

# Core Concepts of a Process Model for Security Engineering

**Michael Hafner**

**Universität Innsbruck**

**Research Group „*Quality Engineering*“**

m.hafner@uibk.ac.at



# Overview

1. Introduction
2. The Security Process
3. A Case Study
4. Summary



# 1. Introduction

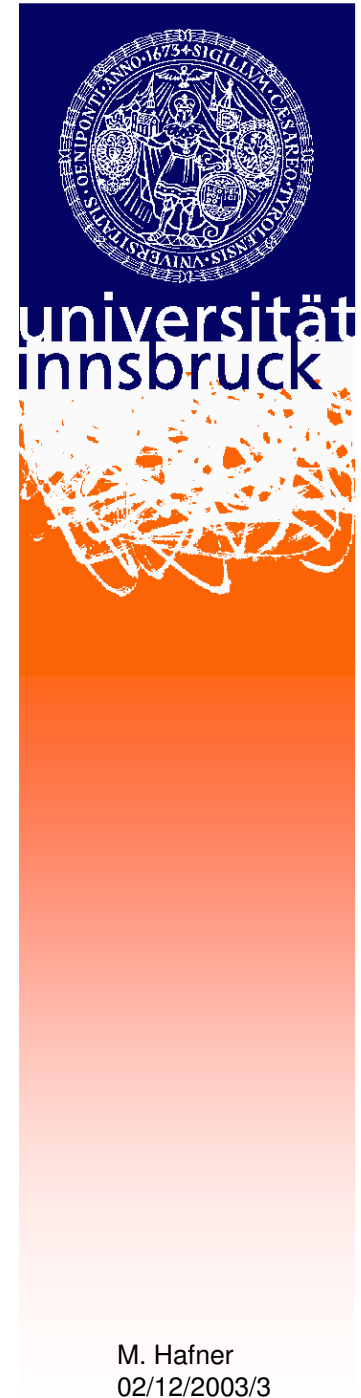
## Distributed Applications – Two Examples:

### ▶ E-Government

- » Execution of administrative procedures via internet
- » Example: registration of a newly formed company

### ▶ Healthcare

- » Electronic files of in-patients
- » Use of mobile devices at the bedside (daily diet registration, diagnosis etc.)



## Our Goal

- ▶ **Development of a Method for The Systematic Modelling of Secure Systems**
  - » Early, high level elicitation and documentation of security requirements
  - » Comprehensive risk analysis
  - » Systematic implementation of security measures
  
- ▶ ***“Security is a requirement which has to be considered in all stages of development and which needs particular modeling techniques to be captured.”***



# The „Core Artifacts“

## *Application Level*

**Business  
Process Model**

- ▶ Analysis and modelling of work processes (activity and class diagrams)

**Use-Case-Model**

- ▶ Identification and description of functionality (use cases)

**Application  
Architecture**

- ▶ Description of abstract message flow
- ▶ Identification of logical components

