universitätbonn

b-it

RWTH AACHEN
RHEINISCH-WESTFÄLISCHE TECHNISCHE HOCHSCHULE AACHEN

Practical Course eXtreme Programming

B-IT/IPEC

Summer School 2007

Organized by
Prof. Dr. Armin B. Cremers, Holger Mügge, Pascal Bihler, Mark Schmatz

Bonn-Aachen International Center for Information Technology

b-it

Established in fall 2002 by:

universitätbonn
RHEINISCHE FRIEDRICH-WILHELMS-UNIVERSITÄT

RWTH AACHEN
RHEINISCH-WESTFÄLISCHE TECHNISCHE HOCHSCHULE AACHEN

Fraunhofer IZB Institutszentrum Schloss Birlinghoven

Fachhochschule Bonn-Rhein-Sieg

Fraunhofer AIS Institut Autonome Intelligente Systeme

International Program of Excellence (IPEC)

„The *International Program of Excellence in Computer Science (IPEC)* at the B-IT offers, mainly in the time between terms, compact teaching units on the highest level. This results in a speed-up of studying and in a simultaneous increase of quality.“

universitätbonn

eXtreme Programming

An Introduction

universitätbonn

What is Extreme Programming?

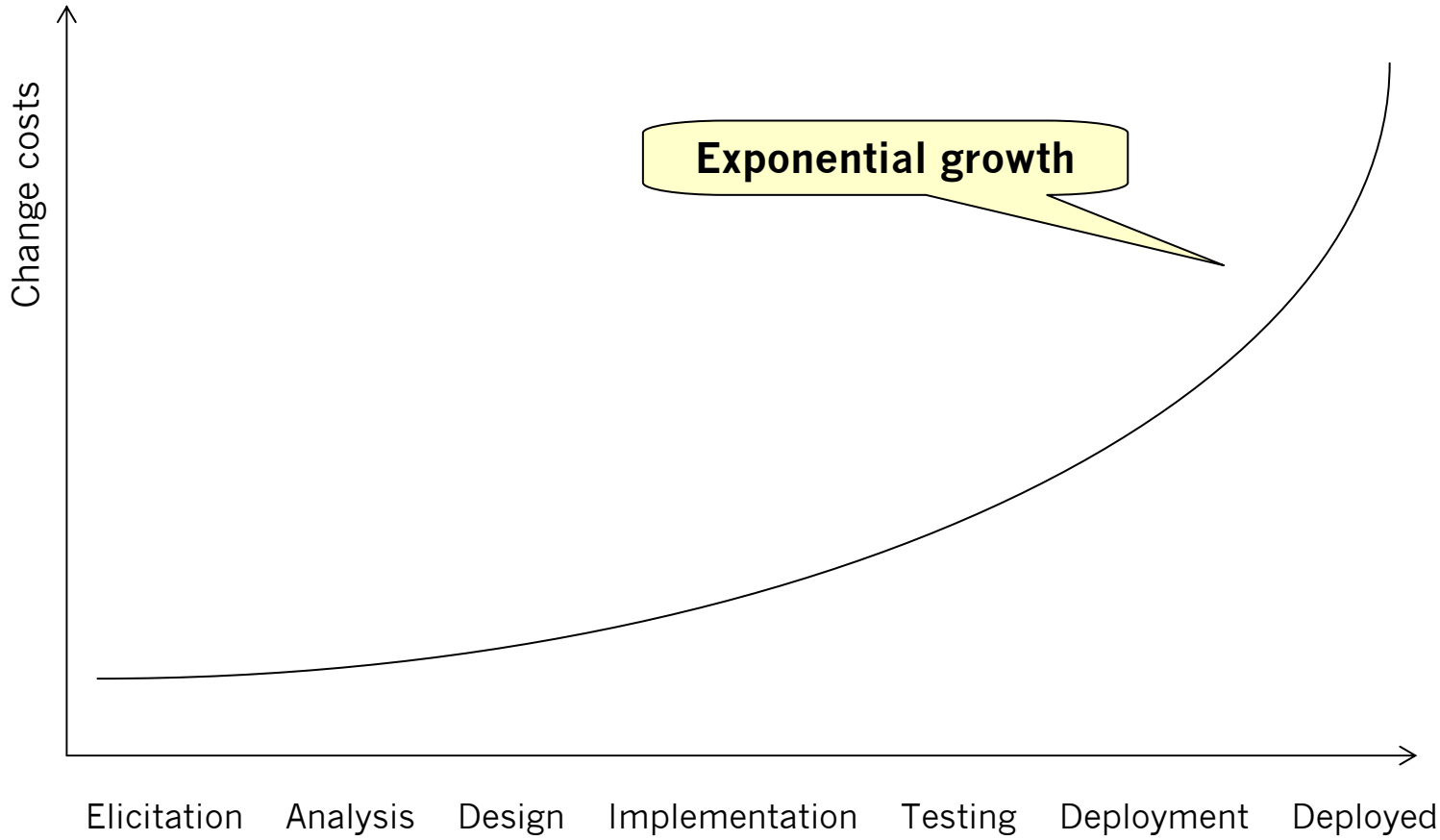
- *XP is a* ...
  - lightweight,
  - efficient,
  - low-risk,
  - flexible,
  - predictable,
  - scientific and
  - fun

  ... *way to develop software.*

- *Kent Beck, eXtreme Programming eXplained, Addison Wesley 1999*

How expensive is change?

