CORONA EMERGENCY LECTURE

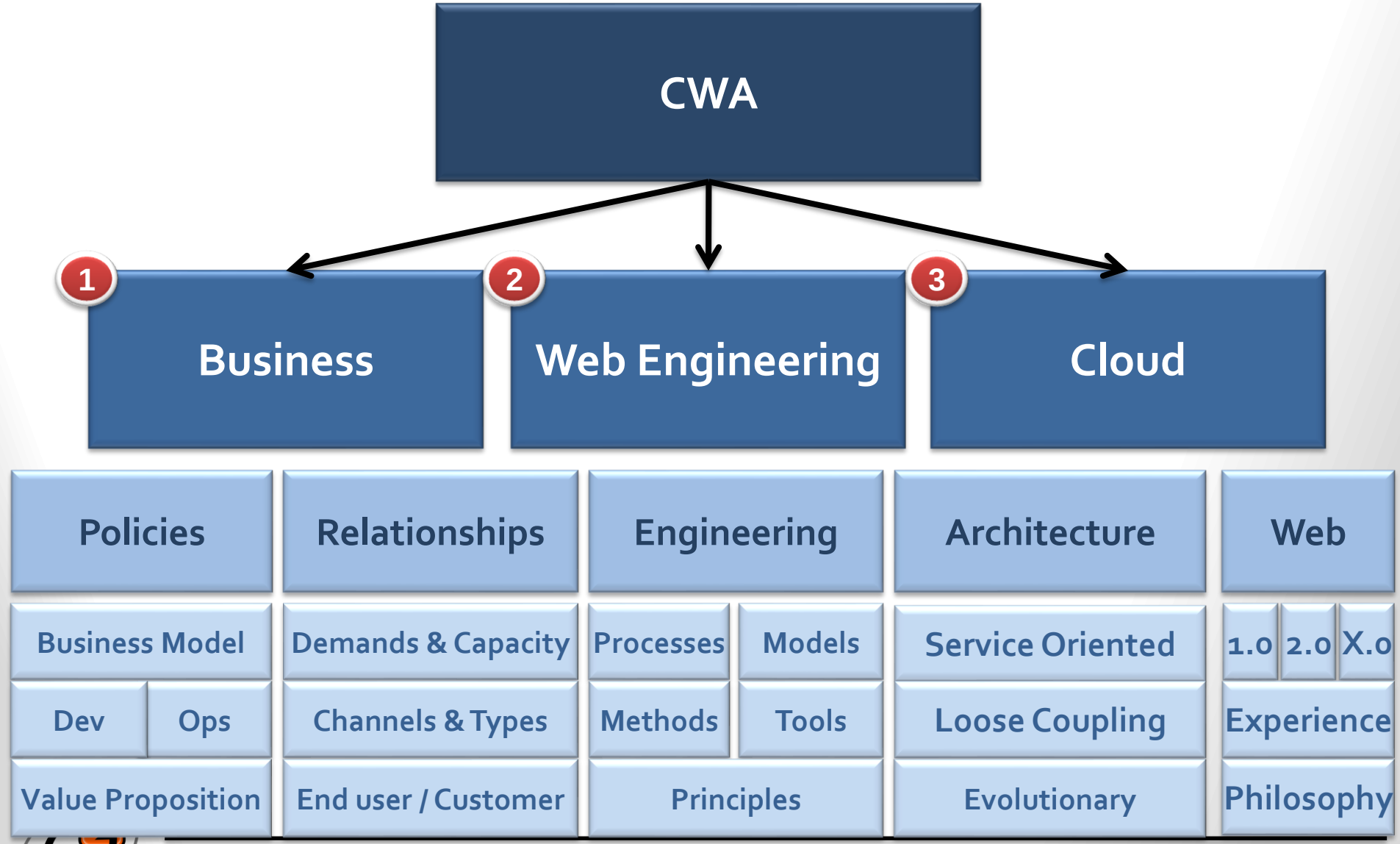
Cloud & Web Anwendungen

Prof. Dr.-Ing. Martin Gaedke
Technische Universität Chemnitz
Fakultät für Informatik
Verteilte und selbstorganisierende Rechnersysteme

Cloud & Web Anwendungen

Prof. Dr.-Ing. Martin Gaedke
Technische Universität Chemnitz
Fakultät für Informatik
Verteilte und selbstorganisierende Rechnersysteme

Lecture Outline



PART II

- Web Engineering

CHAPTER://3

- Introduction

WWW's Technical Aspects

- WWW is a **distributed System**
  - Based on a **Client-Server architecture**
  - Supporting the **Hypermedia Paradigm**
- **Server** provide access to resources
  - E.g. HTML-documents, images, audio, etc.
  - Resources may be created dynamically
- **Client** (**User Agent**) interprets resources
  - Browser present interpretation (Layout, play sound etc.)
  - Other kinds of User Agents may use the resource in other ways (e.g. robots - indexing words)
  - Every request implies a new connection (Stateless)

