Fast Application Development based on MDA

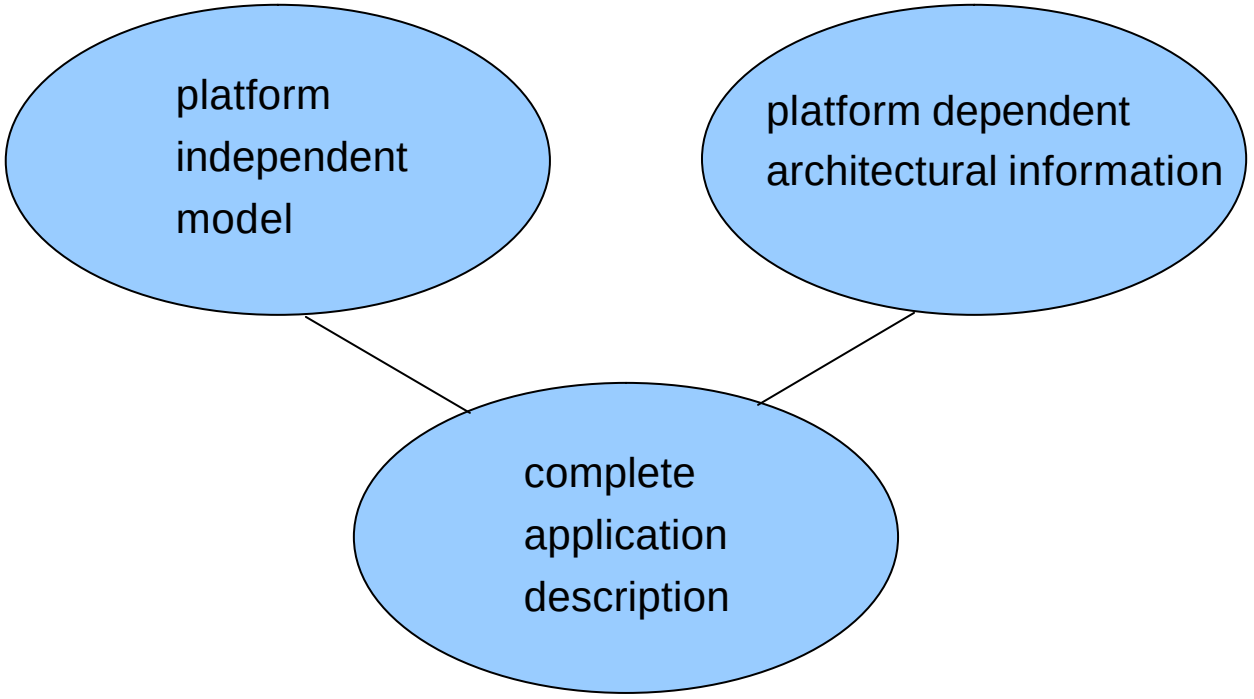
- **Christoph Blaue**

  University of Applied Science Westküste

  b+m Informatik AG

Model Driven Architecture (MDA)

The idea of dividing applications into two separate parts – one containing the business logic, the other containing business independent structural parts – leads to the notion of a Model Driven Architecture



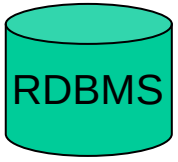
A Modern Application Platform (example)

View (GUI)

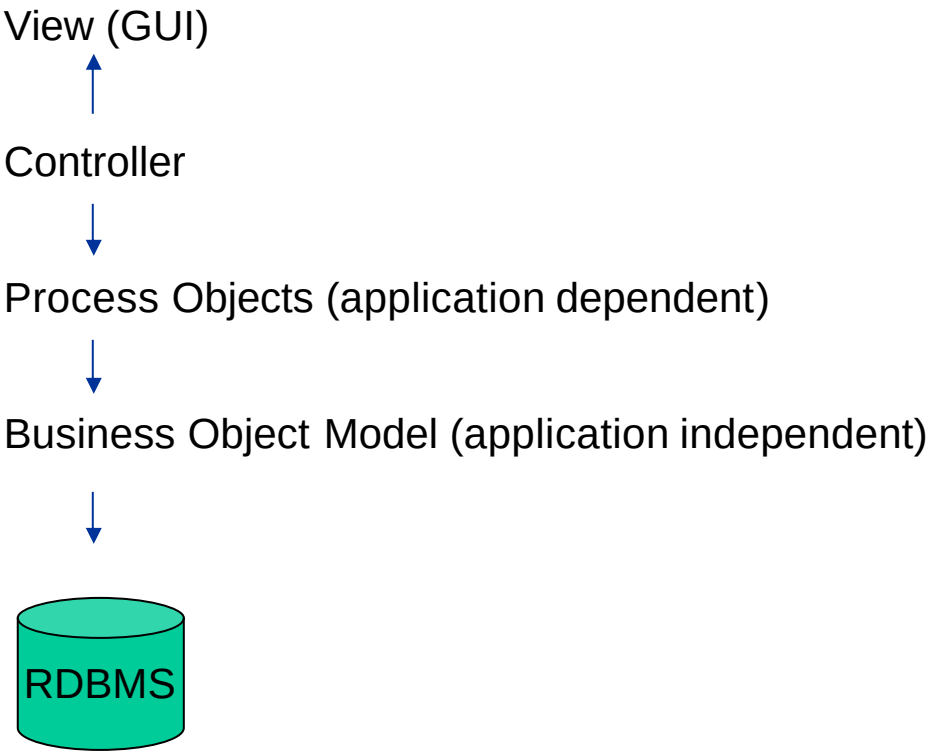
Controller

Process Objects (application dependent)

