VSR | EDU

CORONA
EMERGENCY
LECTURE

# Cloud & Web Anwendungen

**Prof. Dr.-Ing. Martin Gaedke**

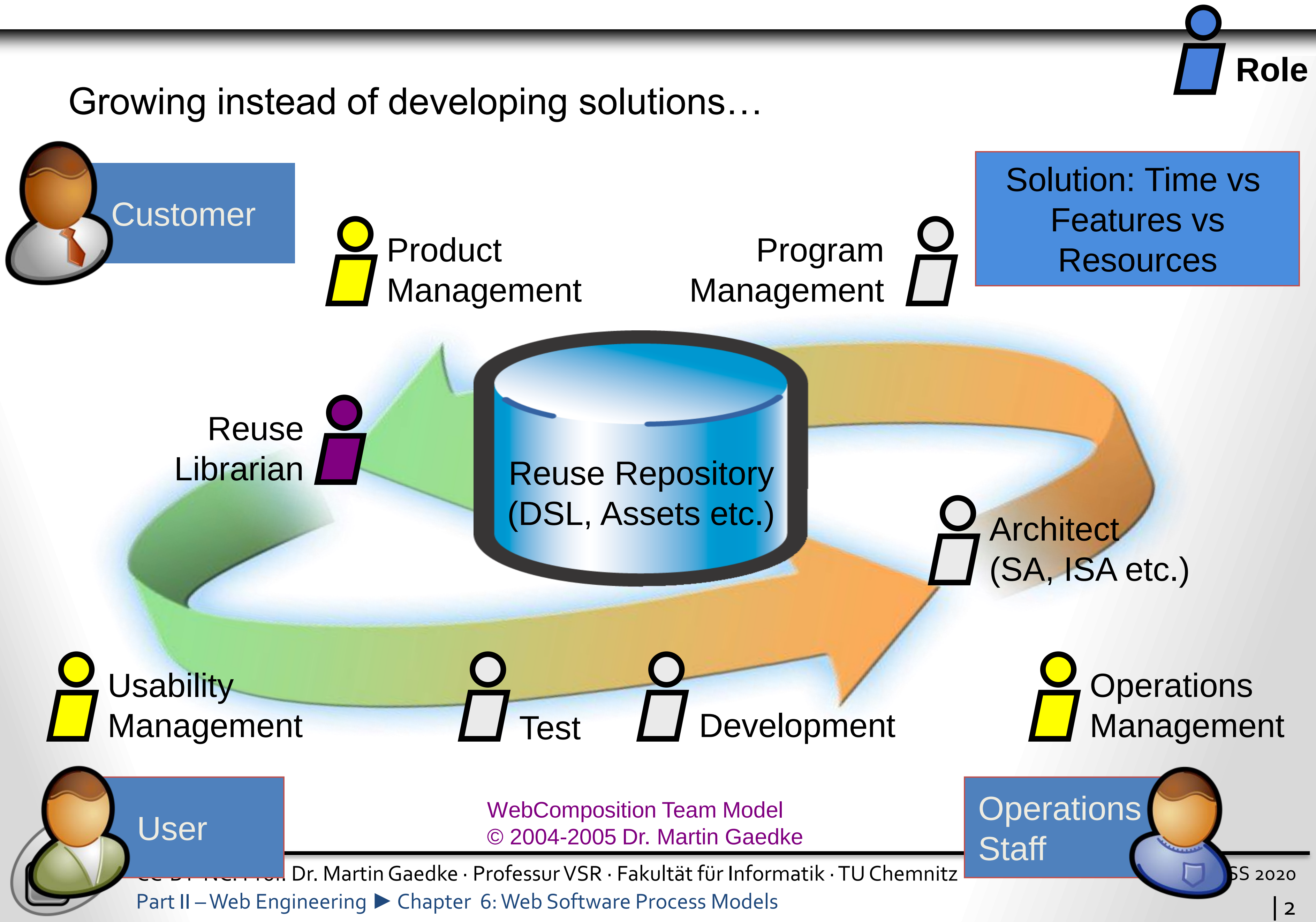
Technische Universität Chemnitz

Fakultät für Informatik

Verteilte und selbstorganisierende Rechnersysteme

# Evolution-oriented Team

Growing instead of developing solutions…

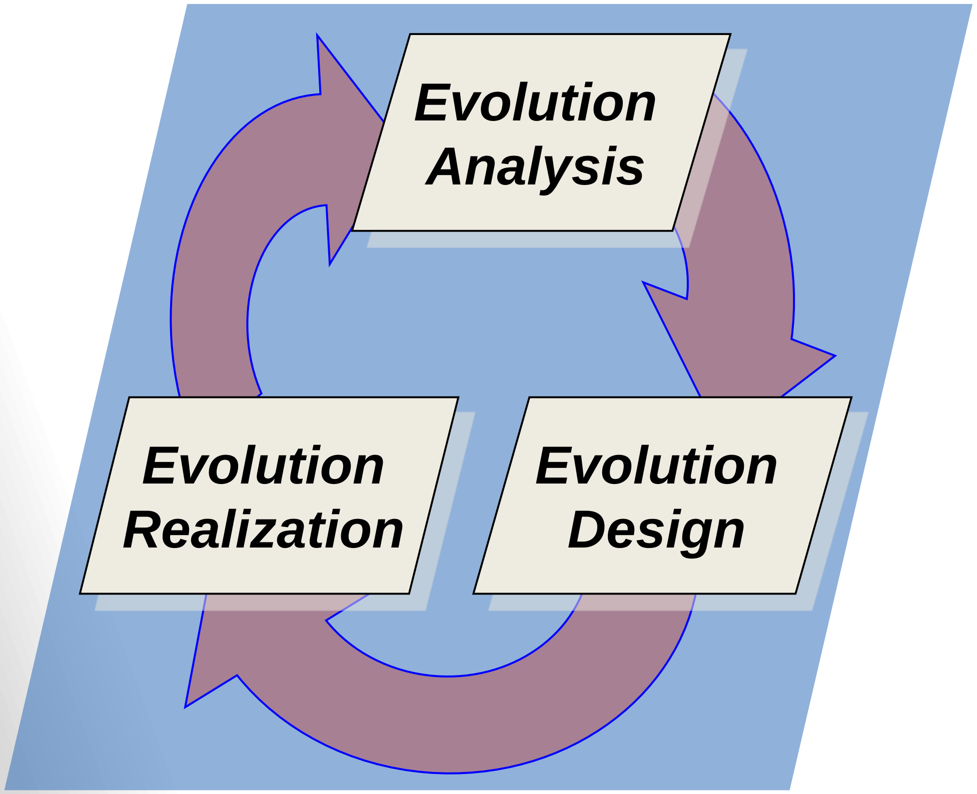


WebComposition Team Model
© 2004-2005 Dr. Martin Gaedke

# WebComposition

- WebComposition project
  - Gaedke et al., University of Karlsruhe, Germany
- Vision:
  - Develop Web applications in an agile way by reusing components and services – Focus on Evolution
- Project:
  - Reuse-Oriented Process Model
  - Middleware: WebComposition Service Linking System (WSLS)
  - Reuse-Repository & UDDI as Registry
  - System Model: i2Map & System Description Framework (SDF)
- History:
  - First developments in 1996
  - Used for different projects, e.g. Global e-Procurement System of Hewlett-Packard, Notebook University Karlsruhe, Mobile University project of Microsoft Research, and builds the core architecture of the project *KIT Integrated Information Management (KIM)*

# WebComposition Life-Cycle Model



- Life-Cycle of any Service with focus on Evolution
  - Web Application is a set of services (realizing features)
- Planning for Reuse (Analysis)
  - Domain Engineering, RNA, Ontology
- Producer Reuse (Design)
  - Development of Reusable Services and Artifacts