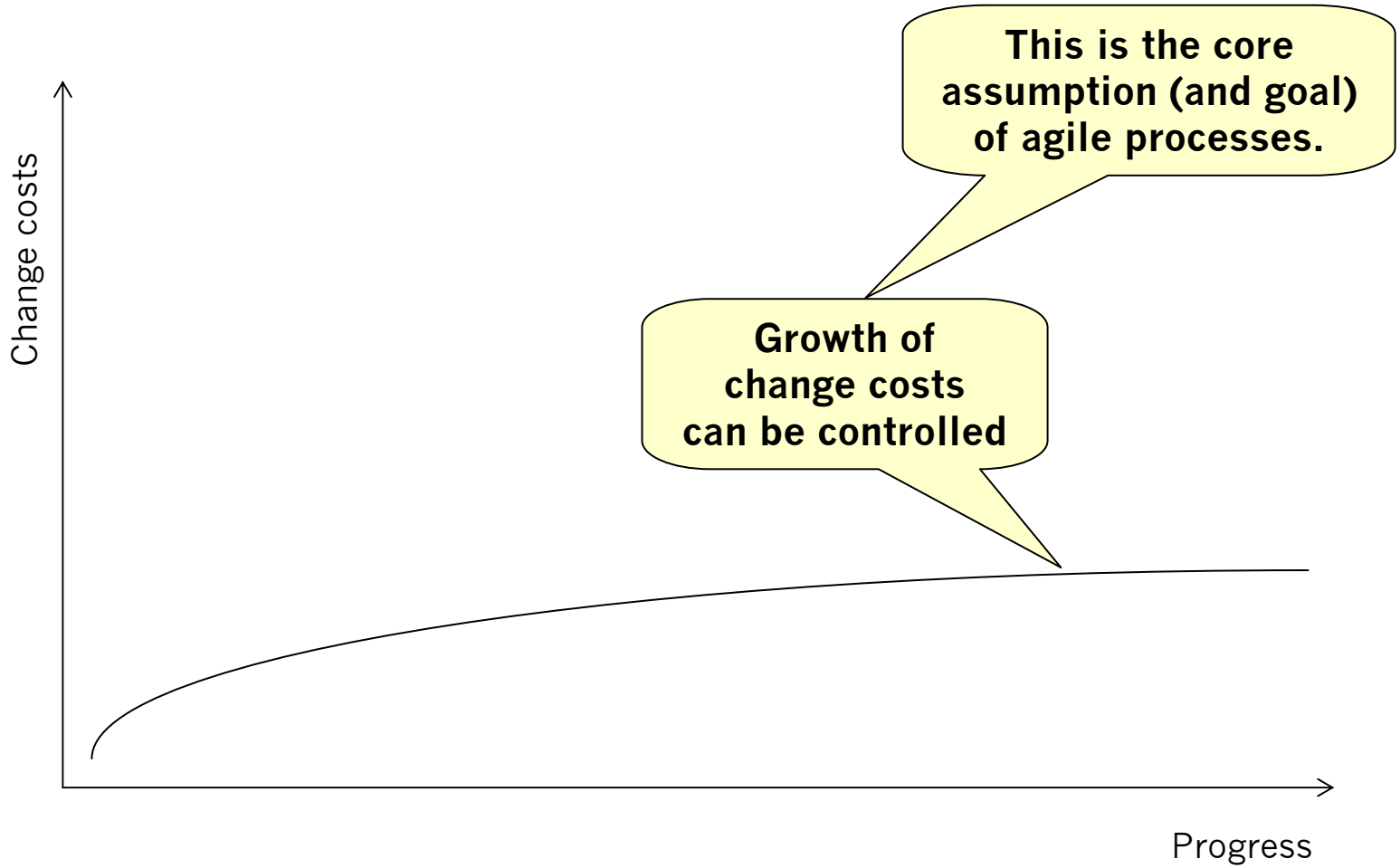
- Traditional view / Experience



universitätbonn

How expensive is change?

- Agile view / Experience



universitätbonn

How can this be achieved? Values!

- **Agile Manifesto** ought to value… http://agilemanifesto.org

  Individuals and interactions *over* processes and tools
  Working software *over* comprehensive documentation
  Customer collaboration *over* contract negotiation
  Responding to change *over* following a plan

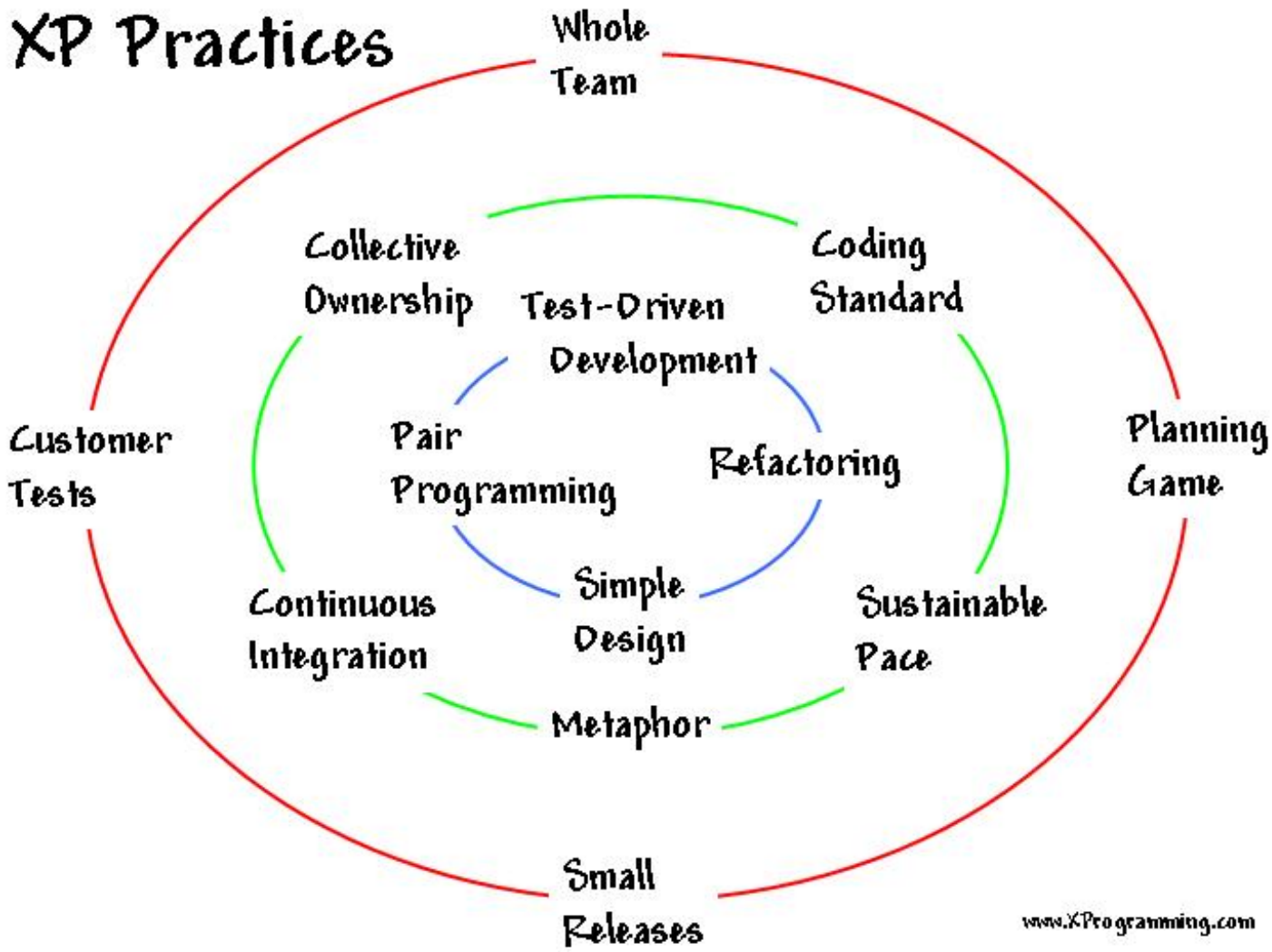
  There is value in the right side items, but we value the left ones more.

- Extreme Programming Values
  - Communication
  - Simplicity
  - Feedback
  - Courage
  - Respect