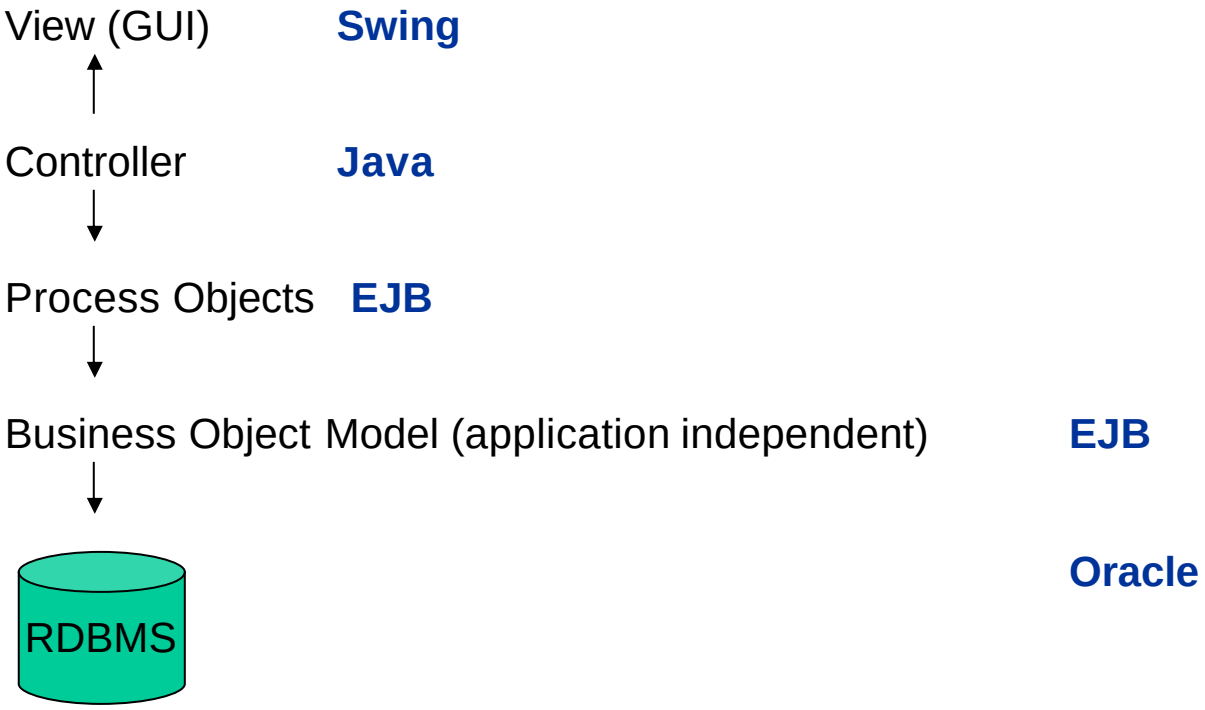
Business Object Model (application dependent)



A Modern Application Platform (example)

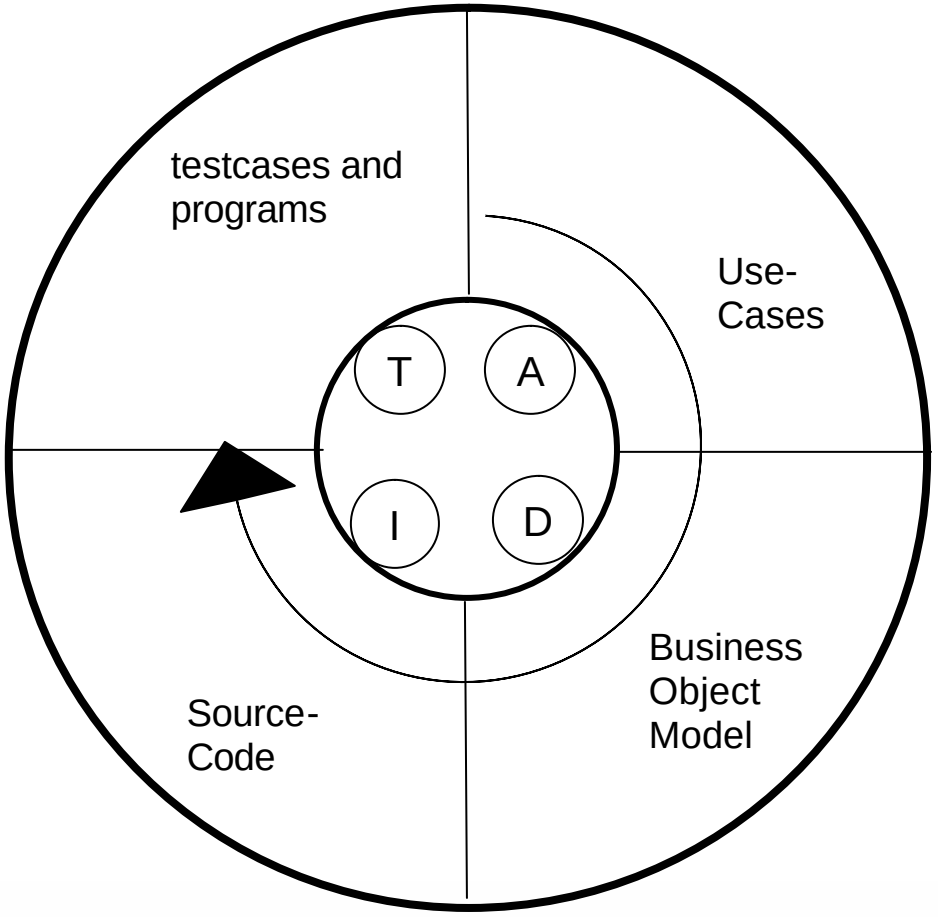


A Modern Application Platform (example)