- Consumer Reuse (Realization)
  - Development with Reusable Services and Artifacts

# WebComposition Applied

- The Goal-Oriented Evolutionary Process
  - Focus on features and getting a real solution fast
  - Start with only one feature
    - The feature the customer can‘t live  without
    - Learn from the experience with the first feature
  - Restart iteration:
    - Define how to improve solution (Goal) w.r.t. overall Vision
    - Measure the current situation w.r.t. Goal
    - Analyze how to improve the situation w.r.t. Goal
    - Improve solution by adding/changing/removing features/services
    - Check if Goal accomplished
- Aspects
  - Focus on real solutions and less planning
  - Allows for Milestones
  - Reuse-Driven – granularity of reuse units divers
  - Indicator-driven (Different indicators are applied in measuring and analyzing the situation, i.e. Risks, Costs, Quality, etc.)
  - Hypermedia & Composition in mind
  - Guiding models

# Agile Manifesto

- Principles defined by Manifesto for Agile Software Development
  - ► Individuals & interactions > processes & tools
  - ► Working software > comprehensive documentation
  - ► Customer collaboration > contract negotiation
  - ► Responding to change > following a plan
  - ► **Manifesto acknowledges the value of the right items, but focuses the value on the left more**
- For further information, cf.: http://agilemanifesto.org/