http://en.wikipedia.org/wiki/Extreme_Programming#XP_values

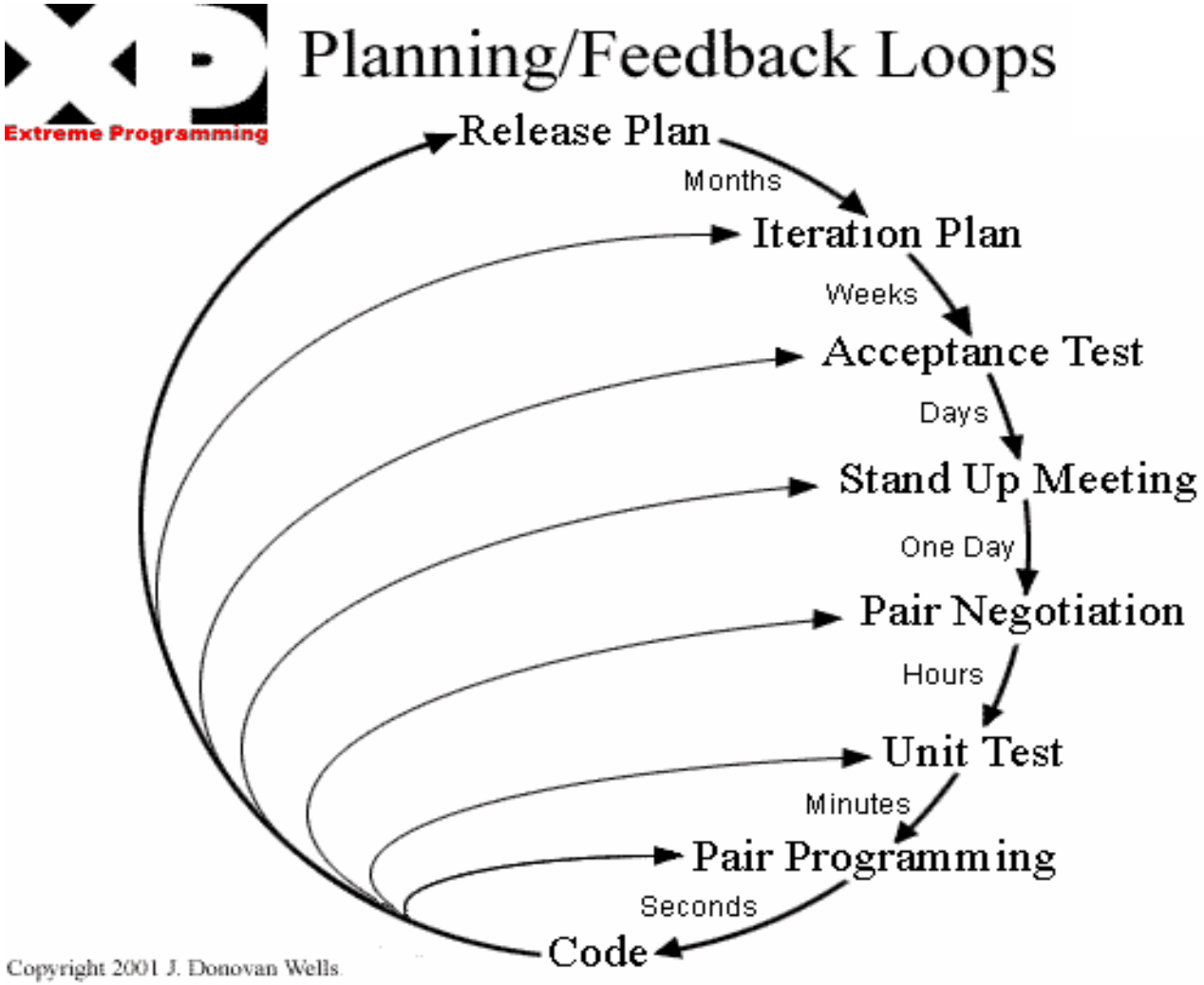
universitätbonn

How can this be achieved? XP Practices!



universitätbonn

Rapid feedback is the secret of success



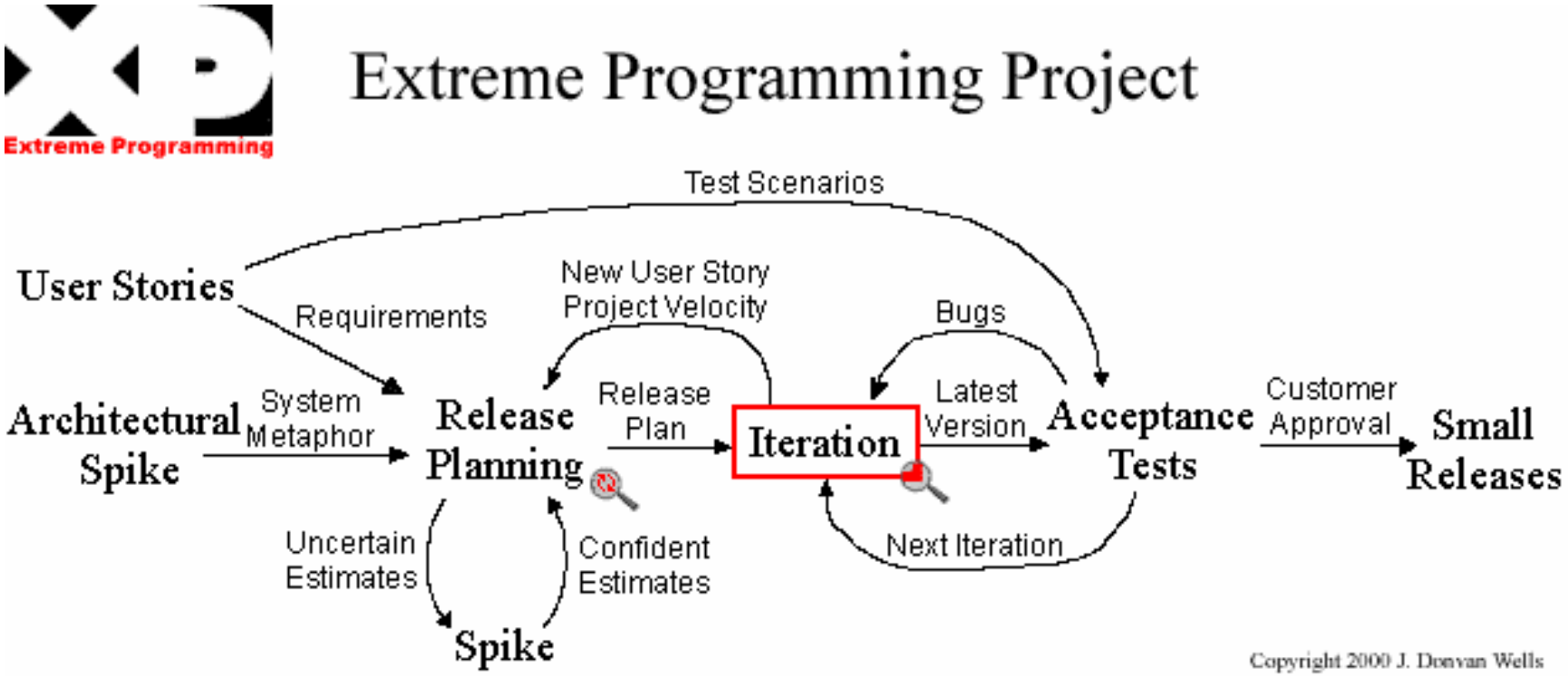
XP is like driving.

"Driving is not about getting the car going in the right direction.

Driving is about **constantly paying attention,** *making a* **little correction** *this way, a* **little correction** *that way."*

(Kent Beck)

Remain responsive to change at any time



eXtreme Programming

Experiences from our courses

universitätbonn

Things that worked! (1/3)

- Refactoring exercise as acceptance test
  - Provided a realistic picture of the required skills
  - Improved OO knowledge
    - Repair a **broken JUnit test**.
    - Refactoring 1: **Break dependency cycle** by introduction of observer pattern.
    - Refactoring 2: **Eliminate code duplication** and parallel inheritance hierarchy by introduction of state pattern.

- Each participants makes himself an **expert of a special area** before the beginning of the course
  - Necessary because very much knowledge needed
  - Makes the value of team communication obvious

- **Wiki** as a common knowledge base

Things that worked! (2/3)

- **First 3 days**: Development of a trivial application. Meanwhile teach...
  - XP basics, Test First, Pair Programming
- **Four iterations** (each at 5 days)
  - Structure:
    - planning game
    - Implementation
    - presentation + acceptance test
- **Planning poker** (see later)
- **XP-Game** to explain the planning Game
  - Clarifies responsibilities of customers and developers
  - Demonstrates the value of realistic estimations.
  - Sometimes a little bit time consuming. But: Fun!