Software Development Process (1/2)

incremental process and artefacts



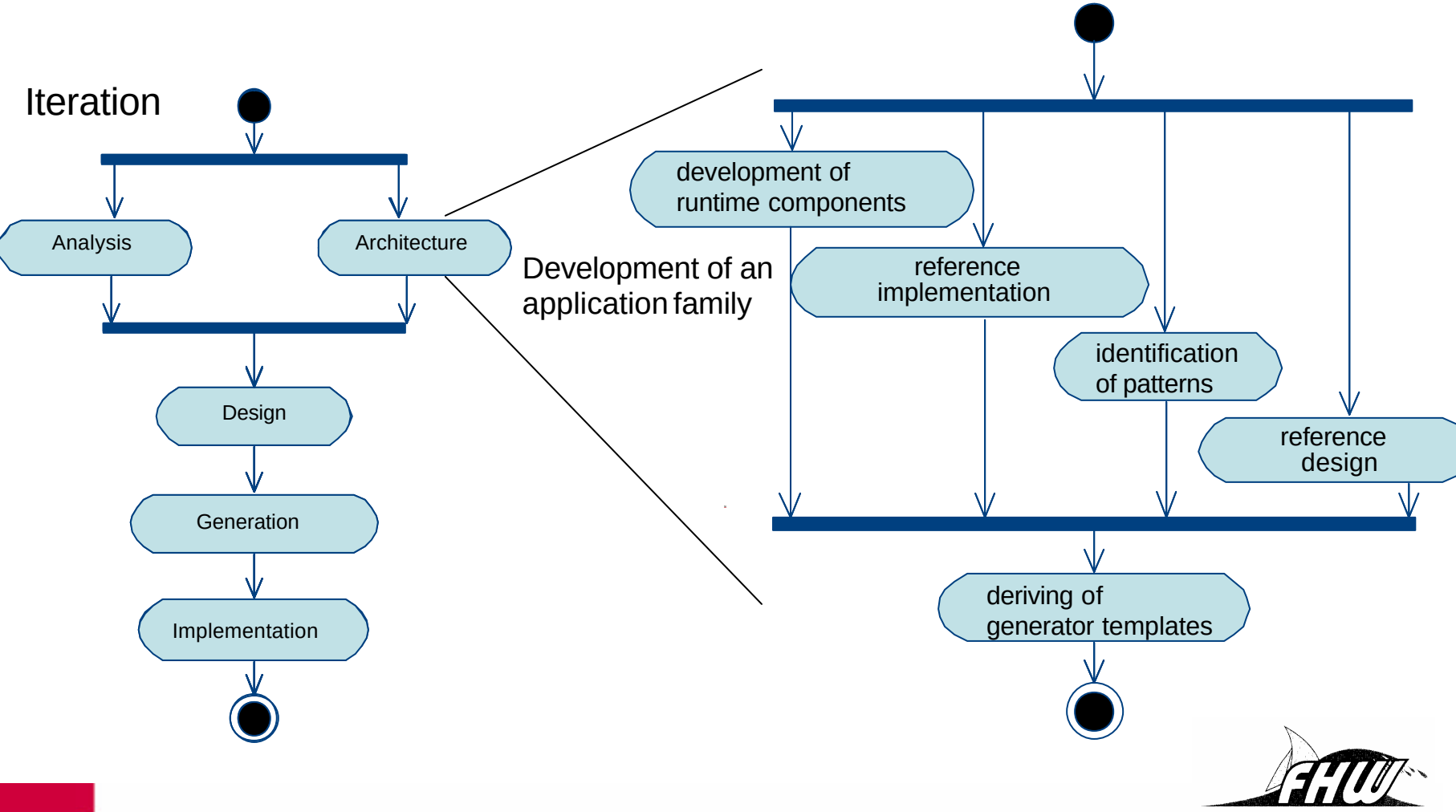
A = Analysis

D = Design

I = Implementation

T = Test

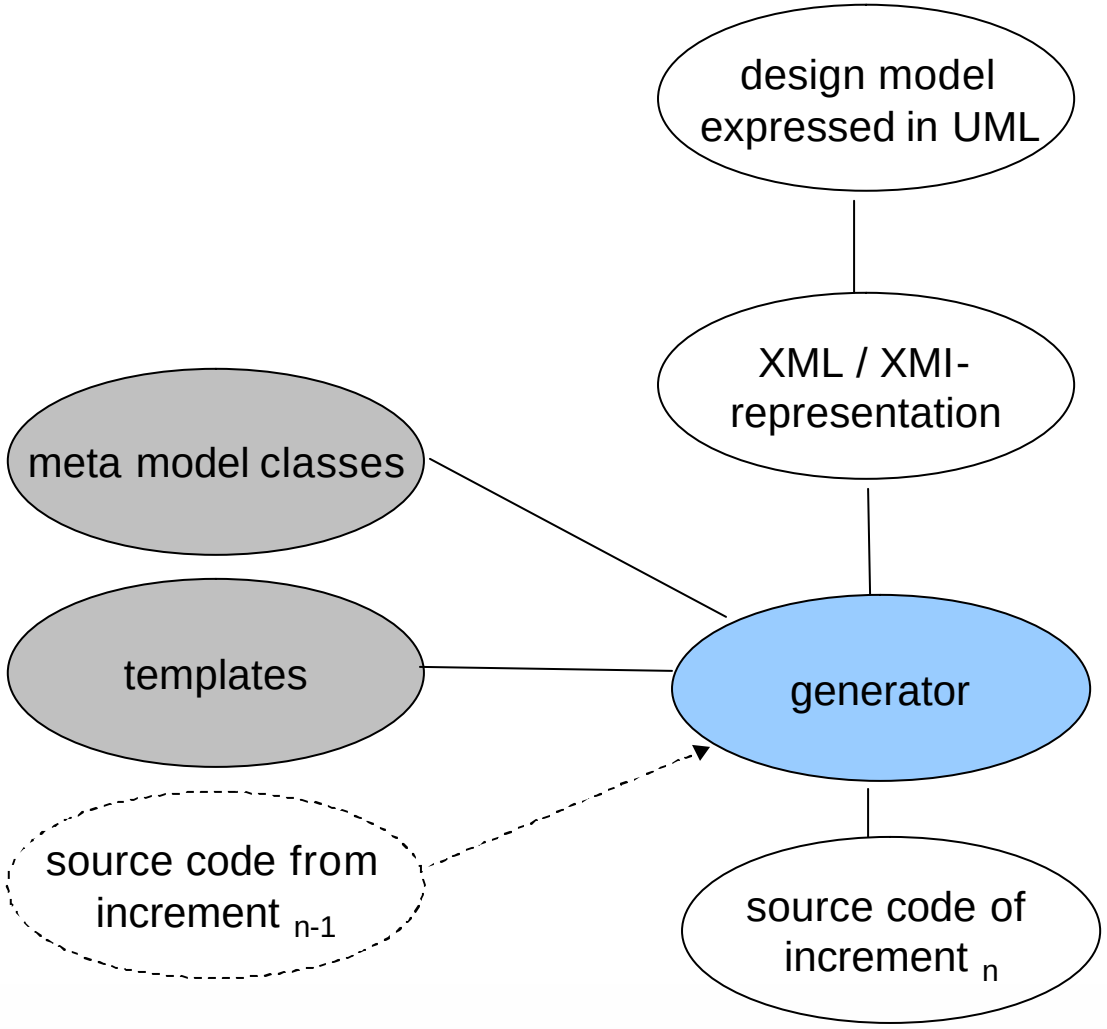
Software Development Process (2/2)



