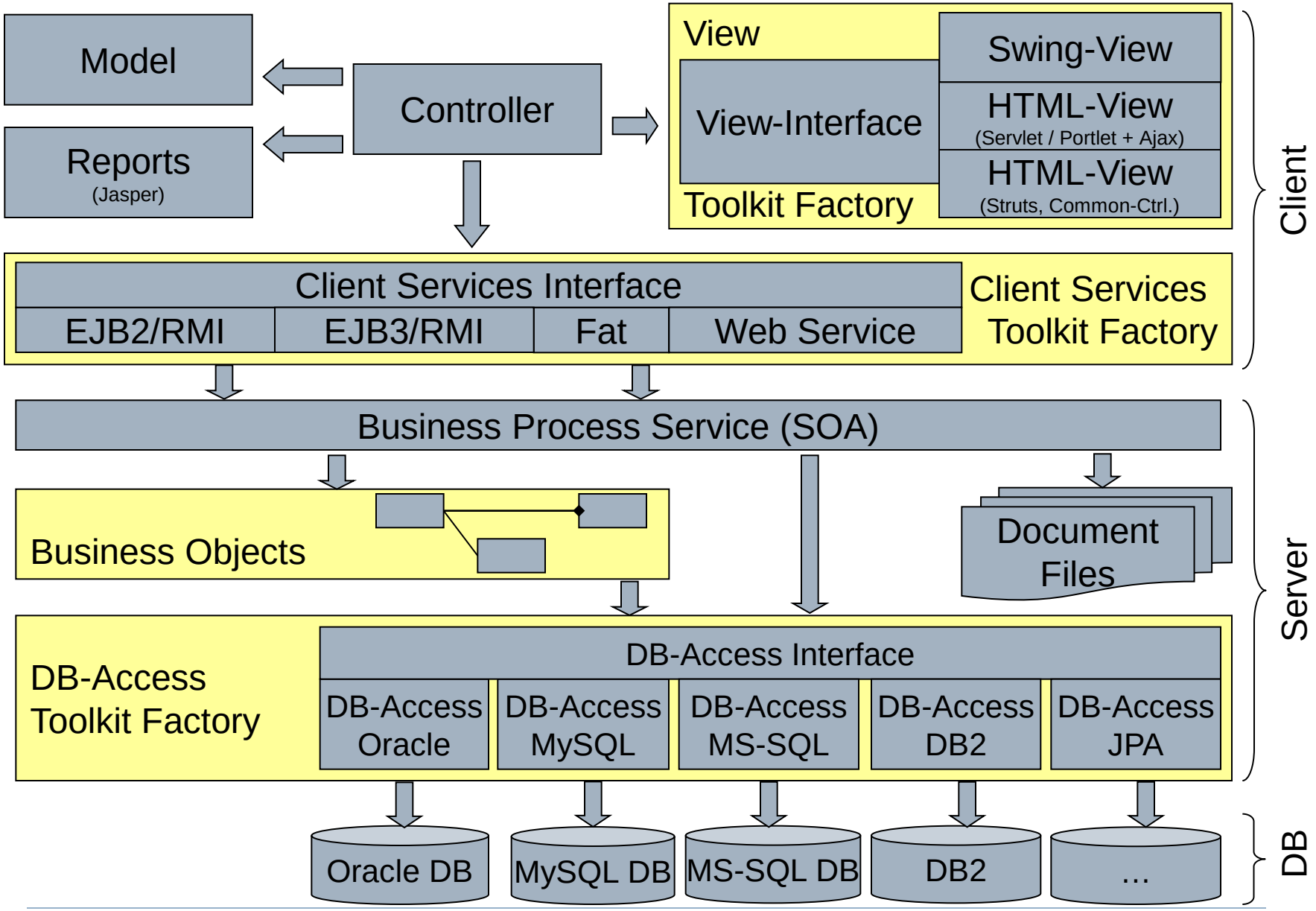
Client

- Frontend for Swing and HTML
- Further frontend types modularly expandable
- High flexibility in the mask organization and user interaction in a standardized structure
- Implementation of MVC design patterns
- Only view differently between Swing and HTML: controller and model will be reused
- Menu guidance
- Multi-Workflow/ Multi-Task with a Frontend Framework
- Hierarchical presentation and maintenance of data
- Input workflow/ assistant (Wizard)
- Communication with the server totally enclosed, replaceable and modularly expandable
- Clear separation from the business implementation
- Central customization of the mask organization with styles

Functional Range of Business Frameworks (IV)