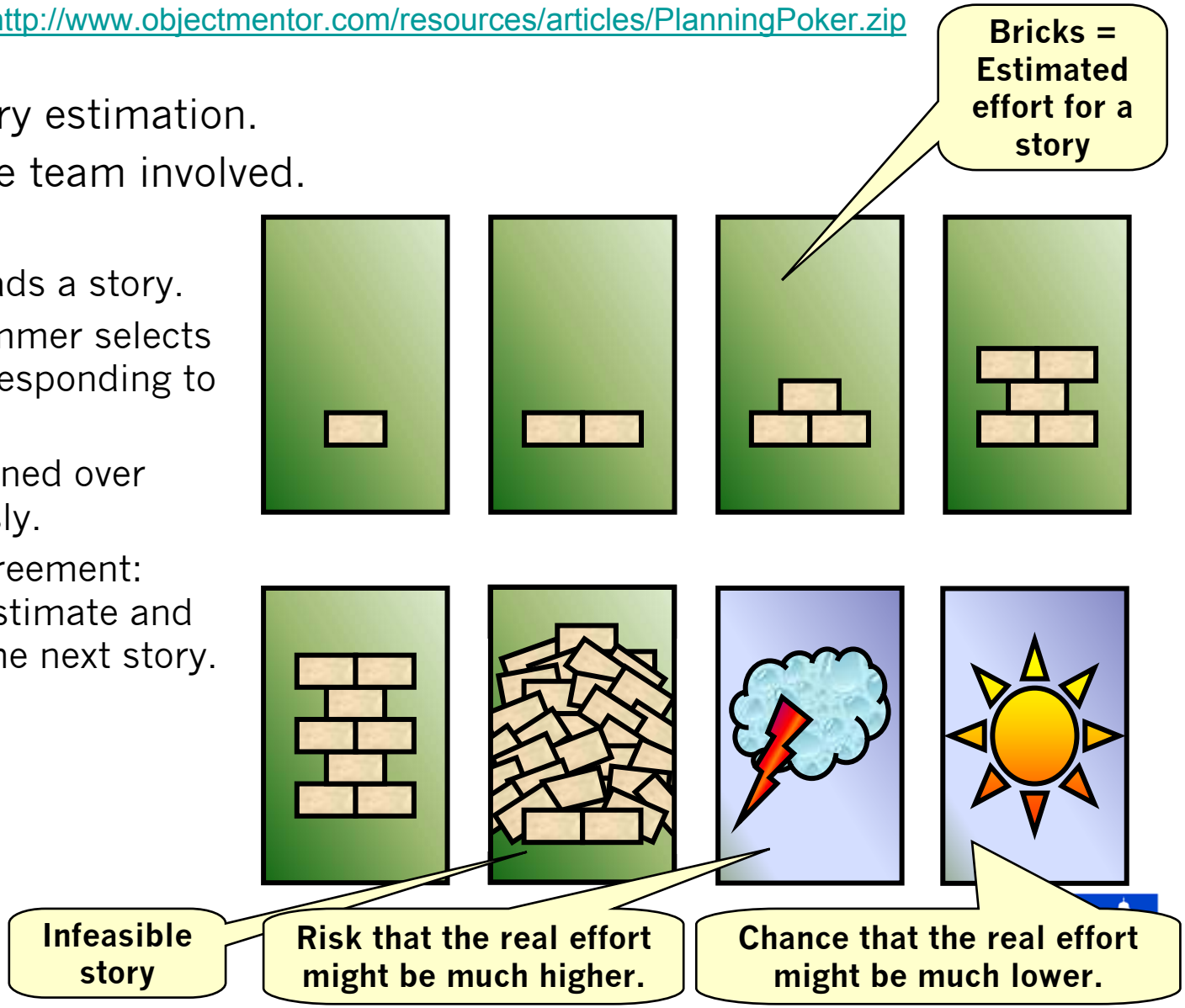
Things that worked! (3/3)

- **Pair Programming**
- Daily **stand up meeting**
  - Everybody answers questions like:
    - *What did I do yesterday?*
    - *What obstacles do I have?*
    - *What am I going to do today?*
    - *What else should the team know about?*
  - No one gets stuck into problems others could solve.
  - Builds team spirit.
- Regular **Retrospections**
  - THE key to excellence by rapid adaptation
  - Example: Burn-Down Charts (see later)
- **Non-conflicting roles** (Keep *What* and *How* separate)
  - customer ≠ team leader ≠ expert

Planning Poker (adapted from James W. Grenning, Object Mentor, 2002)

http://www.objectmentor.com/resources/articles/PlanningPoker.zip

- Accelerates story estimation.
- Keeps the whole team involved.
- Mechanics:
  - Customer reads a story.
  - Each programmer selects the card corresponding to his estimate.
  - Cards are turned over simultaneously.
  - In case of agreement: Record the estimate and move on to the next story.



The XP-Game (adapted from Belgian XP/Agile User Group)

Story descriptions. Business value printed on each card.

Clear responsibilites
(„We are developers“)

Funny „stories“ to „implement“

Already used planning poker like in the real planning game.

http://www.xp.be/xpgame.html

universitätbonn

Simple Design, Testing

- **Simple Design**
  - (+) CRC cards nice tool for fast architectural sketches
  - (–) Difficult to get everybody involved
  - (+) Explore protocols by role playing simulations
  - (–) Hard to explain the difference between simple and quick premature design

- **Testing**
  - Recommended but rarely practiced by the students
  - Requires severe discipline
  - GUI testing is really hard
  - Round trip testing even harder (Java ⇨ Compiler ⇨ JTransformer ⇨ Prolog ⇨ Predicate Evaluation)
  - Untested code forms legacy code after the course

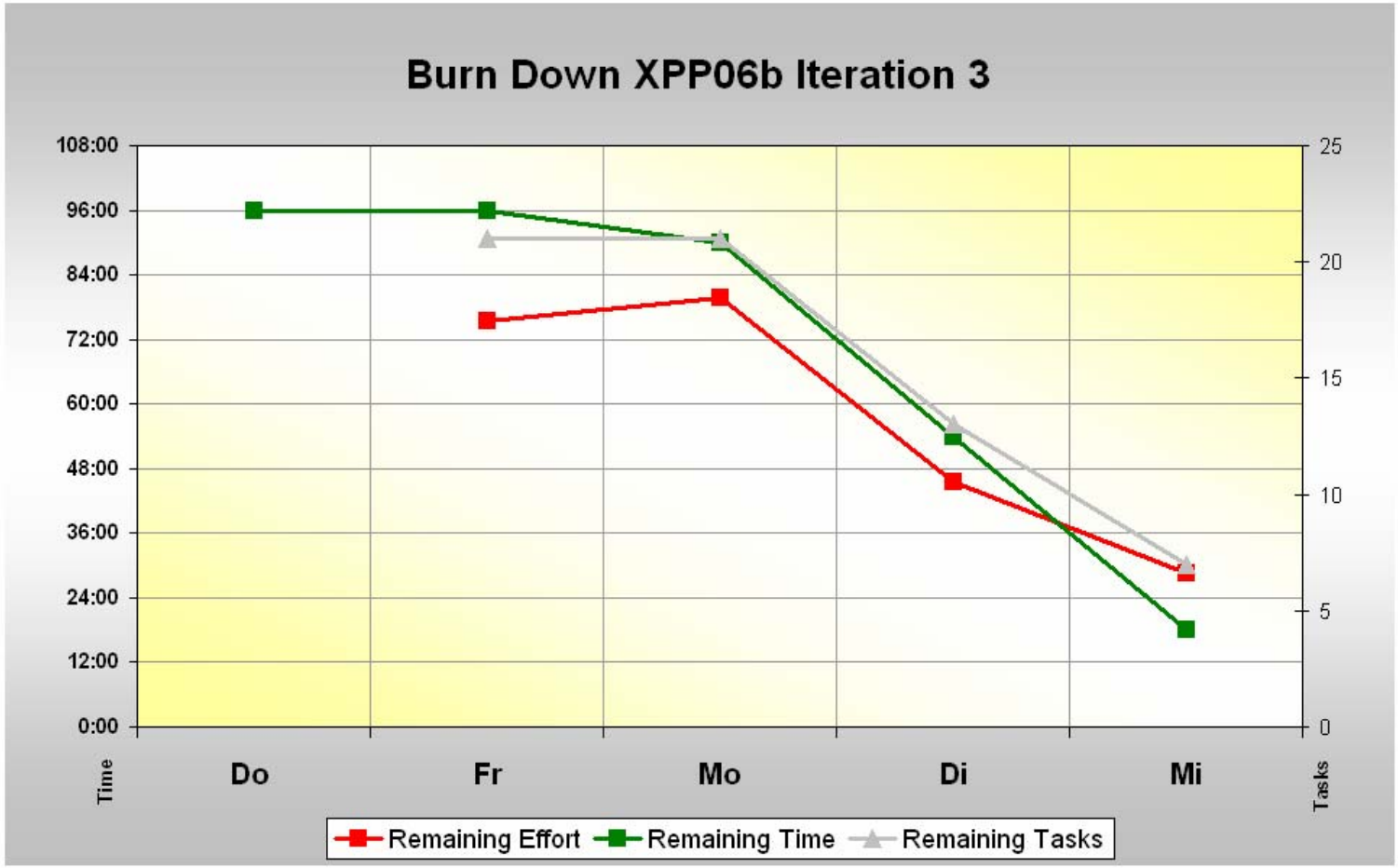
Tracking

- Essential to estimate capacity of the next iteration

- Motivation for tracking (estimation + consequent logging of time spent) is hard to find:
  - The XP coach really has to care about this
  - Fast feedback about the quality of the estimates might help (Exercised once)
  - Hard to avoid to pessimistic estimates
  - Burn down charts (see SCRUM) might be a better way because they constantly visualize the remaining effort

- Virtual unit for effort (bricks, see above)
  - facilitates relative estimates
  - are not always taken serious

Burn-Down Charts



universitätbonn

Reality Check

- The “ideal world” of our courses
  - Our customer is much more friendly than real customers usually are.
  - Employees have much more interfering responsibilities than our students have for the time of the course.