# Agile (Process) Findings

- Separation of design and construction
  - ▶ Construction is automated by the compiler
  - ▶ … all the effort is design (this includes coding)
  - ▶ Design is a creative process …
  - ▶ as such: not easy to plan, predictability is impossible

- Iterative development is essential
  - ▶ Allows to deal with changes in required features
- Style of planning
  - ▶ Long term plans: fluid
  - ▶ Short term plans: stable for a single iteration

## Agile (Process) Findings - 2

- Duration for an iteration
  - ► In general as short as possible (depends: customer & developer)
  - ► E.g. XP - between one and three weeks
  - ► E.g. SCRUM - a month
- Accepting the process rather than imposition of a process
  - ► Accepting a process requires commitment
  - ► I.e. empowers development team
  - ► I.e. everyone in team equal place in leadership

## Agile Methodologies

- XP (Extreme Programming)
  - Testing as foundation of development
  - Write tests first
  - Evolutionary design process with focus on refactoring
- Cockburn's Crystal Family
  - Different projects require different methodologies
  - Focus on least disciplined methodology (that could still succeed)
  - Iteration reviews encouraging the process to be self-improving
- **Scrum → Cf. next chapter**
  - Iteration = Sprints (of 30 days)
  - Scrum = Every day fifteen minute meeting
- Feature Driven Development (FDD)
  - Start: Develop an Overall Model, Build a Features List, Plan by Feature
  - Iteration: Design by Feature, Build by Feature
- Many others and related approaches
  - KanBan
  - Lean Software Development / Lean Thinking
  - RUP and MSF can be used in an agile manner – but don‘t have to!

## SECTION://3.3

- Literature

# Literature

- Barry Boehm, A Spiral Model of Software Development and Enhancement, IEEE Computer, 1988(5), pp. 61-72
- Chapter 2, 4: Thomas A. Powell, Web Site Engineering, Prentice Hall PTR
- Chapter 6, 7: David Lowe and Wendy Hall, Hypermedia and the Web – an Engineering Approach, John Wiley & Sons
- Chapter 1, 2, 6, 7, 26: Ian Sommerville, Software Engineering, Addison-Wesley
- Jim McCarty, Dynamics of Software Development, Microsoft Press
- MSF: http://www.microsoft.com/msf
- PMI: http://www.pmi.org (or check IEEE Std 1490-1998)
- Standish Group / CHAOS Report: http://www.standishgroup.com

========================================

Further information available at Lecture Web Site

========================================

# (Agile) Literature

- Kent Beck, Extreme Programming Explained: Embrace Change, Addison-Wesley
- Kent Beck, Martin Fowler, Planning Extreme Programming, Addison-Wesley
- Ken Schwaber, Mike Beedle, Agile Software Development with SCRUM, Prentice Hall
- Alistair Cockburn, Agile Software Development, Addison-Wesley
- Stephen R Palmer, John M. Felsing, A Practical Guide to Feature-Driven Development (The Coad Series), Prentice Hall

# CHAPTER://7

- Scrum

## SECTION://1

- Scrum Introduction

# Scrum in 100 words

- Scrum is an agile process that allows us to focus on delivering the highest business value in the shortest time.
- It allows us to rapidly and repeatedly inspect actual working software (every two weeks to one month).
- The business sets the priorities. Teams self-organize to determine the best way to deliver the highest priority features.
- Every two weeks to a month anyone can see real working software and decide to release it as is or continue to enhance it for another sprint.

# Scrum origins

- Jeff Sutherland
  - ▶ Initial scrums at Easel Corp in 1993
  - ▶ IDX and 500+ people doing Scrum
- Ken Schwaber
  - ▶ ADM
  - ▶ Scrum presented at OOPSLA 96 with Sutherl
  - ▶ Author of three books on Scrum
- Mike Beedle
  - ▶ Scrum patterns in PLOPD4
- Ken Schwaber and Mike Cohn
  - ▶ Co-founded Scrum Alliance in 2002, initially within the Agile Alliance

Microsoft
Agile Project Management with Scrum
Ken Schwaber
Agile Software Development with Scrum
red yellow green blue red blue yellow green blue
Ken Schwaber Mike Beedle
ENTERPRISE SCRUM
Ken Schwaber