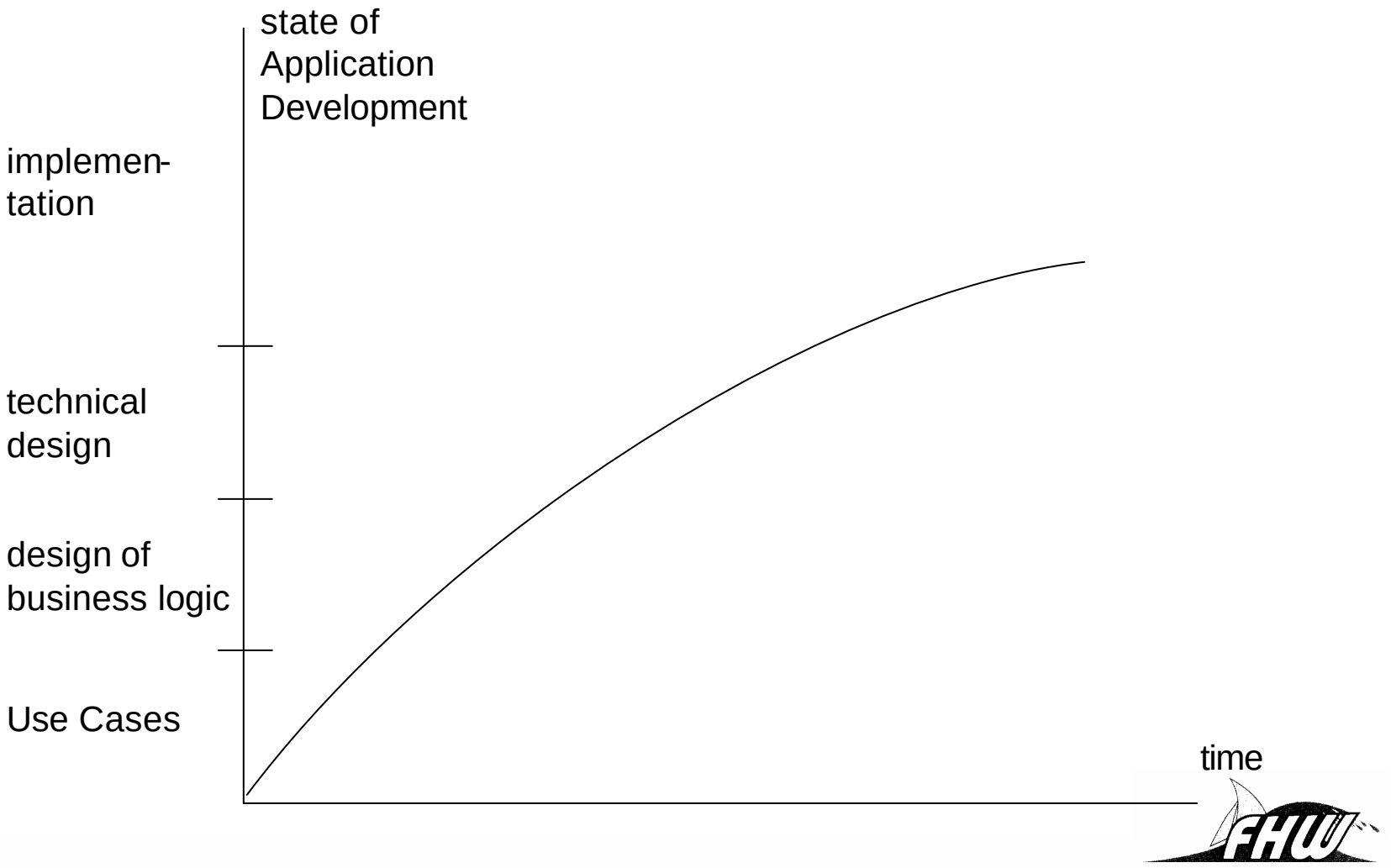
Characteristics of the Software Development Process