- Problems that also occur in the “real world”
  - Development teams are seldom self-organizing
    - But those, that are, are the best
  - Building a reliable testbed is really hard

b-it: Excellent working environment

universitätbonn

Our practical courses

universitätbonn

What we are offering the students...

- Good supervision
  - 2 (+4) research associate for 12 students
  - ... 8 hours a day!

- Ideal working environment
  - Our own office
  - Up-to-date technical equipment (computer, beamer)
  - Wikis, Eclipse, UMPCs, ...

- An interesting and realistic project
  - Developing process is an essential parts of research projects
  - Industry partner ensures quality of product

- A certificate after four and a half weeks
  - ECTS Credit Points 10

universitätbonn

The „products“ of the practical courses

Rich Client Applications on Mobile Device

- 2006b: Context Sensitive Mobile Application (CSI Navigator)
- 2005b: Context Sensitive Mobile Application (CSI PimPro)

Plug-Ins for the Java Development Platform Eclipse

- 2005a: Visual Tool Support for Refactoring to Pattern (Cultivate, PatchWork)
- 2004b: Program Analysis by Logic Meta Programming (JTransformer, Cultivate)
- 2004a2: Tool Support for Pattern Management (PatchWork)
- 2004a1: Synchronized Logic Representation of Java Code (JTransformer)
- 2003b: Improved Editor for Conditional Transformations (ConTraCT) [based on the result of two earlier practical courses]

The CSI project

Towards
Context Sensitive Intelligence

What is the team you‘re working with?

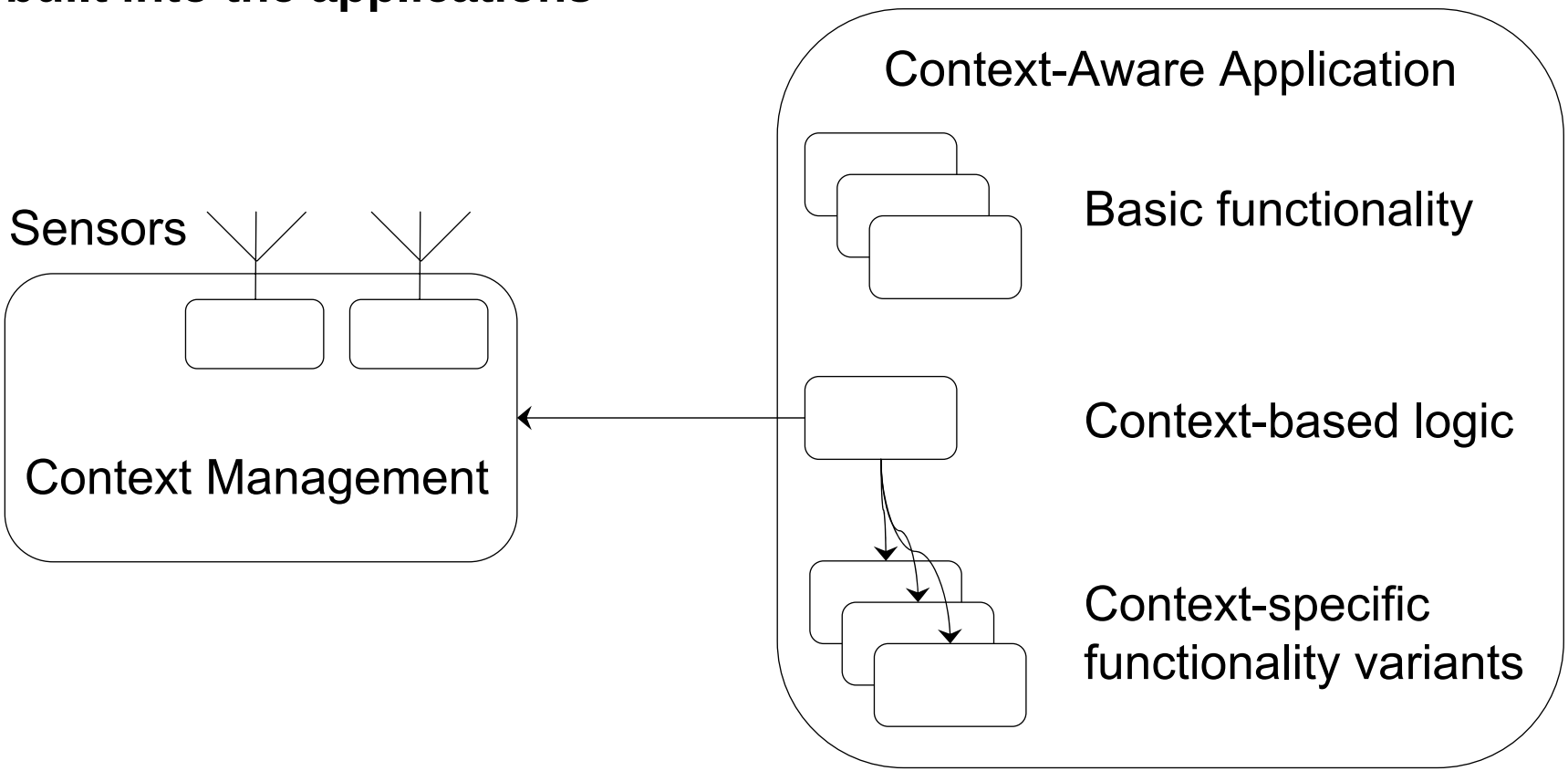
- The Context Sensitive Intelligence Project
  - funded by Deutsche Telekom Laboratories
  - focuses on
    - context-sensitive adaptations
    - the developer view
  - aims for
    - minimal anticipation
    - automatic adaptation

Deutsche Telekom

universitätbonn

Status Quo of Context-Aware Systems (1)

Context-Awareness is preplanned and built into the applications



Status Quo of Context-Aware Systems (2)

Context-Awareness today is characterized by:

- Planned and built in context-sensitivity (what context to take into account)
- Predefined context-based logic (when use which functional variant)

Recent developments allow for:

- Dynamic Sensor Integration (e.g. use external sensors when available)
- Automatic Adaptivity (poll context and determine adaptation automatically)

Mobile Computing needs more Flexibility

Our View:

- Mobile Devices are capable of many more context-specific user support
- Web-Based Services will soon be ubiquitously available
- Most situations and functional requirements can not be anticipated and preplanned in detail

Therefore:

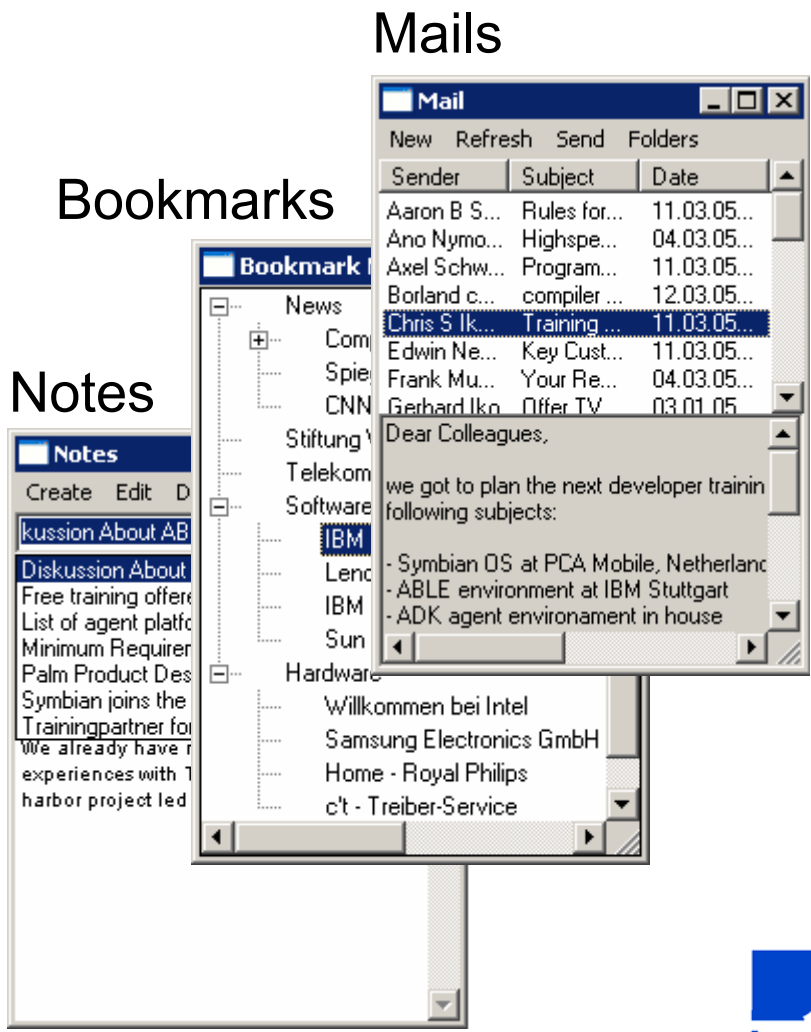
- Application should support unanticipated adaptivity
- Dynamic integration of new functionality is needed

A Concrete Scenario

Business user visiting
a trade fair:

- Many different meetings
- Tight schedule
- Has a lot of documents
- Documents are complexly related to meetings

➔ Needs prompt access to relevant documents



universitätbonn

Solution – Personal Information Prompter

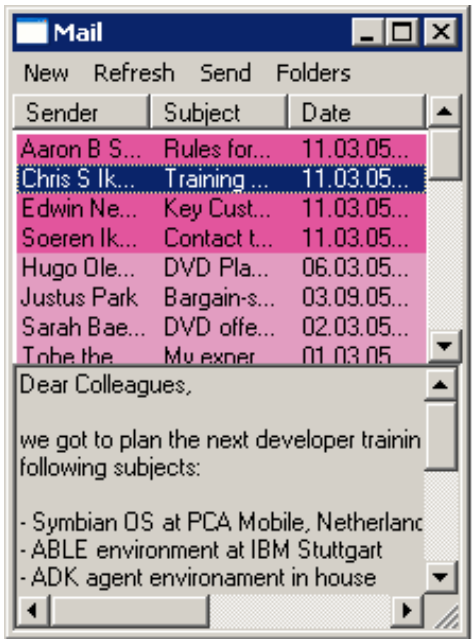
- Idea:
  - Highlight or filter relevant data according to the stand the user is currently located close to.

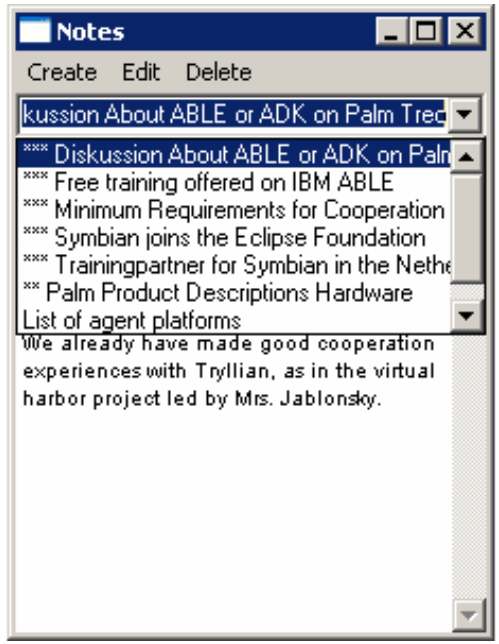
- Means:
  - Detect current user location
  - Find closest stand on fair map
  - Stands described by keyword lists
  - Documents can be indexed

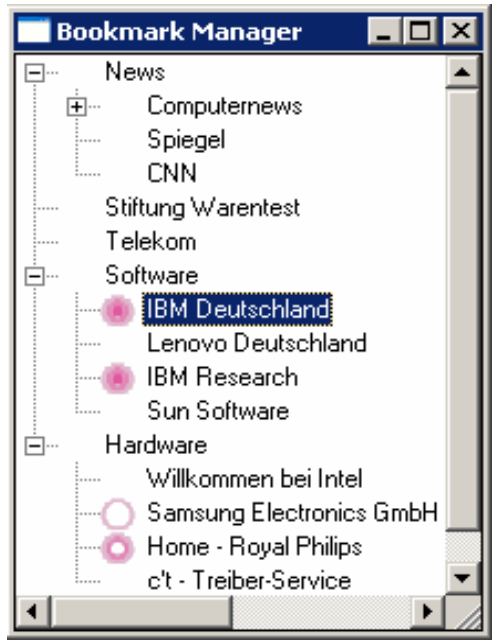
  ➔ Document relevance for stand can be assessed

Solution – Adapted Applications

- Adaptation cross-cuts multiple applications
- Adequate for different base applications







universitätbonn

Main Challenge

This scenario

- is useful for the user
- can not be predicted

The main challenge:

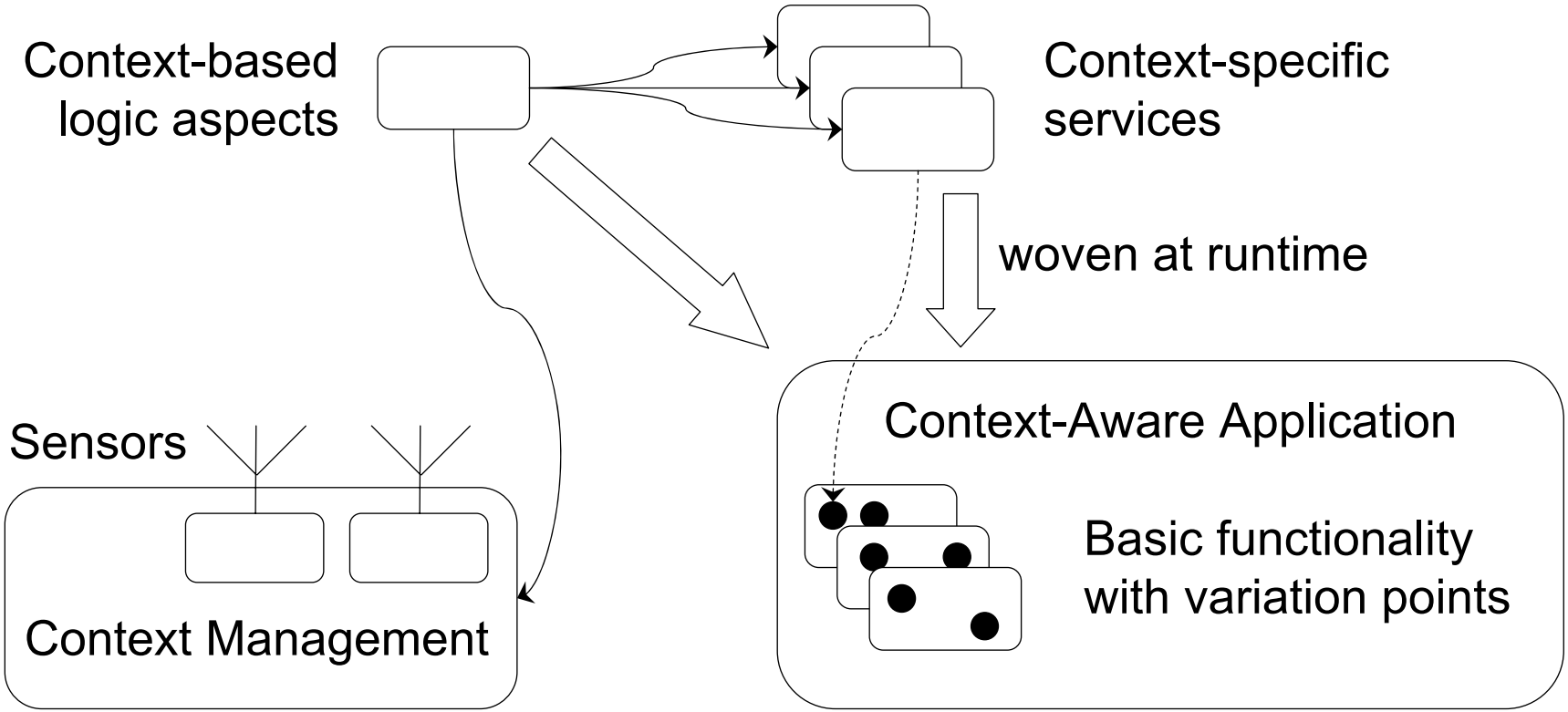
- developers of mail client, notes tool etc. are not able to plan for it