Web 1.0 (beg. 1990): C/S-Model, Form, eBiz

Client-Side

Presented Resource

Browser

Server-Side

Request

Response

Resource

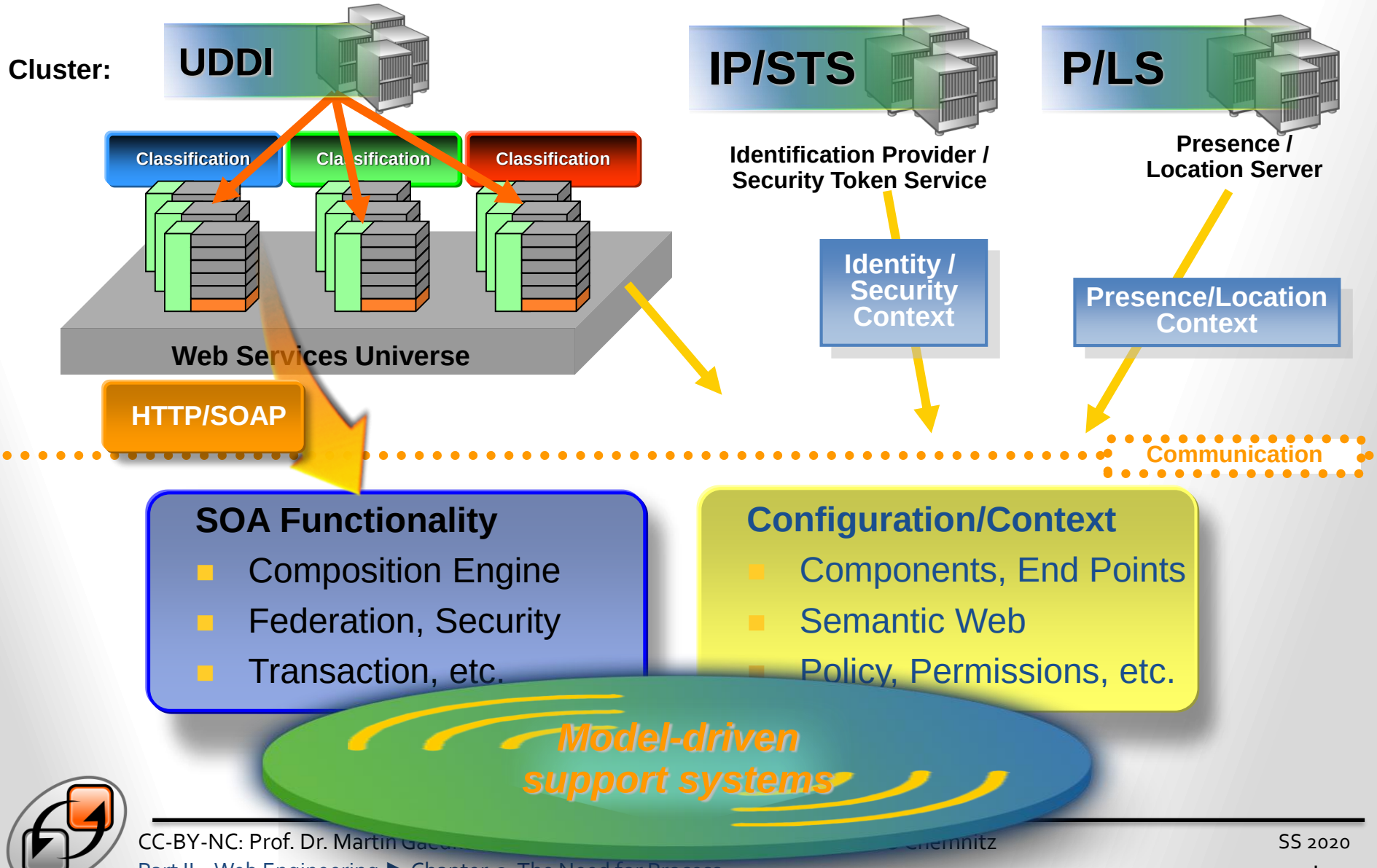
Resource Store

Web Server

Web System
Applicationprotocol: HTTP

Internet
Transportsystem: TCP

Web 2.0 (beg.2000): Services, Prosumer, ID, P/L



"Web 3.0": Social, API1st, AI, Emotions, Serverless

Emotional relationships are key

Distributed application (take crowd into account)

Gamification & other Relationship – oriented part of the application

Social Web – oriented part of the application

User Interface – oriented part of the application
UI/UX & Interaction & Navigation & Client-side code & Sensor-code

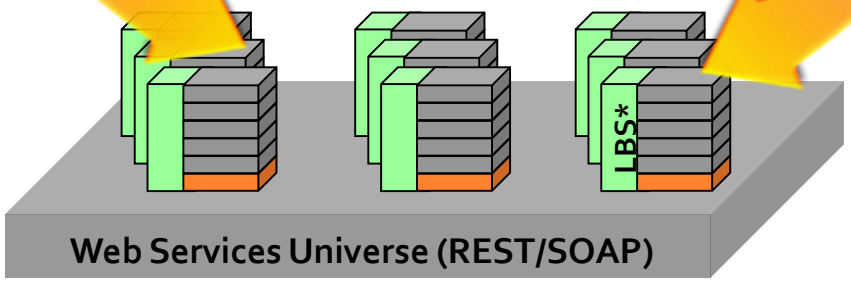
Browser (several)

Embedded User Agents

Mobile Phones and other devices (Tablets)

Communication

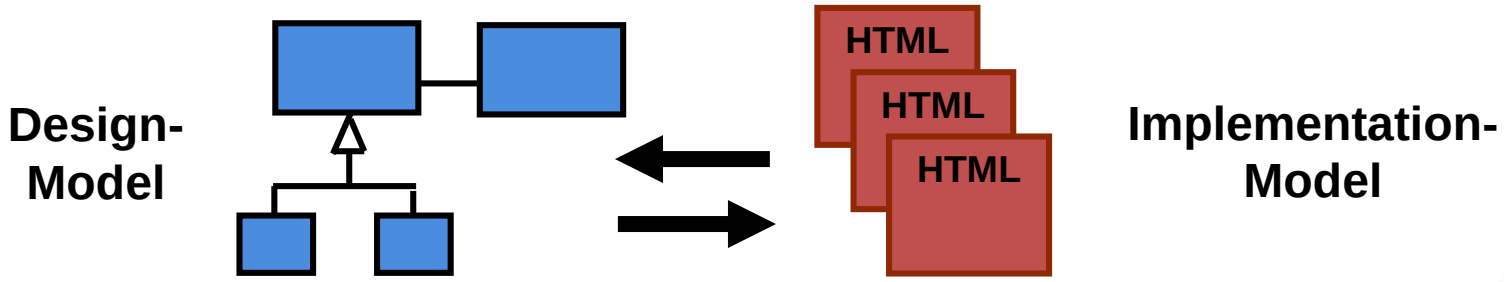
API-First Principle



several Identity Systems

*Location-based Service

Surprisingly, developing Web Applications…



- Still Ad-hoc instead of a disciplined procedure
  - Often Copy-and-Paste Paradigm
  - Lack between design-model and implementation-model
  - Design-concepts get lost in the underlying implementation-model
- Short lifecycles and trends of a Web Application:
  Maintenance and Evolution problems → Reuse Problems