Server

- Implementation of business logic into service oriented and reusable form
- Separation between business services and business model
- Examination of the data consistency
- Transaction management alternatively:
  - In server, per business service or service clip
  - Alternatively, also controlling of client transaction clip possible
- Data retention:
  - Multi-database and multi-database type accessing
  - Access to further database types modularly expandable
  - Multi-user control
  - Connection-Pooling (= saving of performance and resources)
  - Automatic production and adjustment of database structures
- Protocol logs/ logging


Model
Reports
(Jasper)
Controller
View
View-Interface
Toolkit Factory
Swing-View
HTML-View
(Servlet / Portlet + Ajax)
HTML-View
(Struts, Common-Ctrl.)
Client
Client Services Interface
EJB2/RMI
EJB3/RMI
Fat
Web Service
Client Services
Toolkit Factory
Business Process Service (SOA)
Business Objects
Document
Files
Server
DB-Access
Toolkit Factory
DB-Access Interface
DB-Access
Oracle
DB-Access
MySQL
DB-Access
MS-SQL
DB-Access
DB2
DB-Access
JPA
Oracle DB
MySQL DB
MS-SQL DB
DB2
…
DB

Summary

- The four steps:
  - Business Model
  - Requirement specification
  - Prototyping, user experience
  - Coded enterprise business software in production quality

  are organically interrelated and integrated with complete synchronization and without information loss
- This unique new approach allows:
  - Dramatic reduction of costs and project time
  - Standardized high quality of the applications
  - Easy transition between programmers due to standardized code (easily understandable code)
  - Software maintenance and improvement
  - Stable enterprise application with quality of standard software
  - Special design to perfectly match to the needs of the client
  - Short time interactive verification of business requirements with the customer based on running applications that evolve to the end product

References

- **Fraunhoferinstitut:** diverse interne und externe Projekte
- **Soltrx (Tochter der Commerzbank AG):**
  Ticket Management für den Wertpapier-Handel
  CWP (companyworld payment)
- **Kaiser‘s Tengelmann AG:**
  Warenwirtschaft Stammdaten Migrationsystem
  Konfigurationsmanagement
  Projektmanagement
  Personaleinsatzplanung für über 700 Filialen
- **Wüstenrot & Württembergische AG**: Beraterarbeitsplatz
- **FORMAXX AG**: CRM System für Versicherungsmakler
- **Großer Einzelhändler**: Gesamte Warenwirtschaft Plattform
- **INVERTO AG:** Ausschreibungsplattform, Vertragsmanagement, Rohstoffrechner
- **dbde Deutsche Bildung AG:** Softwareplattform für die gesamte Prozesssteuerung (Fondmanagement)
- **GEVA GmbH:** Internet Frontend für internationaler Zahlungsverkehr (SEPA)
- **ESCADA AG:** Vertragsmanagement
- **agentes AG:** Zeiterfassungssystem, Versicherungsmakler Plattform
- **SinnLeffers GmbH:** Personalbedarfsplanung
- **bäurer GmbH:** Entwicklung der ERP-Standardlösung
- **Nöll+ Partner Architekten:** Facilitymanagementsystem
- **Romservice Telecommunication:** ERP-System

Thank you very much

for your attention ☺

Software Process Platform 4: Motivation (I)

- It is unusual practice to copy and manually adapt old similar programs to new ones.
- If an error is found in the old used program then the correction must be done manually in all programs that resulted from it. This is usually no longer able or very difficult to accomplish.
- For modern 3-layer-applications a lot of code must be written. From this code, 70-90% is standard code which does not require business process intelligence. This code must however be written otherwise nothing would function.