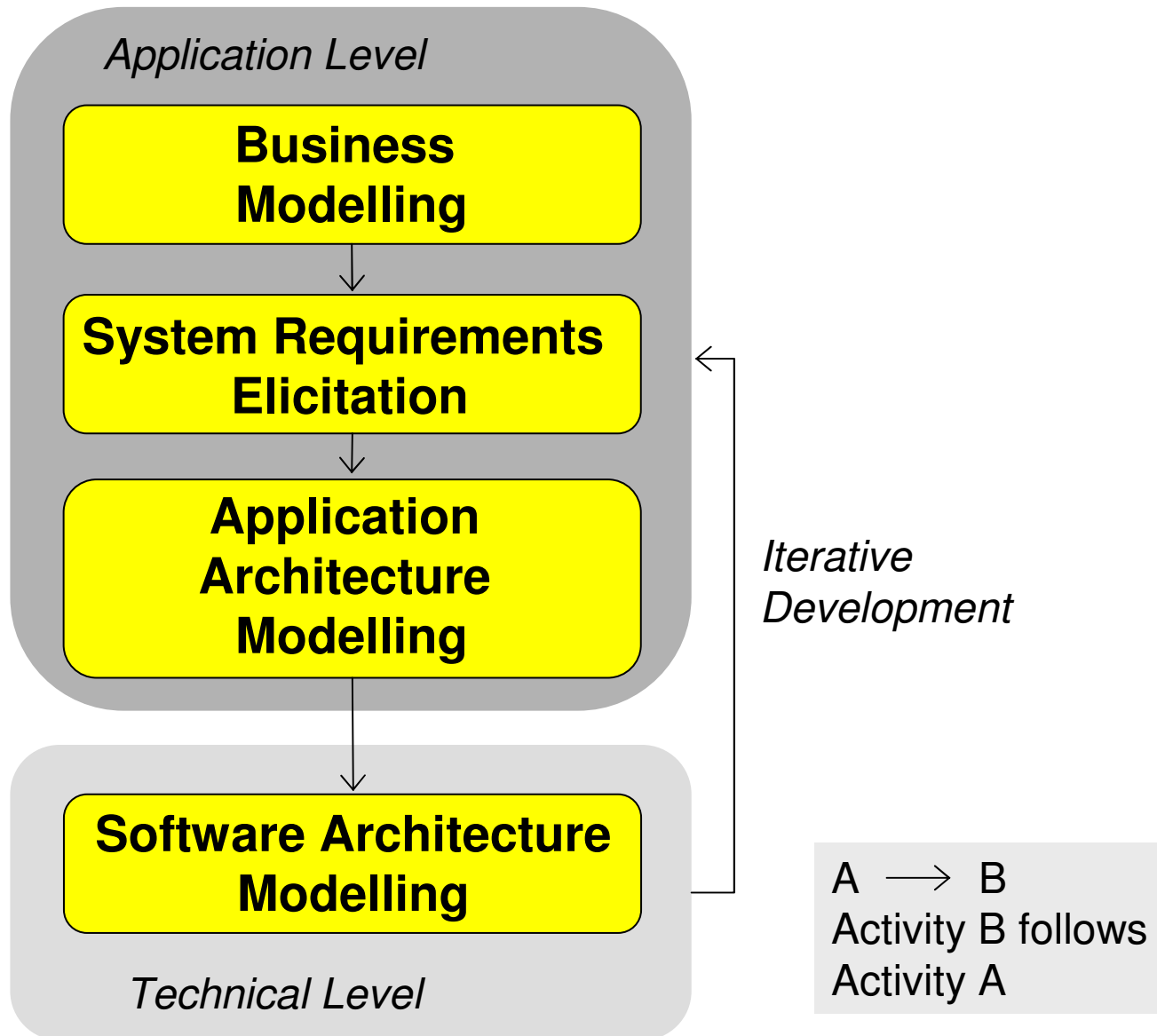
## *Technical Level*

**Software  
Architecture**

- ▶ System and network structure
- ▶ Definition of technical components

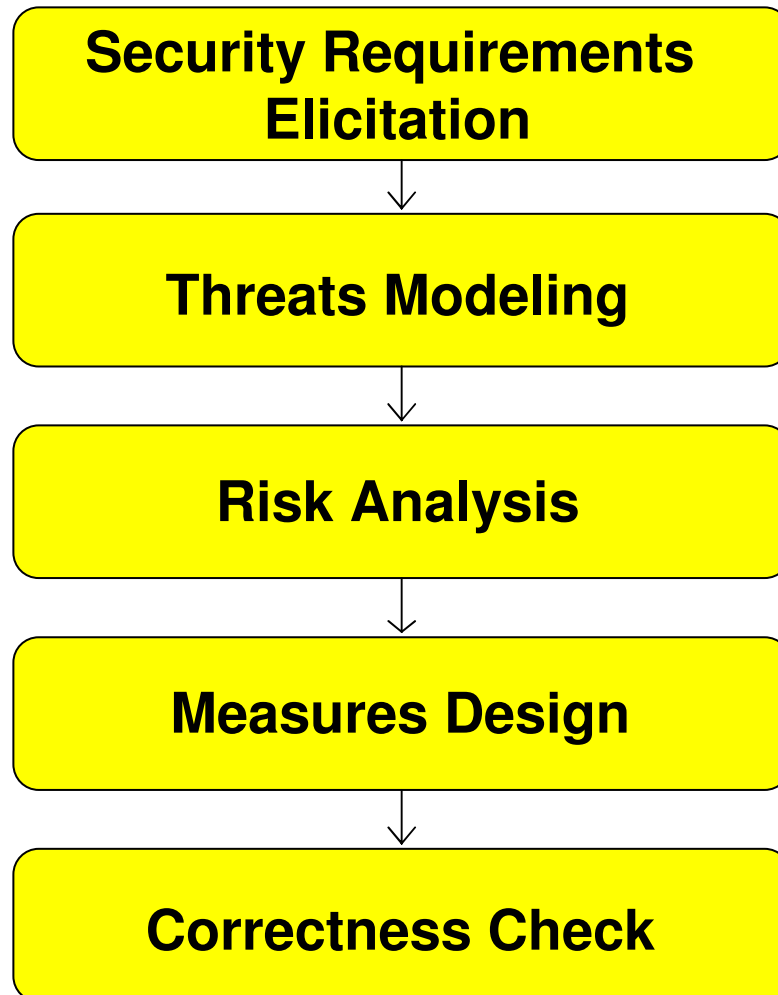


# The „Core Process“

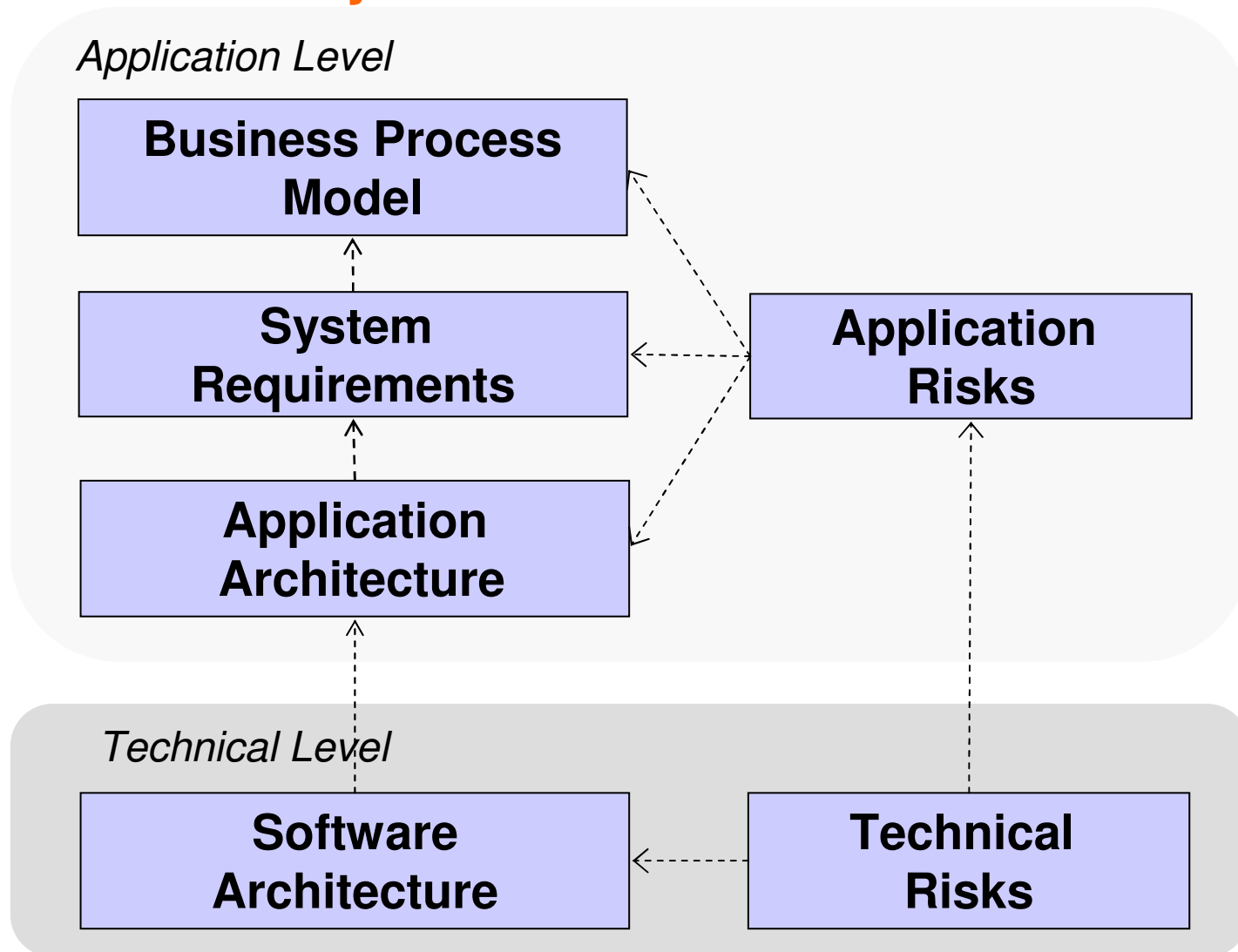


## 2. The „Security Process“

### Security Analysis as a Microprocess

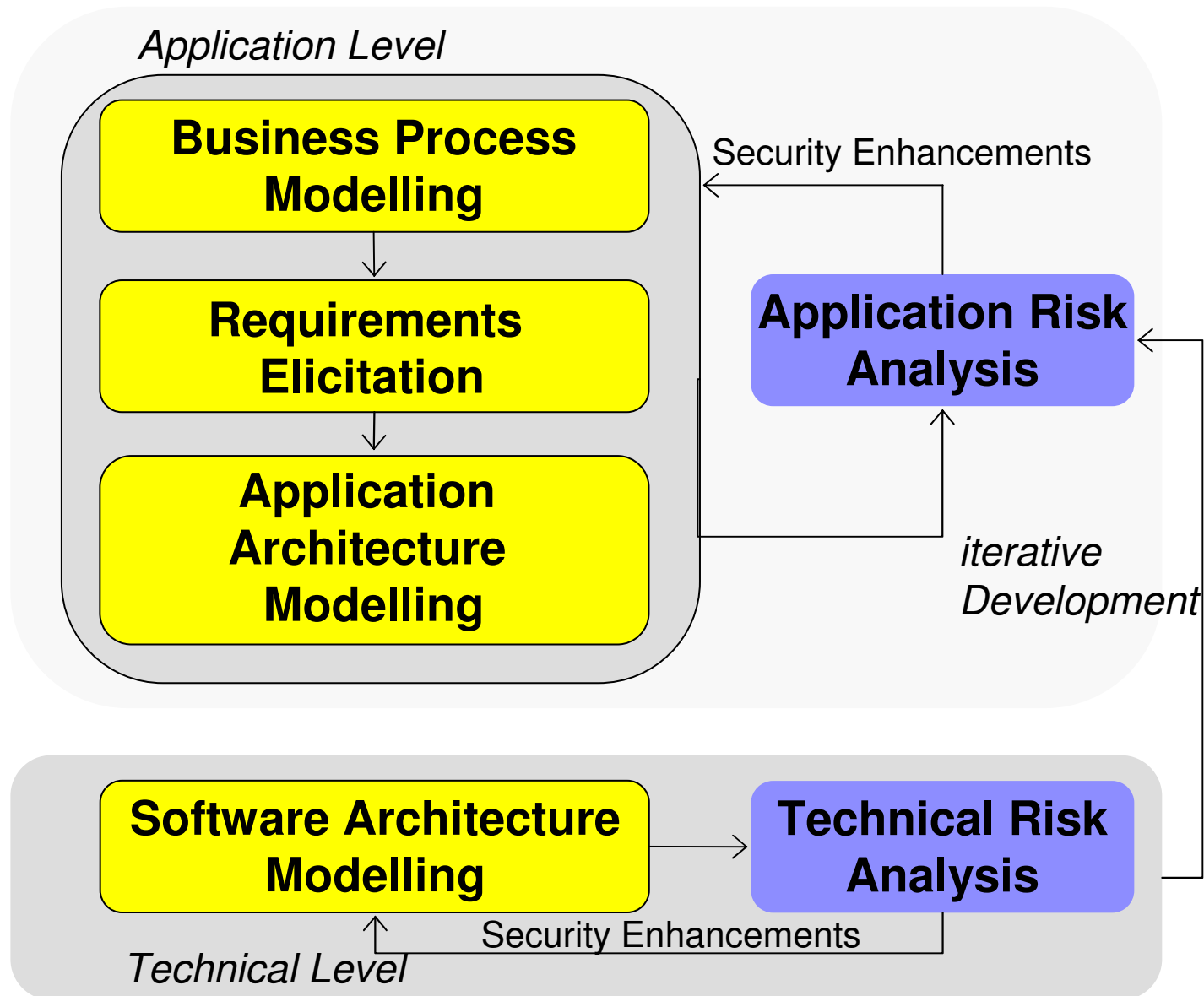


# The Artifacts of the Security Process





# The Security Process



### 3. Case Study - *TimeTool*

- Based on “*TimeTool*” - a software project which was realized at the University of Innsbruck.
- “*TimeTool*” is a software package supporting project controlling and administration.
- Working time is calculated in real time through a log in / log out timestamp.
- The system performs specific checks (on date and booked time) automatically and offers administrative and controlling features to the project manager (team worker management, statistical reports generation).
- 3-tier architecture on top of the J2EE platform.
- Accessed through a web front end.



# Security at the Business Process Level



## ► Confidentiality and Data Integrity

- » Which actors are granted read access to which object?
- » Which object flows are confidential?

## ► Authentication

- » Which activities require actor authentication in order to be performed?
- » Do actors need to be authenticated when exchanging objects?

## ► Non Repudiation

- » Which activities must not be denied?



## A Scenario – „Non Repudiation“

### ▶ Security Requirements

» A team member can not deny adjustment postings.

### ▶ Threat

» A team member could try to increase her billable time by posting a positive time adjustment.

» The project manager performs negative time adjustments in order to hide budget overruns.

### ▶ Risk

» Probability of occurrence is high and potential damage is substantial.