→ Web-Crisis (1st mentioned around 1995)

Need for Process

- Domination of the different requirements calls for a systematic approach
- Producing high-quality Products in a cost-effective way
- Goal – Product should be
  - Maintainable and evolvable
  - Reliable
  - Efficient
  - Appropriate for User Interface (also wrt Hypermedia)
  - Delivered in time with predictable cost

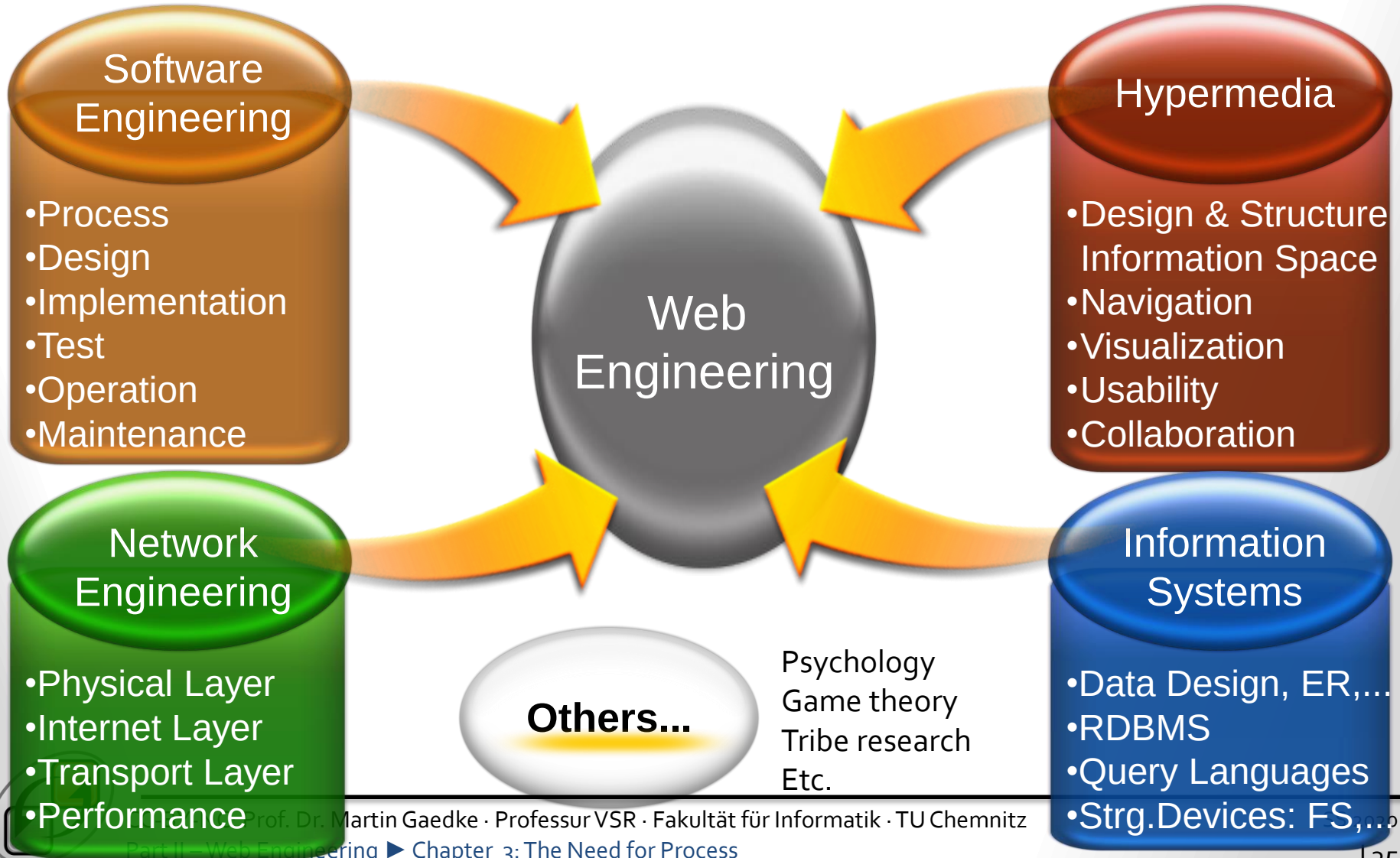
Or simply Software Engineering?

"Fundamental differences [between hypermedia and other disciplines] however, make a pure transposition of techniques both difficult and inadequate. An important part of hypertext design concerns aesthetic and cognitive aspects that software engineering environments do not support."