- iterative and incremental
- compact models by seperating architectural aspects and concrete business logic from the model
- the collection of information about the architecture determines the application family
- information about the application family is reusable
  - in a common development process by copy/paste (error prone)
  - in a generator based approach by an automated process

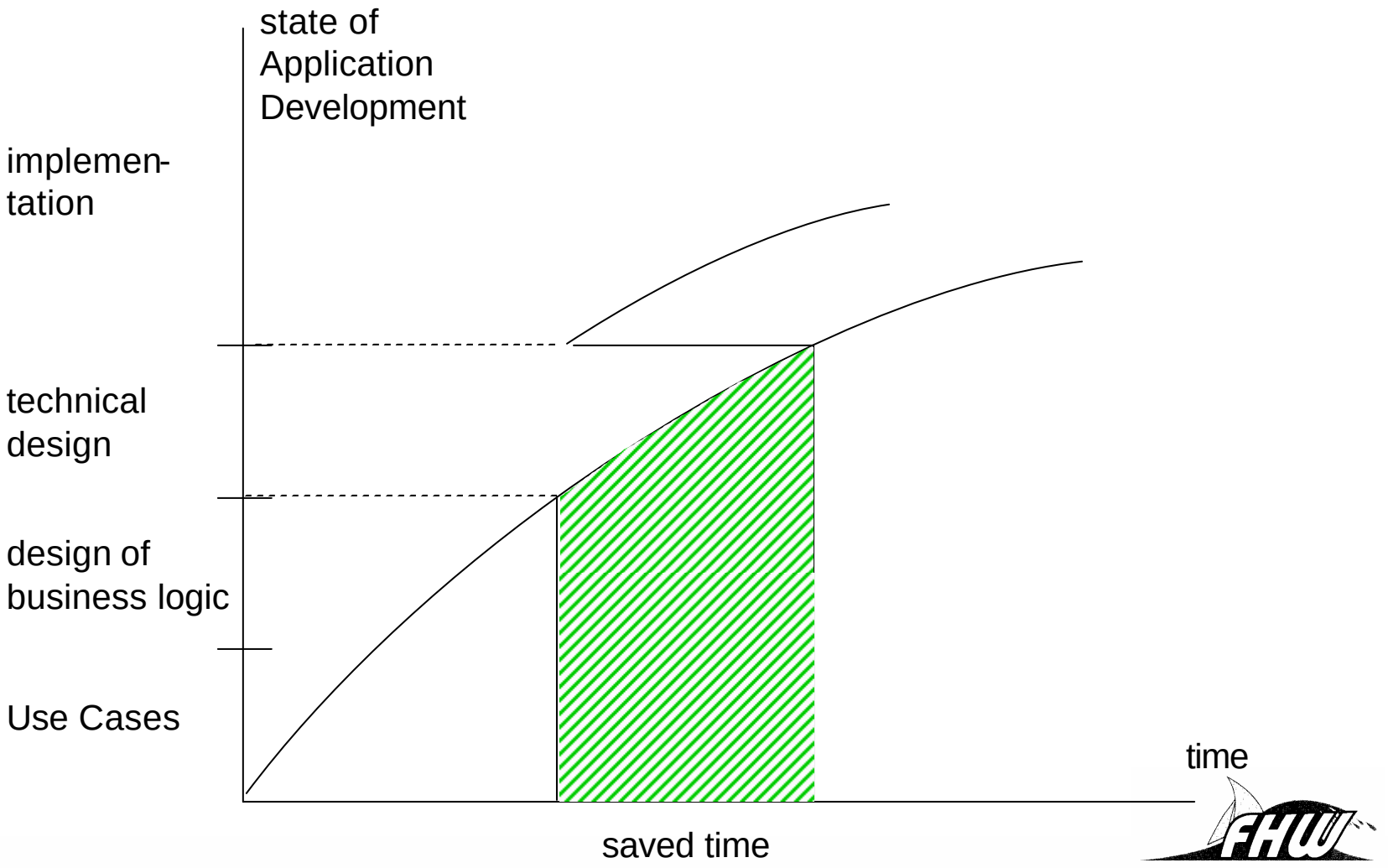
The b+m GeneratorFramework (TM)



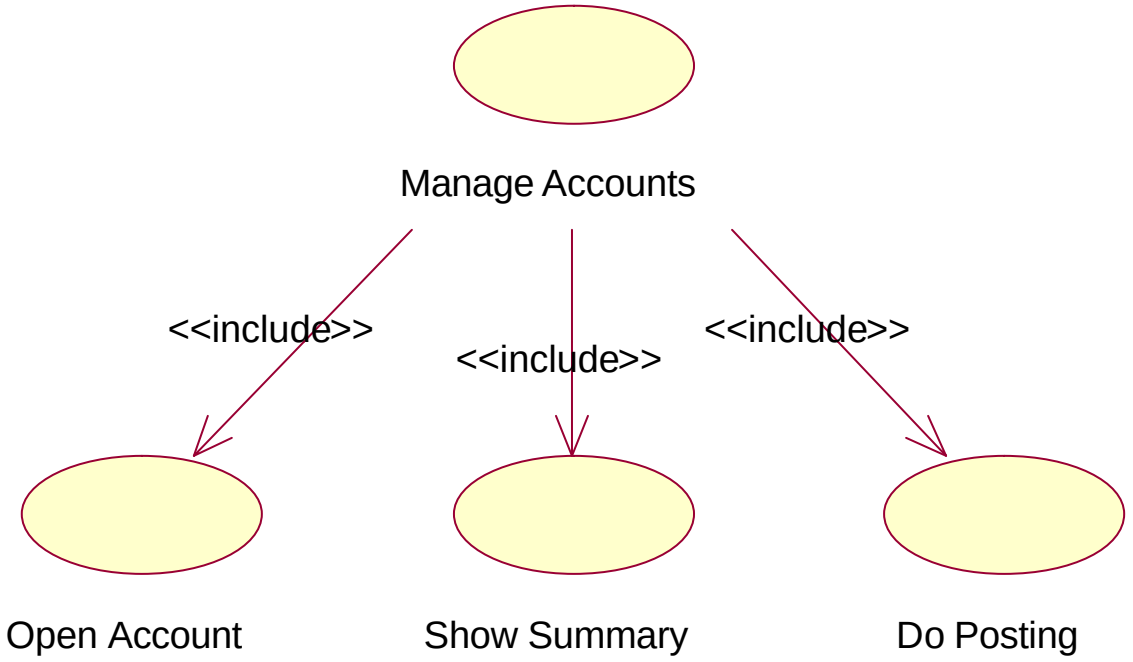
Classical vs. Generative Process (1/5)