## Scrum has been used for:

- Commercial software
- In-house development
- Contract development
- Fixed-price projects
- Financial applications
- ISO 9001-certified applications
- Embedded systems
- 24x7 systems with 99.999% uptime requirements
- the Joint Strike Fighter
- Video game development
- FDA-approved, life-critical systems
- Satellite-control software
- Websites
- Handheld software
- Mobile phones
- Network switching applications
- ISV applications
- Some of the largest applications in use

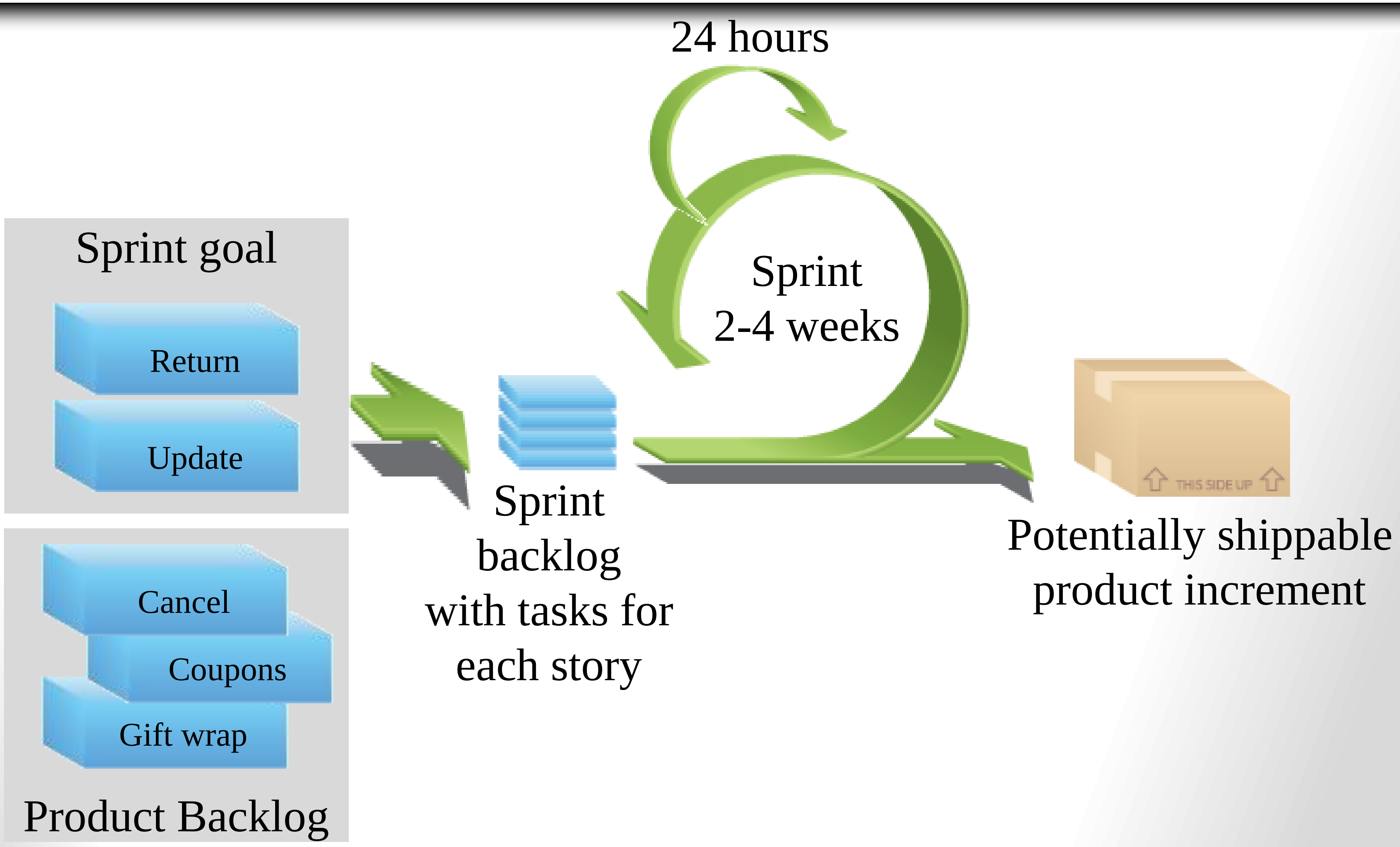
# Characteristics

- Self-organizing teams
- Product progresses in a series of month-long “sprints”
- Requirements are captured as items in a list of “product backlog”
- No specific engineering practices prescribed
- Uses generative rules to create an agile environment for delivering projects
- One of the “agile processes”