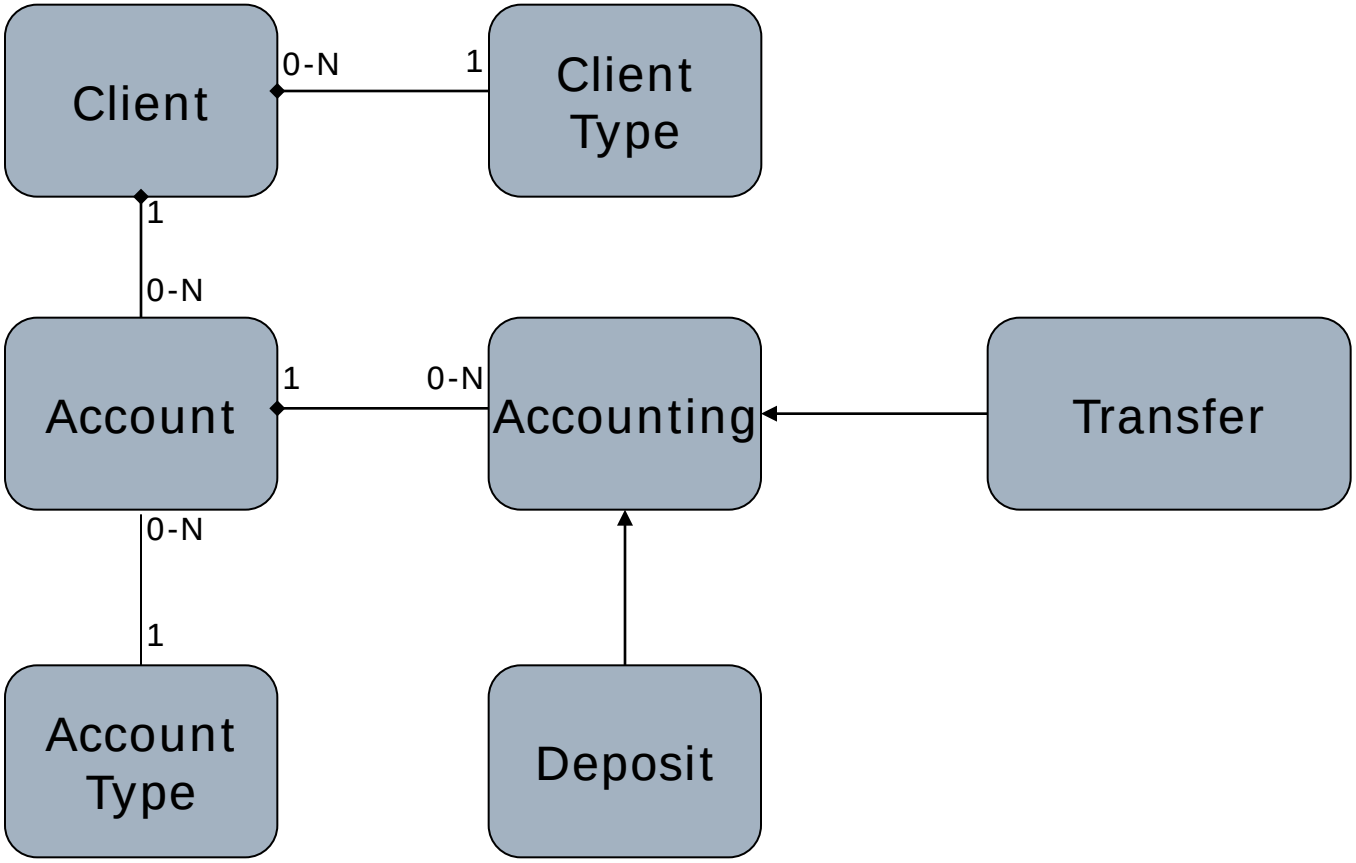
...

Software Process Platform 4: Motivation (II)

- If a new field must be added to a finished application then it must be manually and consistently written in the following:
  - In the detail mask and the tabular view
  - In the HTML / JSP-Forms with Web-Applications
  - In the communications layer Frontend<>Application-Server
  - In the object model on the application server
  - In the database access layer
  - In the database itself (SQL)
  - etc.

  Very much expenditure and very error-prone!