(Nanard and Nanard, 1995)

Key Knowledge Areas…

…for the production on top of distributed Web-based Systems



Web Engineering

Web Engineering – is the application of systematic, disciplined, and quantifiable approaches to the design, production, deployment, operation, maintenance and evolution of Web-based software products. [Gaedke, 2000]

Evolution of "WebE"

- Web Engineering is a young discipline
- Early 1995/1996 notion of Web Page Design and Web Site Development
  - Development suffers from ad hoc processes
  - Déjà-vu experience of software development in the sixties (cf. Software Engineering, Software Crisis – Workshop 1968)
- 1998 Workshop Web Engineering at the 7th World Wide Web Conference, Brisbane
- Further activities at conferences and workshops
  - WWW, ICWE, HICSS, ICSE, WEBIST etc.
- Journal of Web Engineering (JWE), Rinton Press
  - http://www.rintonpress.com/journals/jwe

Literature

- Thomas A. Powell, Web Site Engineering, Prentice Hall PTR
- David Lowe and Wendy Hall, Hypermedia and the Web – an Engineering Approach, John Wiley & Sons
- San Murugesan, Web Engineering, Sigweb Newsletter Vol. 8, No. 3, Oct. 1999, pp. 28-32
- San Murugesan und Yogesh Deshpande, Web Engineering: Managing Diversity and Complexity of Web Application Development. LNCS 2016, Springer
- SELFHTML, cf. Lecture Web Site
- Gerti Kappel, Birgit Pröll, Siegfried Reich, Werner Retschitzegger (Hrsg.): Web Engineering – Systematische Entwicklung von Web-Anwendungen. dpunkt.verlag
- Martin Gaedke: Komponententechnik für Entwicklung und Evolution im World Wide Web, Shaker Verlag, 2000, ISBN 3-8265-8059-1

==========

Further information available at Lecture Web Site

==========

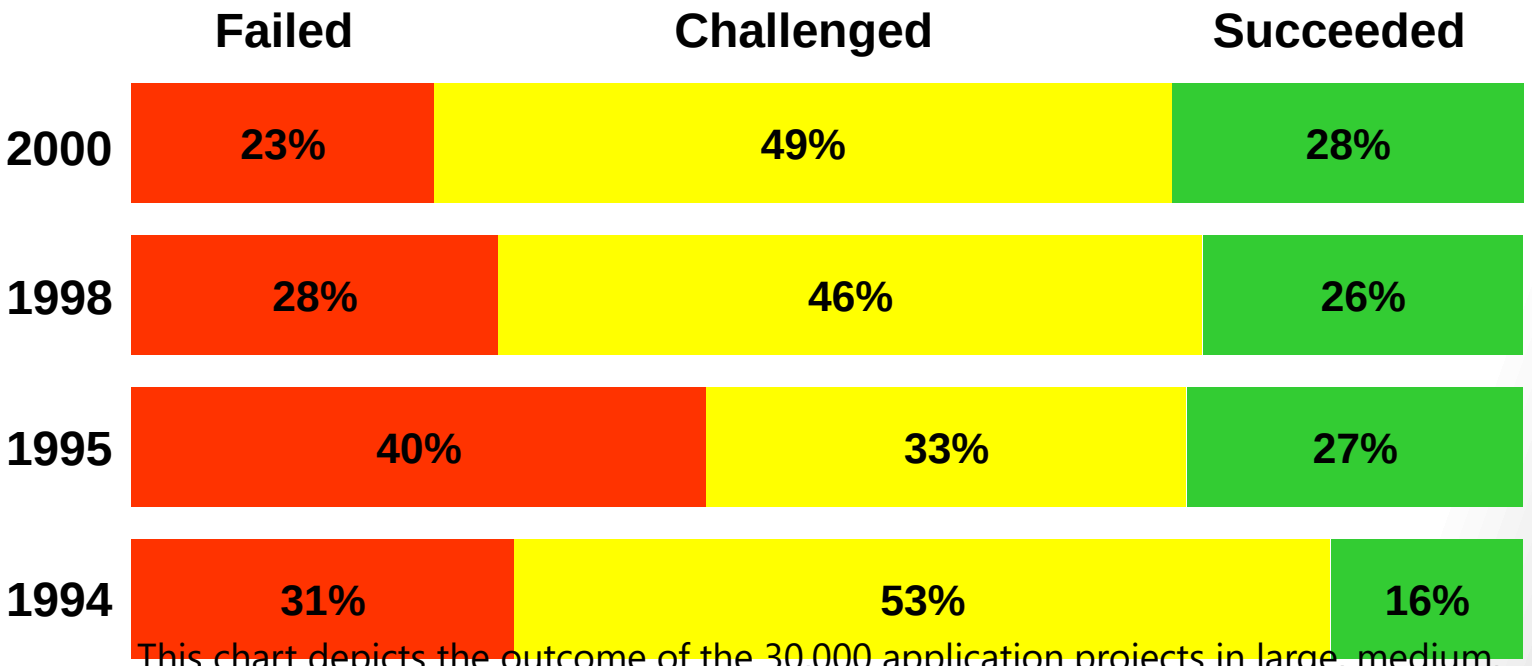
CHAPTER://4

- People, Projects, and Chaos

Where it starts…

- …Understanding the problem
  (wrt "Accomplishing a business need")
- What is the problem domain?
  - Defining the problem – asking questions
  - Which requirements exist?
  - Does Evolution play a major role?
- The problem:
  - What does the customer need vs.
    What does the customer want