### ▶ Measures

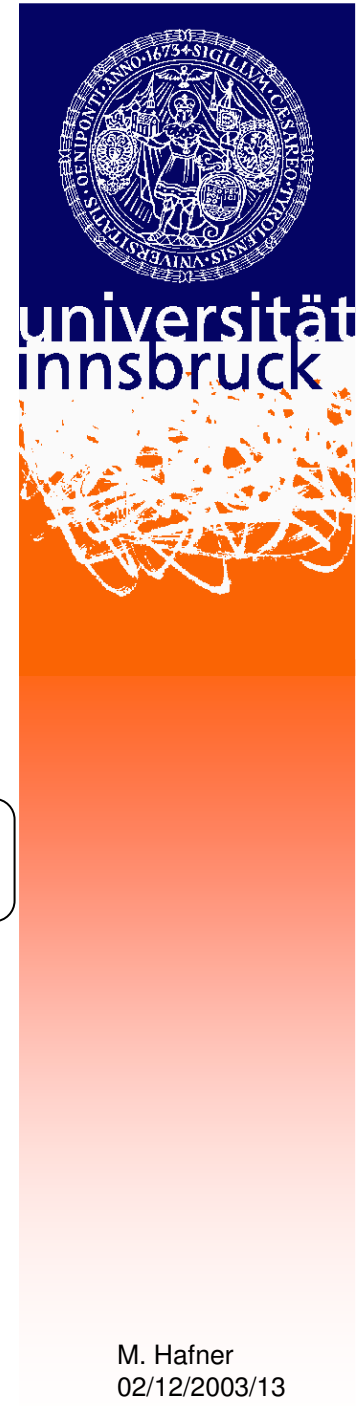
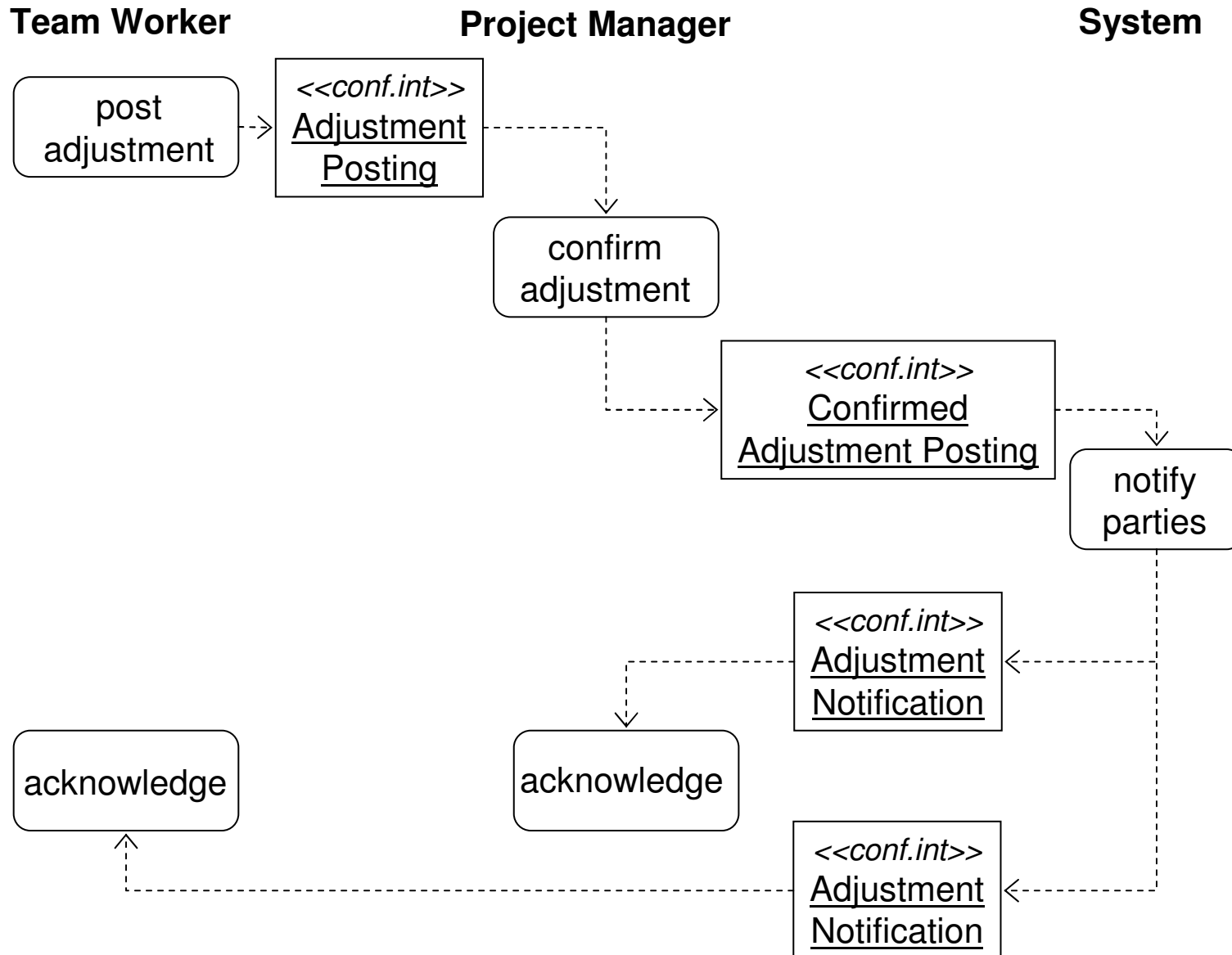
» Adjustment postings require justification and involved parties are automatically notified.

» Adjustment postings are documented and archived.



universität  
innsbruck

# Business Level: The Security Work Flow Model



# System Requirements Level: The Security Use-Case-Model

**use case** Adjustment Posting

... (previous textual description of the use case from Figure 4)

## **security**

A1 The adjustment posting is logged by the system

A2 The team worker has to be authenticated before starting the use case.

A3 Web browser and TimeTool have to be authenticated before the transaction starts.

A4 The system must guarantee the confidentiality and integrity of the input data.

A5 The use case must be available during extended working hours (6 a.m. to 22 a.m.)  
with a maximum of 2 continuous working days breakdown per month.

*Realisation of requirements from BP model (Non Rep., Authent.)*

*Requirements related to external systems*

*Availability*



# Application Architecture Level

## ► Main Activity:

» Design of appropriate security measures (security patterns).

## ► Steps:

» Definition of logical security components and their interfaces.

» Extension of the sequence diagrams describing the message flow of use case execution by security specific messages.