The Goal of the CSI Project:

- allow for unplanned adaptations
- apply up-to-date methods

The CSI Approach

Context-Awareness and context-specific functionality is woven into the applications at runtime when appropriate



universitätbonn

Requirements & Means

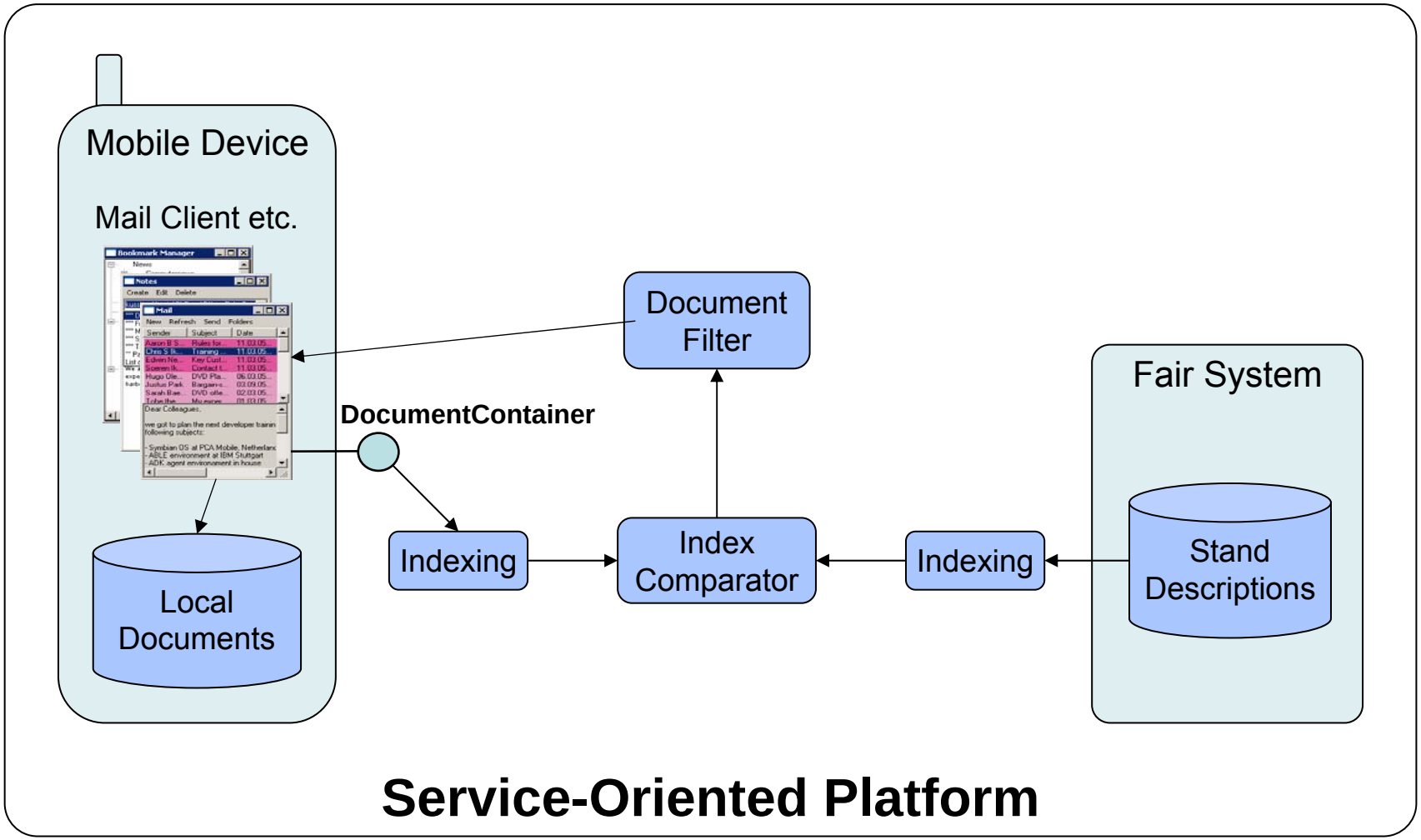
The CSI Approach presupposes:

- A Service-Oriented Architecture
- Abstract Preparation in terms of Semantic Annotations

The CSI Approach utilizes:

- Runtime Aspect-Orientation
- Access to Context Data within Aspects
- Realization of Virtual Services from Annotated Semantics
- Detection and Context-Sensitive Assessment of Web-based Services
- Runtime Weaving and Reconfiguration of Services and Compositions

Realization for our Example



universitätbonn

Context-Awareness

Characteristics

- continuously at runtime
- sensor based
- planned context-sensitive behavior

CSI supports

- Context-Management based on Logic
- Including Ontological descriptions (OWL)
- dynamic Sensor-Integration

Adaptivity

Characteristics

- seldom occurring
- at runtime
- event-based
- changes user software / configuration

CSI supports

- Plain OO Coding
- Service-Oriented Architectures

Anticipation

	Anticipated	**Unanticipated**
Context-Awareness	Self-contained context-aware applications	Dynamic sensor integration
Adaptivity	Plug-Ins, Planned reconfiguration	Using standard applications for new purposes

Refactoring to Adaptive Design

Features:

- no preconditions
- helps introducing good design
- in particular to prepare flexibility

Drawbacks:

- adaptivity relies in detail on common implementations (Java Interfaces)
- adaptation specificities need to be implemented manually

Context-aware Service Aspects

Features:

- implementation effort for adaptation drastically reduced (service management)
- context-awareness enabled
- anticipation reduced to service level

Drawbacks:

- applications must be tailored appropriately in advance
- potential adaptivity must be foreseen and considered in architecture