Project Failure Rates

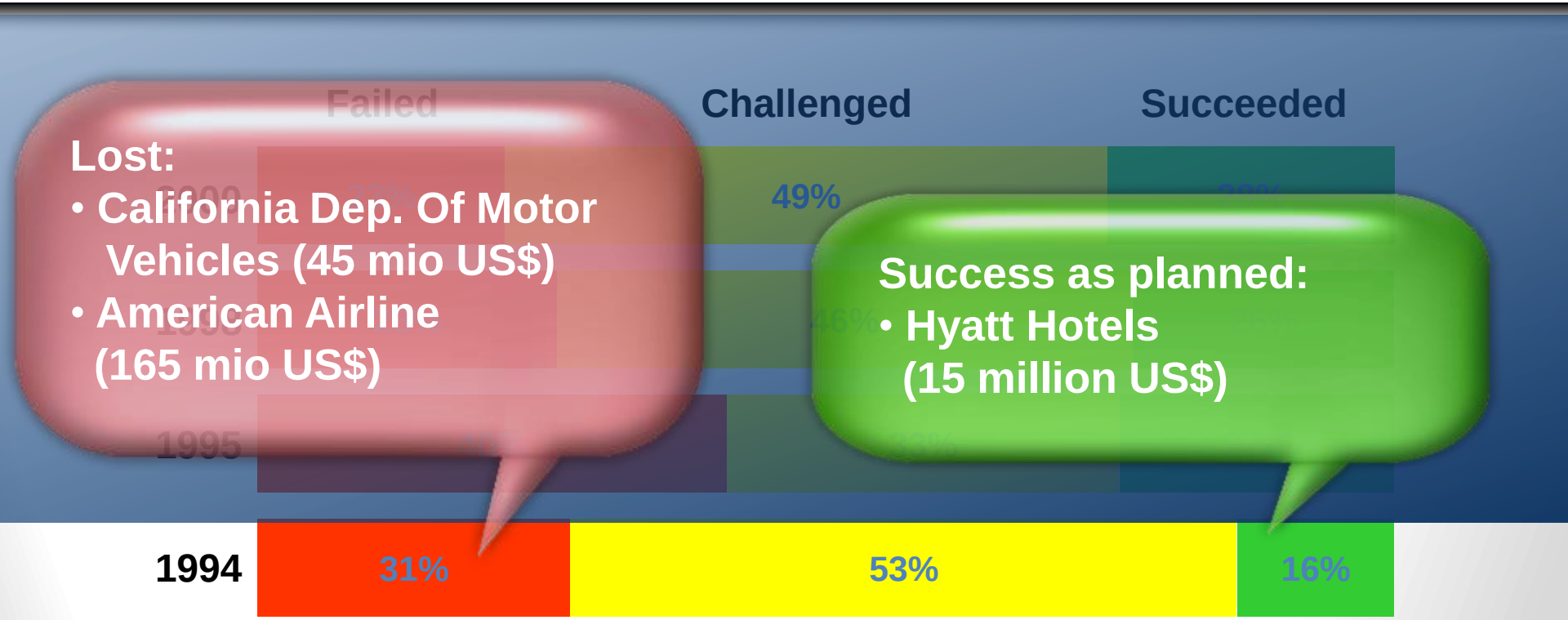


This chart depicts the outcome of the 30,000 application projects in large, medium, and small cross-industry U.S. companies tested by The Standish Group since 1994.

Source: The Standish Group International, *Extreme Chaos, The Standish Group International, Inc., 2000*
http://www.standishgroup.com/

Further Numbers:
2004: Failed: 18%, Challenged: 53%, Succeeded: 29%
Later Chaos Report 2009: Failed: 32%, Challenged: 44%, Succeeded: 24%

Project Failure Rates (2)