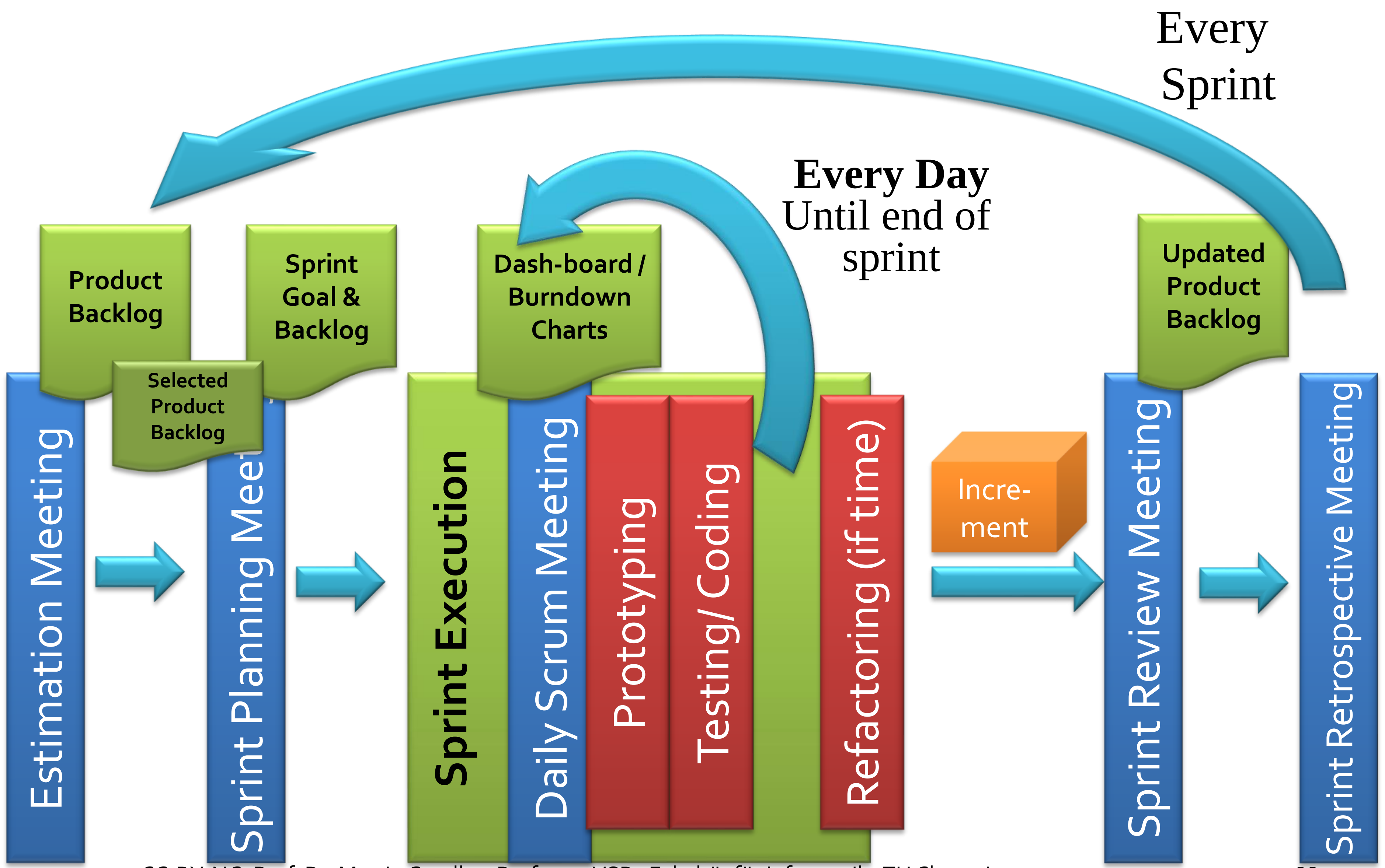
## SECTION://2

- Framework Overview

# Scrum


24 hours
Sprint goal
Return
Update
Cancel
Coupons
Gift wrap
Product Backlog
Sprint
2-4 weeks
Sprint
backlog
with tasks for
each story
THIS SIDE UP
Potentially shippable
product increment

# Scrum 'applied' in more detail…



# A few words about the team approach

- Important roles and concepts
  - ▶ Scrum Master
  - ▶ Scrum Product Owner
  - ▶ Scrum Team
- Where to learn more about Scrum?
  - ▶ CTWE Course in Winter Semester