Annotated Semantic Structures

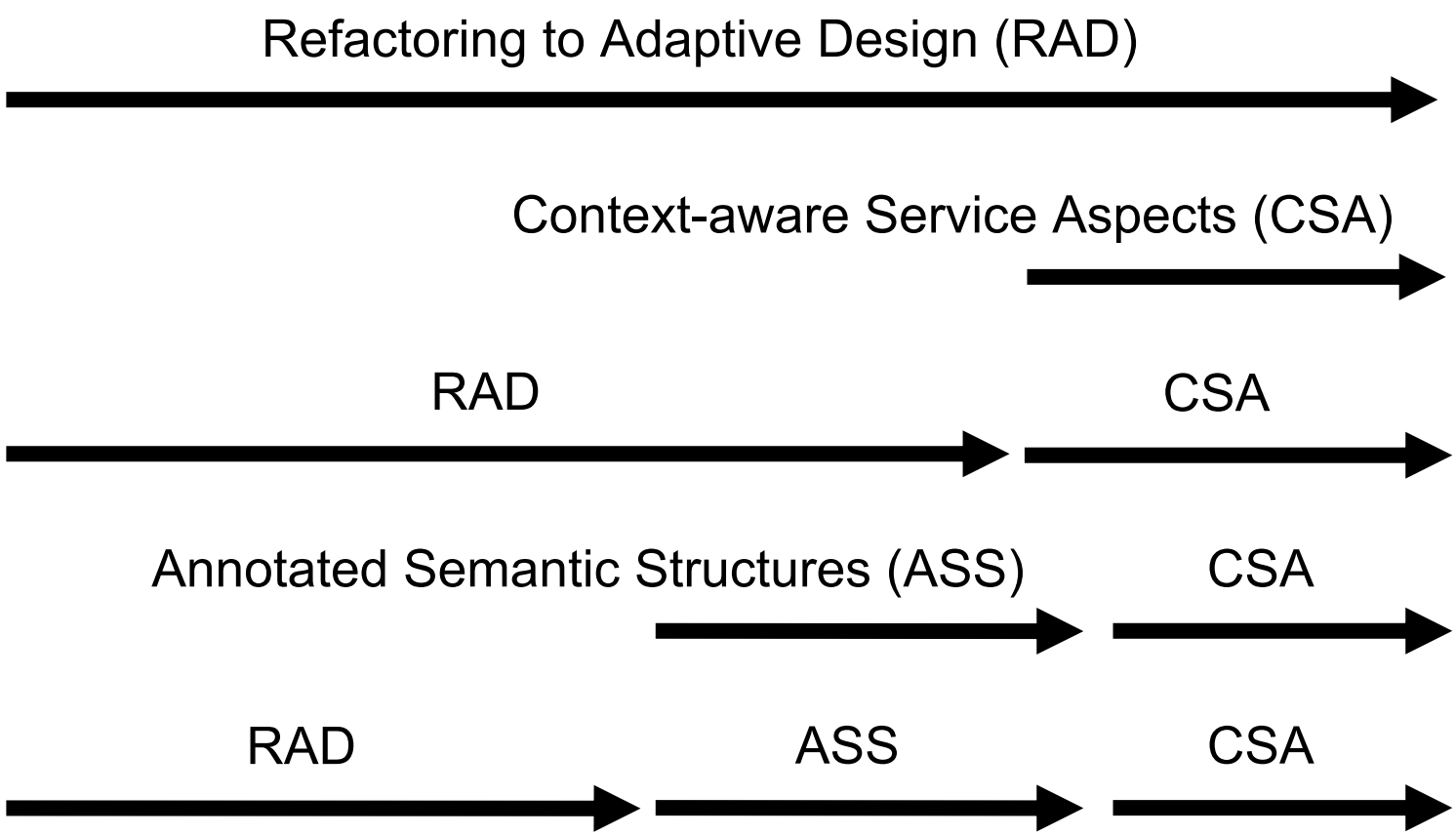
Features:

- reusable semantic concepts predefined
- declarative style
- consistency checks and further editing support
- semi-automated converting to services

Drawbacks:

- extensive description instead of anticipation

Five ways to Adaptivity



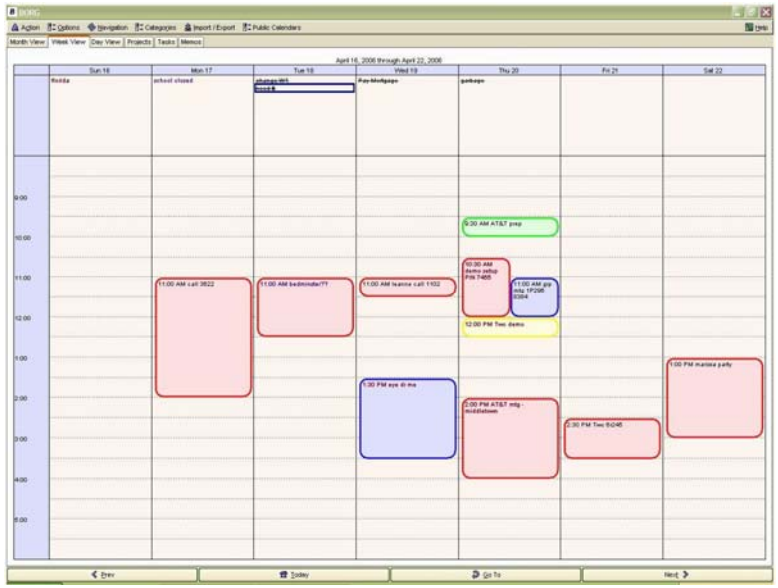
This year‘s scenario

Adaptive Mobile Gaming

universitätbonn

What are we developing this time?

- Customer has developed tools, to condition existing applications for adaptation
  - Integrated into Eclipse
  - Focusing on Design Patterns, Services, Aspects, and Annotations
- Customer wants to evaluate these tools



- Example:
  - Existing calendar application needs to be made adaptive to support other usage scenarios.

universitätbonn

The scenario

- Use case: **Adaptive Mobile Gaming**
- Mobile Games:
  - PDA/UMPC based
  - Use context data
  - Augment reality

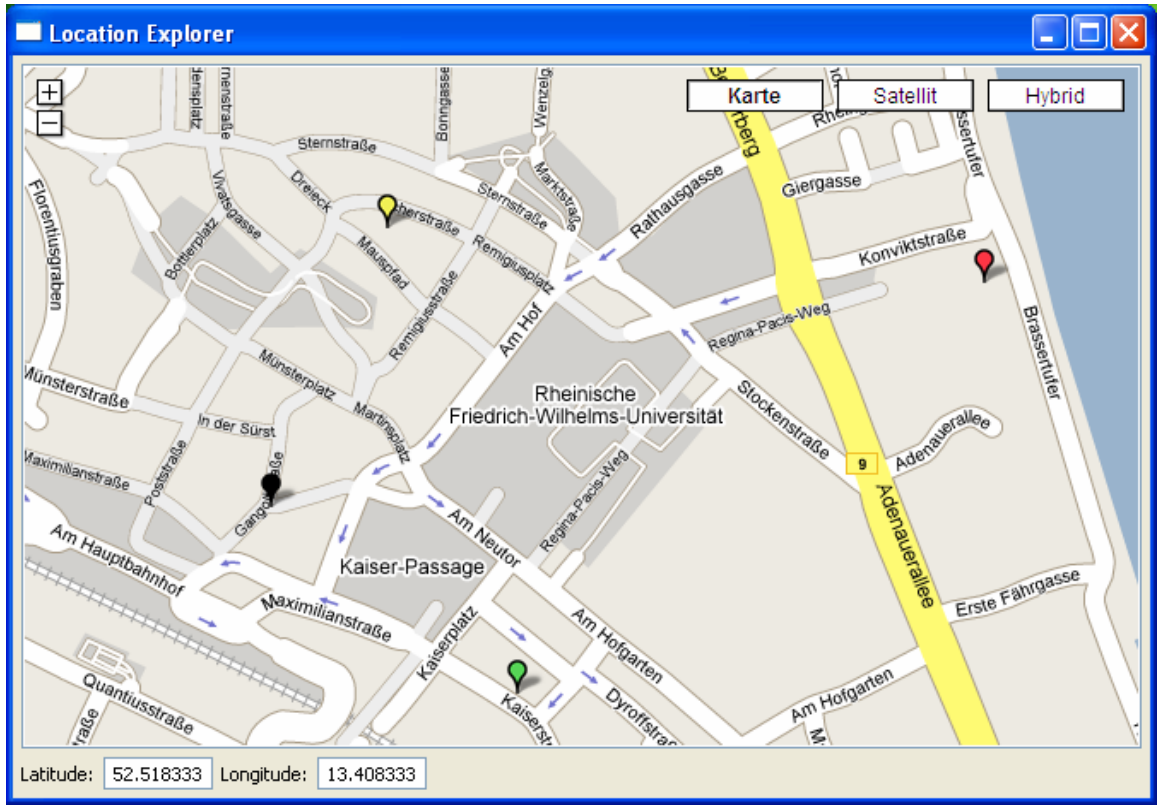
supported by:

Ravensburger

universitätbonn

Move Legacy code to adaptation

- A simple version of the game alread exists
- It needs to be made adaptable



universitätbonn

Studious and productive collaboration

universitätbonn

Satisfied students, associates and prof.