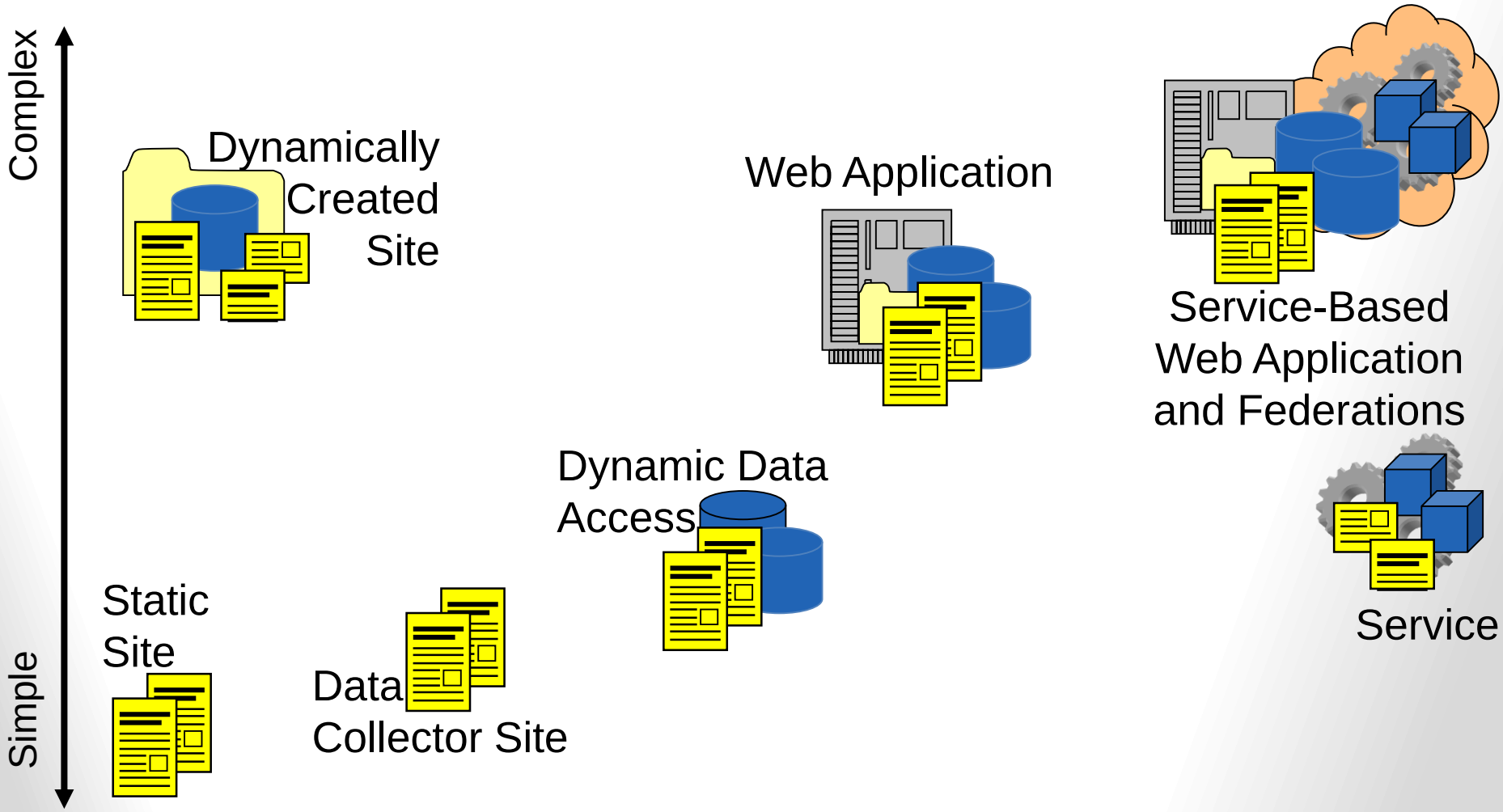


This chart depicts the outcome of the 30,000 application projects in large, medium, and small cross-industry U.S. companies tested by The Standish Group since 1994.

Source: The Standish Group International, *Extreme Chaos, The Standish Group International, Inc., 2000*
http://www.standishgroup.com/

Range of Complexity

Following **"Web Site Engineering: Beyond Web Page Design"**, by Th. Powell et al.



Strategies – towards a solution

- Analyze existing solutions
  - ▶ Outsource
  - ▶ Find & Buy
- Develop new solution
  - ▶ **From the scratch**
  - ▶ **Development with Reuse of existing parts**
- Desired Solution vs. Product Complexity vs. Time vs. Costs

CHAOS Top Success Factors (oldie but goldie)

- What makes a project successful? Success factors identified in 2000.
- Cf. Standish Group Web Site

Success Factor (2000)	Factor
Executive Management (No. 2 in 2003)	18
User Involvement (No. 1 in 2003)	16
Experienced Project Manager	14
Clear Business Objectives	12
Minimizing Scope	10
Requirements Process	8
Standard Software Infrastructure	6
Formal Methodology	6
Reliable Estimates	5
Skilled Staff	5