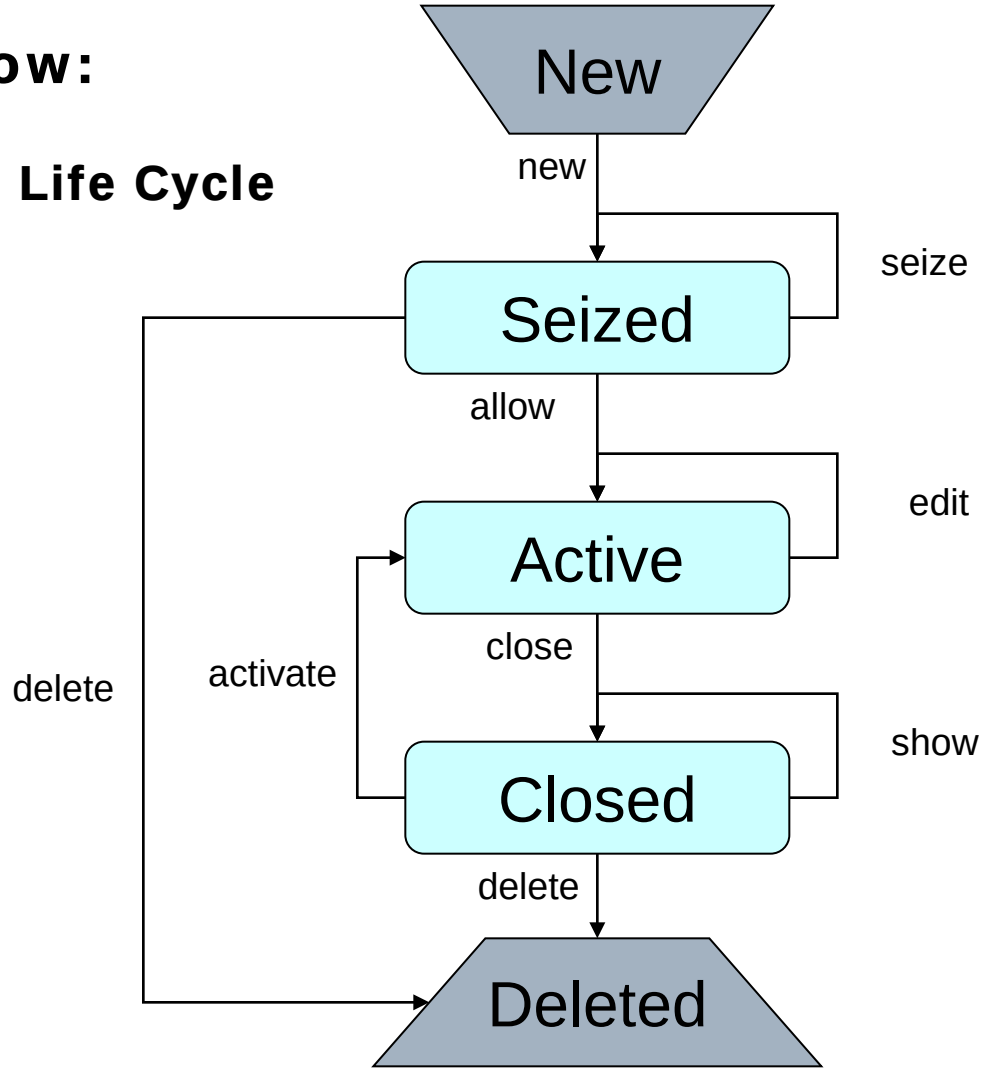
Example Applications (I)



Example Applications (II)

Workflow:

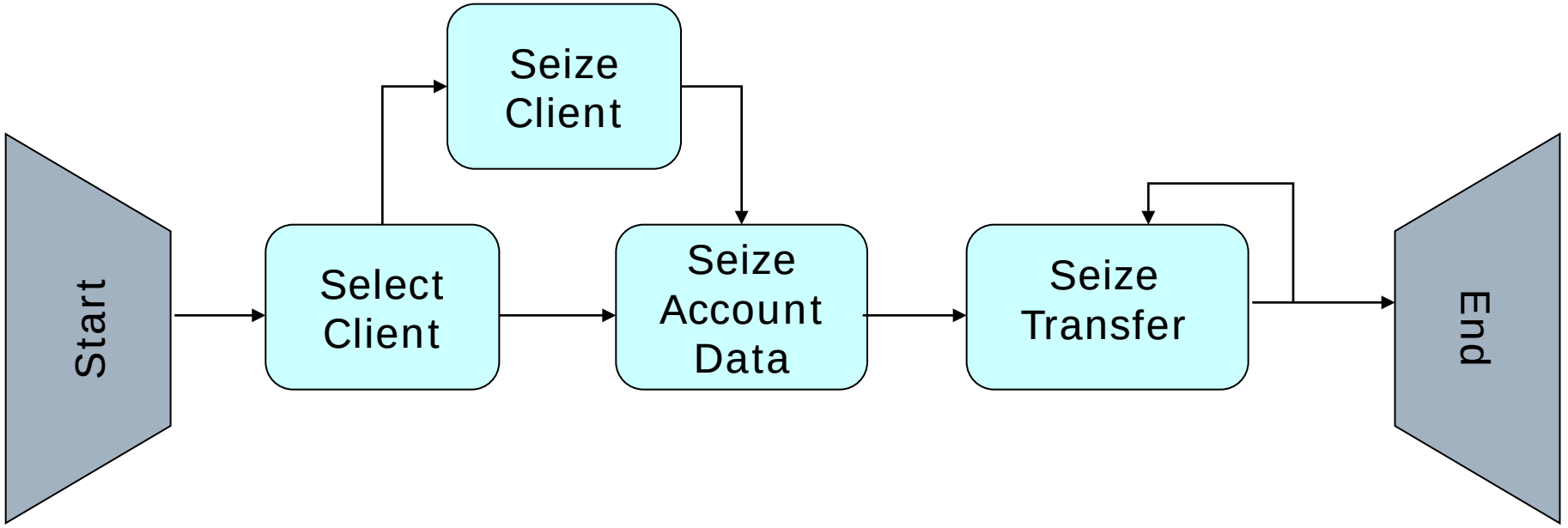
Account Life Cycle



Example Applications (III)

Workflow Dialog:

Account Collection



Example Applications (IV)

Workflow Collaboration:

Edit Account Type