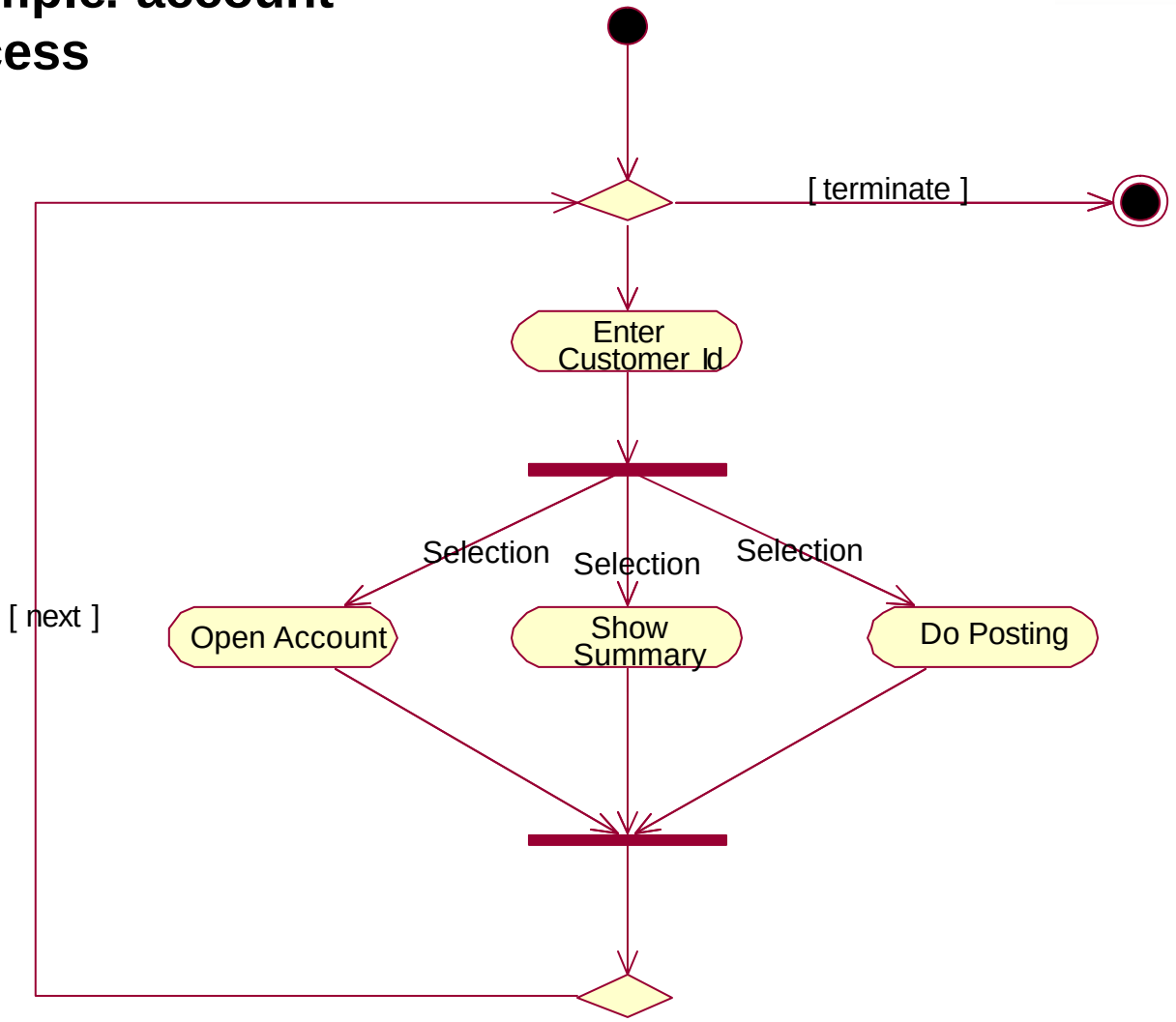
Classical vs. Generative Process (2/5)