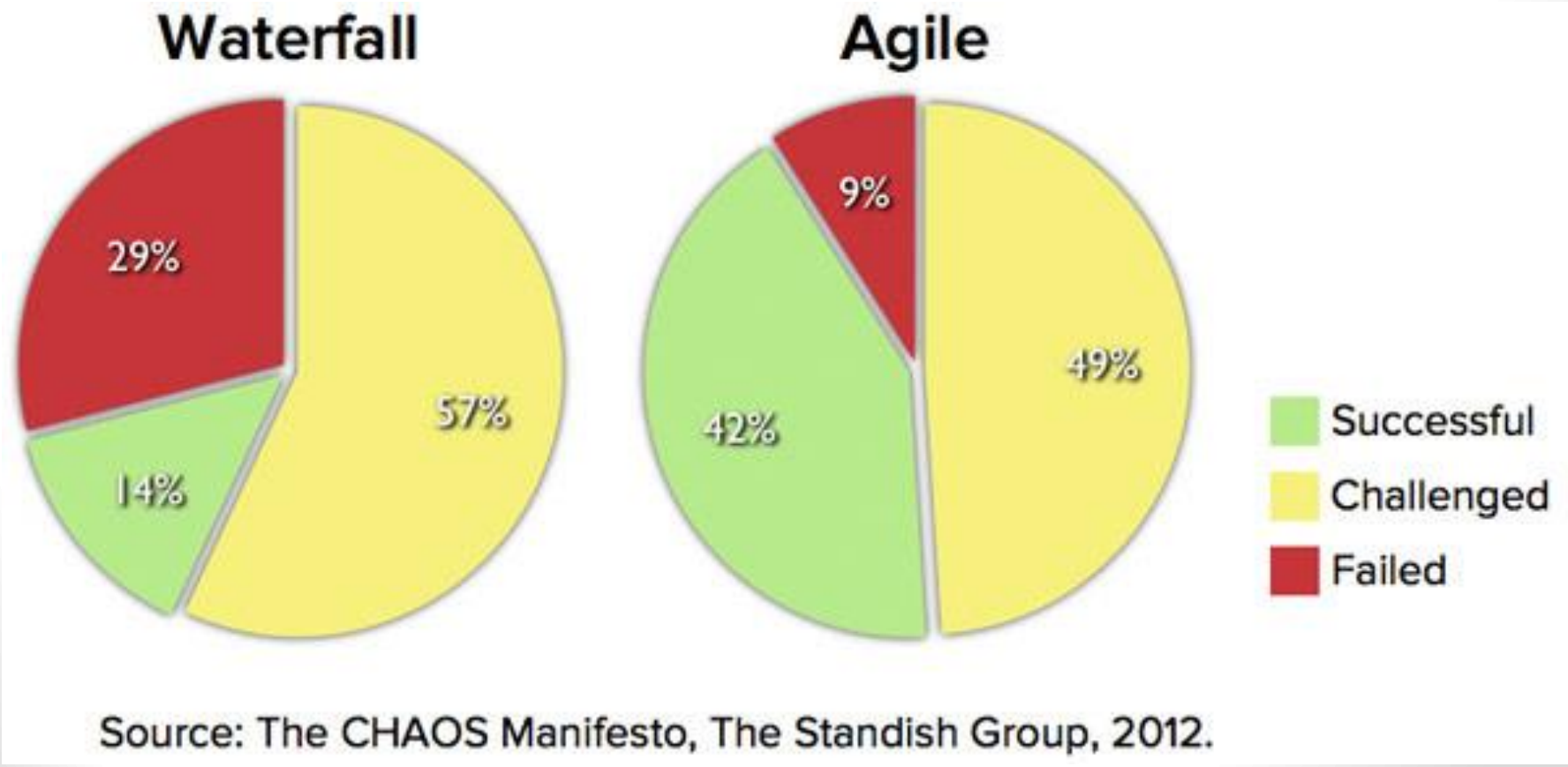
Skilled Team – aka the Product Team

- You need a great team to develop great Web-applications/products
  - "Great" – is a difficult term, if you do not know what you are looking for…
  - "Candidate Attributes" include: Expertise, commitment, attitude, behavior, team skills, thirst for knowledge – usually candidates are great or perfect in all areas…
  - Your team composition must handle this
- Impact of bad choices
  - One poor candidate → "one bad apple can ruin the brunch" → do not hire / usually can be handled
  - Worse: longer period of bad staffing → low performance, bad detail and quality, late products

Thoughts about Budget, People, Time

- Small Teams! These numbers are still a good rule of thumb today (you can also call it Pizza team, if this sounds better)
  - CHAOS: Best success rates
  - 1999:
    time < 6 month, people < 6 and budget < 750.000 US$
  - 2001:
    time < 4 month, people < 4 and budget < 500.000 US$
  - Otherwise try to scale!
- Furthermore:
  Minimize Scope, Open communication, and focus on using Standards
- And go agile….

Waterfall vs Agile project management style



Source: The CHAOS Manifesto, The Standish Group, 2012.

What´s next?

- Agile from project to company
- Culture eats strategy
- DevOps
- And many more trends and developments…

▶ **Homework**

- **Continuous Delivery: Reliable Software Releases Through Build, Test, and Deployment Automation**
  (Addison-Wesley Signature Series)
  by Jez Humble and David Farley

CHAPTER://5

- Project Management

Introduction

- What is Project Management all about?
- Project Management Institute Body of Knowledge, provides over 35 years of experience
  - Project management is the application of knowledge, skills and techniques to execute projects effectively and efficiently. It's a strategic competency for organizations, enabling them to tie project results to business goals — and thus, better compete in their markets.
  - PM Knowledge Areas and required skills: Integration, Scope, Time, Cost, Quality, HR, Communications, Risk and Procurement Management
  - Cf. PMI's PMBOK http://www.pmibok.com and Standish Group

Problems with Project Management

- Unrealistic Schedules
  - ▶ May yield to artificial documents
- Avoid cost of iteration/going back a stage
  - ▶ Rewriting documents by bad solutions
- Seriously reviewing and approving takes time
  - ▶ Next stage may start before document approved
- Deliverables not suitable for reuse-oriented models
  - ▶ Documents are likely to constrain reuse
- People skills
  - ▶ delegation, negotiation
- Leading vs. Managing

PRJ-Language: WBS & Program

- Work Breakdown Statement (WBS)
  - Categorized list of tasks
  - With an estimate of resources required
- Program
  - A group of projects managed in a coordinated way to obtain beneftis not available from managing them individually
  - Examples
    - E.g. Program: PRJ Design, PRJ Construction
    - E.g. Program: PRJ Version 1, PRJ Version 2

Some Activities to start with…

- Activities to start with in the Initial Phase
- Prepare for Product Life Cycle Management
- Projected Organization and Personnel Management
- Establish open communication in the team
- Advocate for customer vs.
  Advocate for team vs. …
- Interim Milestone (IM) of Initial Phase:
  - ▶ Core Team Organized