## ► Example Measures:

» Authentication procedures

» Access control and access rights management

» Error tracing (e.g. for access control)

» Introduction of security protocols (e.g. challenge – response)



universität  
innsbruck

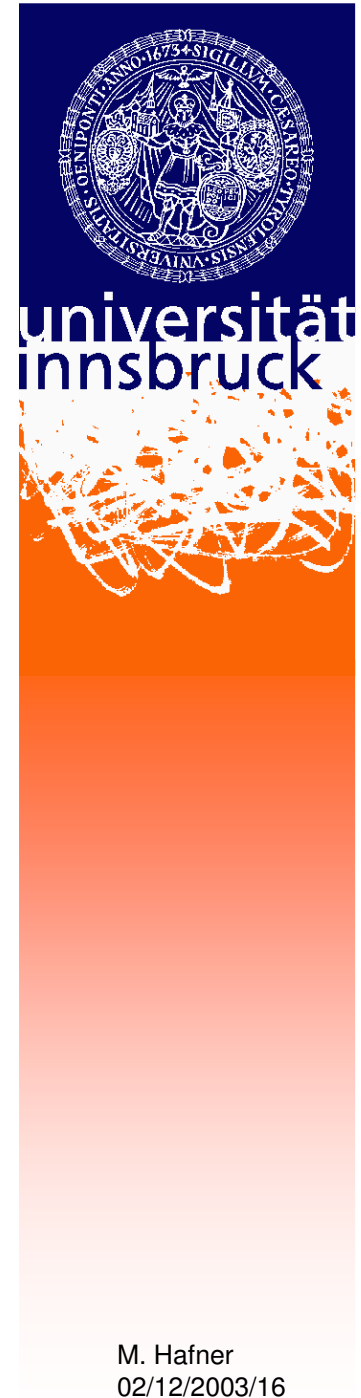
# Software Architecture

## ► Starting Points:

- » Design of basic architecture.
- » Deployment of logical components over network.

## ► The security enhancements to the software architecture are developed according to the *Micro Process Model*:

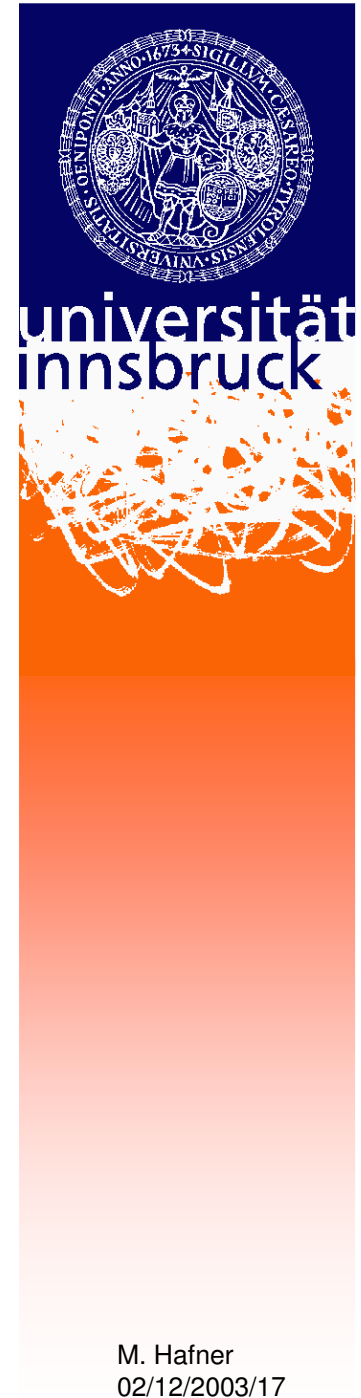
- » Mapping of security requirements stated in the *system requirements* and the *application architecture*.
- » Analysis of technical threats independently of the application domain (e.g. based on checklists).
- » Every technical threat is related with application level threats for a cross-checking.
- » Estimation of associated risk.
- » Design of *security architecture*.
- » Evaluation of compliance of measures to the requirements.





## 4. Summary

- ▶ **Sketch of a process model for security engineering**
  - » Within framework object oriented use case based modelling
  - » Integration of security analysis as a micro process
  - » Comprehensive view of the whole design process
- ▶ **Current project activities**
  - » Management of IT security risks
  - » Security in cross organizational workflows
  - » Modelling and documentation of security architectures
  - » Formal Specification of access rights





universität  
innsbruck



**Thank you for your attention!**