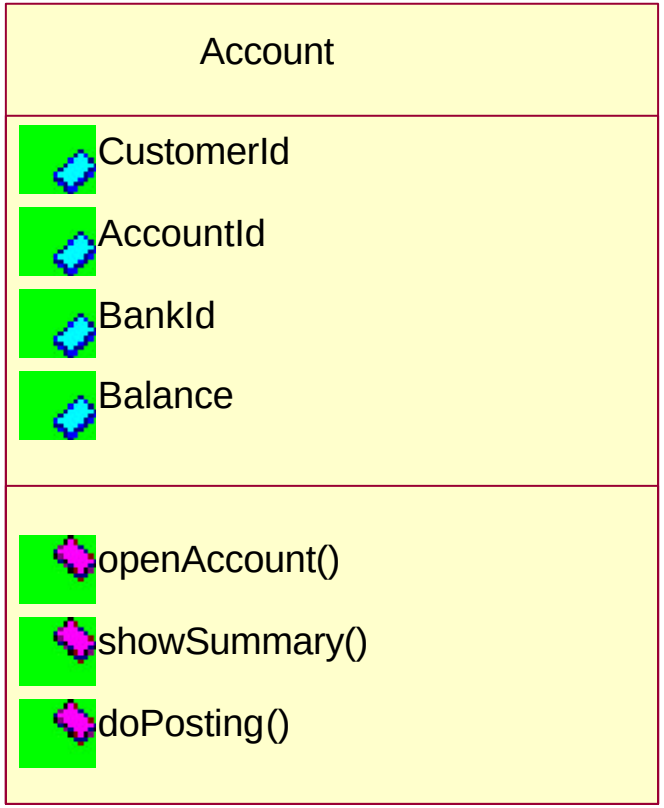
Example: account use case



Example: account process



Example: account business requirements



Template

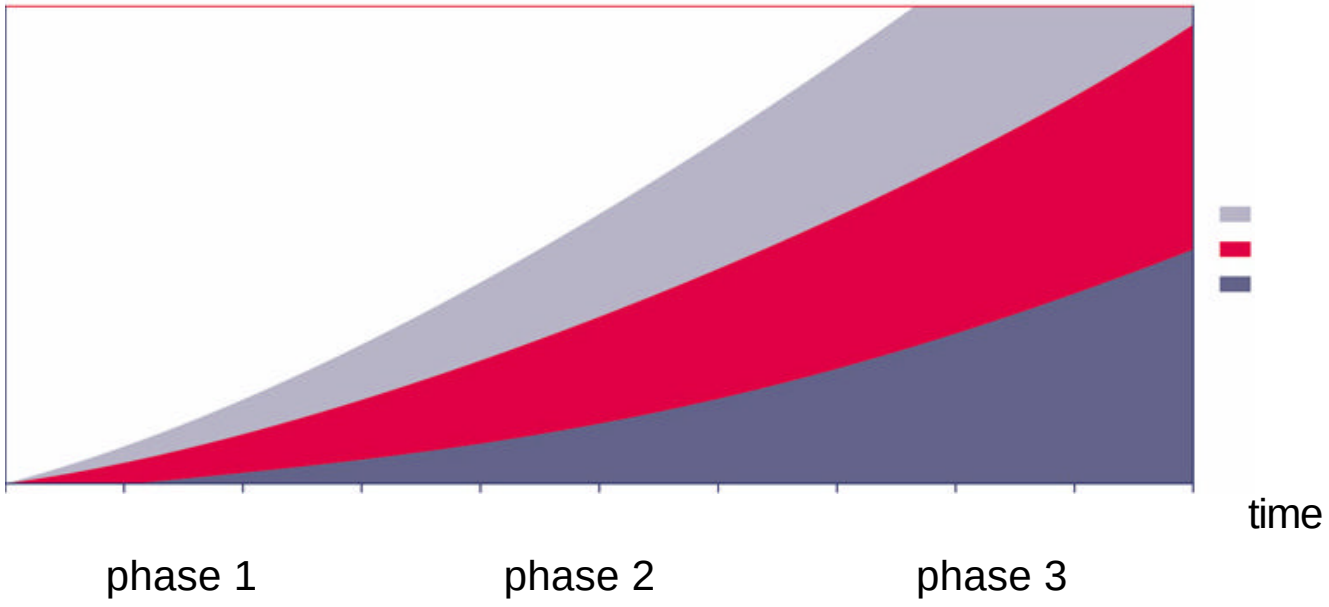
```
«FOREACH BusinessOperation AS curOperation EXPAND USING SEPARATOR
   "\n\n\n"»
«EXPAND JavaObject::Signature FOR curOperation» throws RemoteException {
      «PROTECT CSTART "//" CEND "" ID curOperation.Id»
      // please customize ...
      try {
         // ...
      } catch (Exception e) {
         throw new RemoteException("$EntityObject '«Name»' : couldn't execute
   operation
          '«curOperation»'!$");
      }
      «ENDPROTECT»
   }
«ENDFOREACH»
```

Example: account generated code

```
public int showSummary() throws RemoteException {
//PROTECTED REGION ID(3CF1E93C037E) START
// please customize ...
try {
// ...
} catch (Exception e) {
throw new RemoteException(
"$EntityObject 'Account' : couldn't execute operation ‘showSummary'!$");
}
//PROTECTED REGION END
}
```

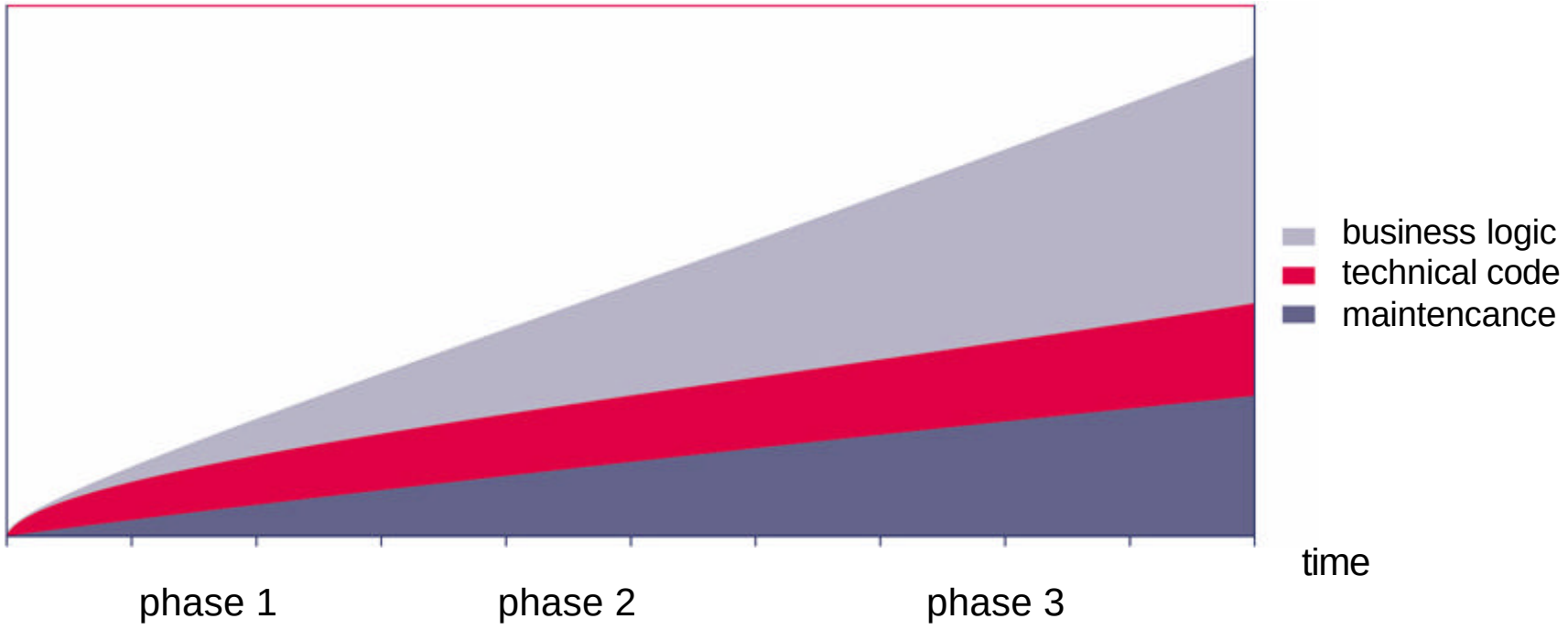

Classical vs. Generative Process (3/5)

costs for application development (without generator)



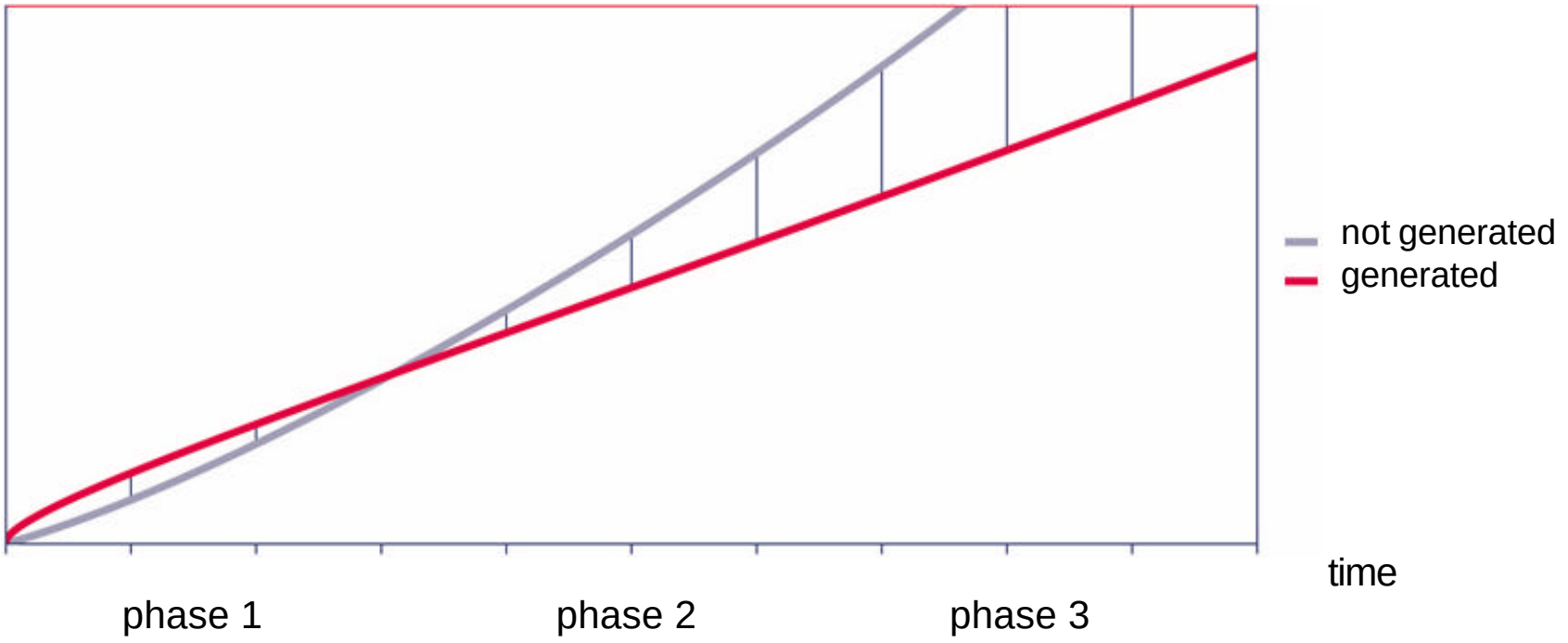
Classical vs. Generative Process (4/5)

costs for application development (with generator)



Classical vs. Generative Process (5/5)

comparison of costs



Concluding Remarks

- top down development
- model based source code generation based on platform independent models (PIM)
- models are part of the application, they are no longer „sugar to the customer“
- separation of business logic and architectural concerns
- incremental process, no round trip engineering
- only customer defined parts are generated
- iterating by means of freely definable protected regions
- not restricted to a specific field of applications
- enhanced speed and enhanced quality