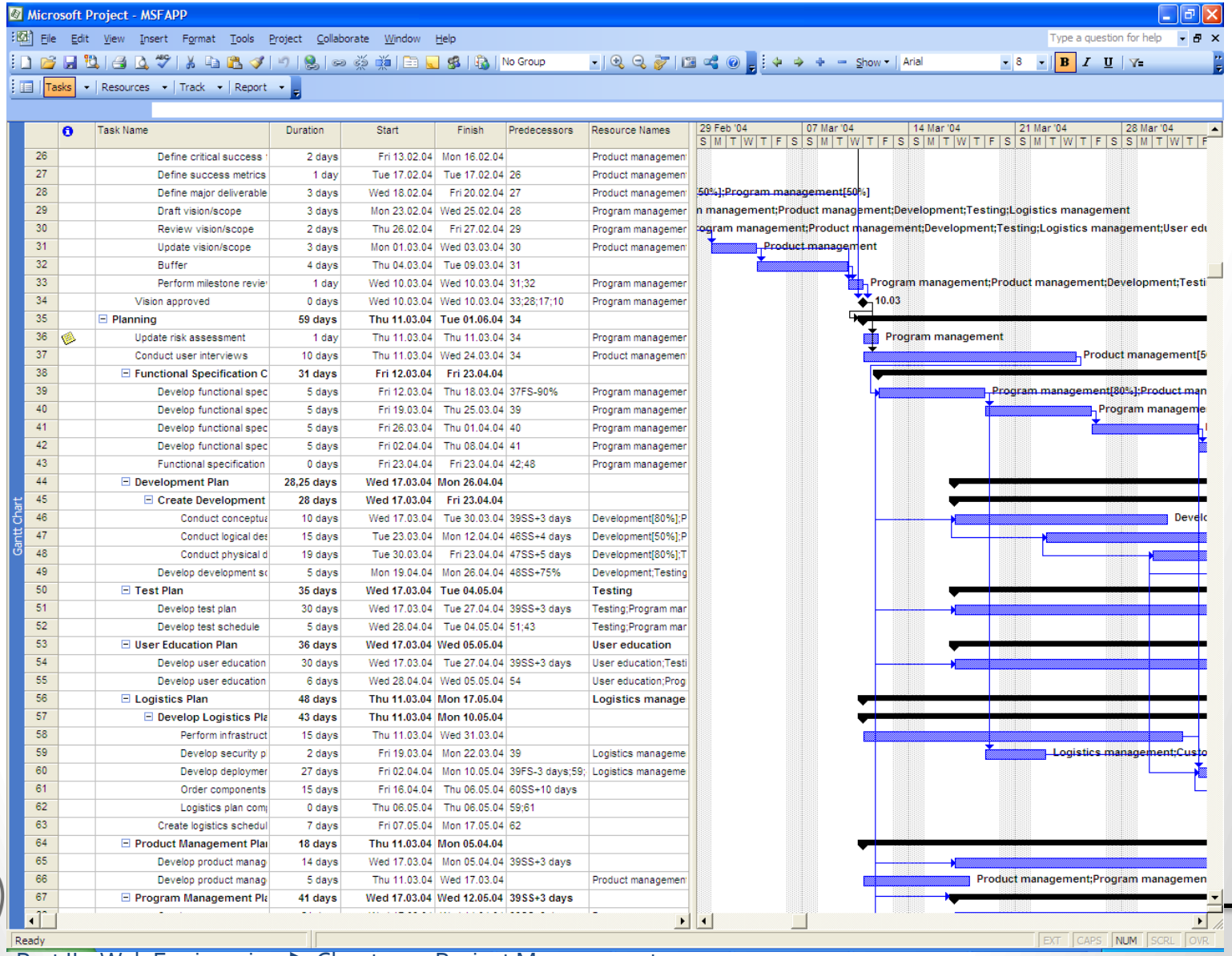
A little PM Toolkit

- General Project Management Tools
  - GANTT charts
  - Deliverables
  - Milestones
  - Many more, e.g. PERT diagrams
- Dedicated tools
  - Risk management
  - Prediction knowledge bases
  - Requirement tools
- Few (if any) project management research dedicated to Web Application production
  - Requires experience
- Process Models
  - Deliverable-oriented process models

PM-Tool: Gantt Chart

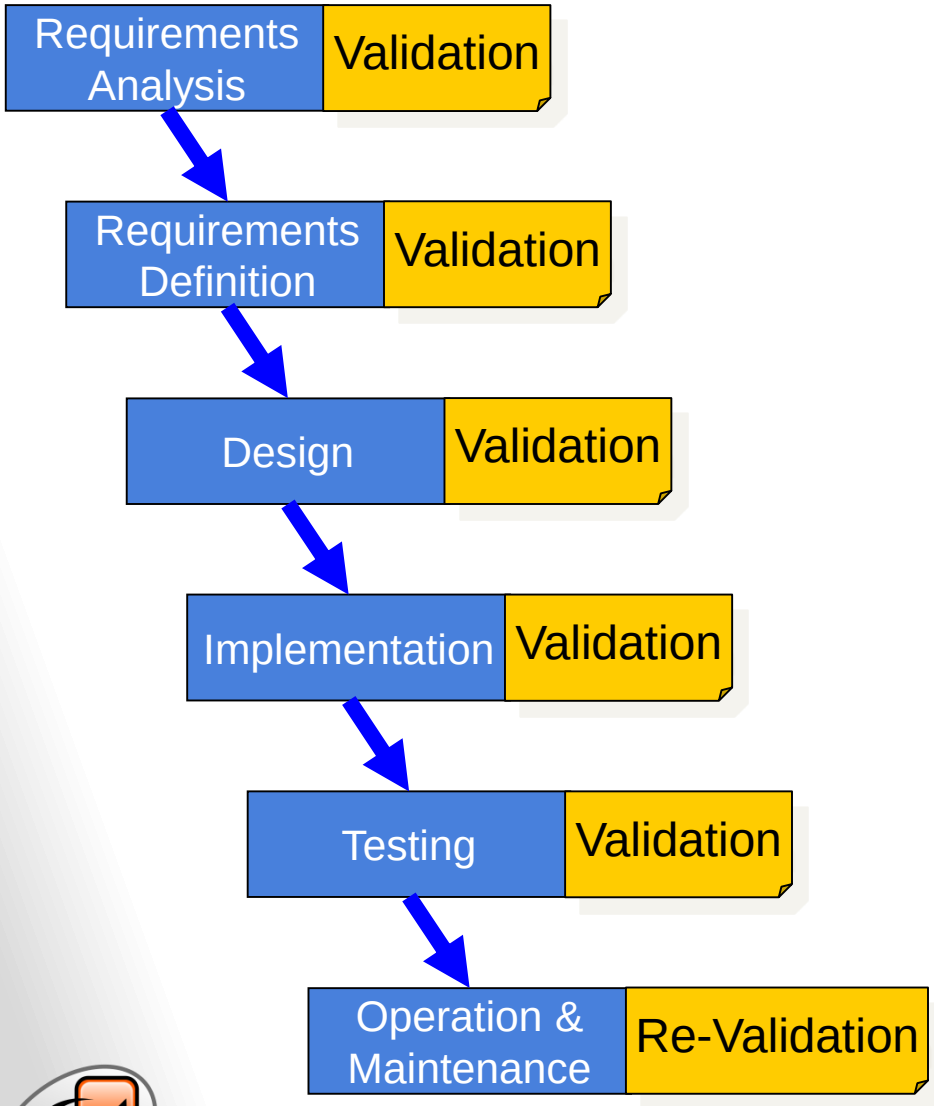
- Gantt Chart - Preferred visual reporting device used for conveying a project's schedule.
  - Graphically displays the work breakdown, total duration needed to complete tasks, as well as %completion
  - Does not display level of effort, and is not an effective planning tool on its own
  - May be integrated with other spreadsheet-type reporting devices that convey additional information related to project planning

PM-Tool: Gantt Chart 2



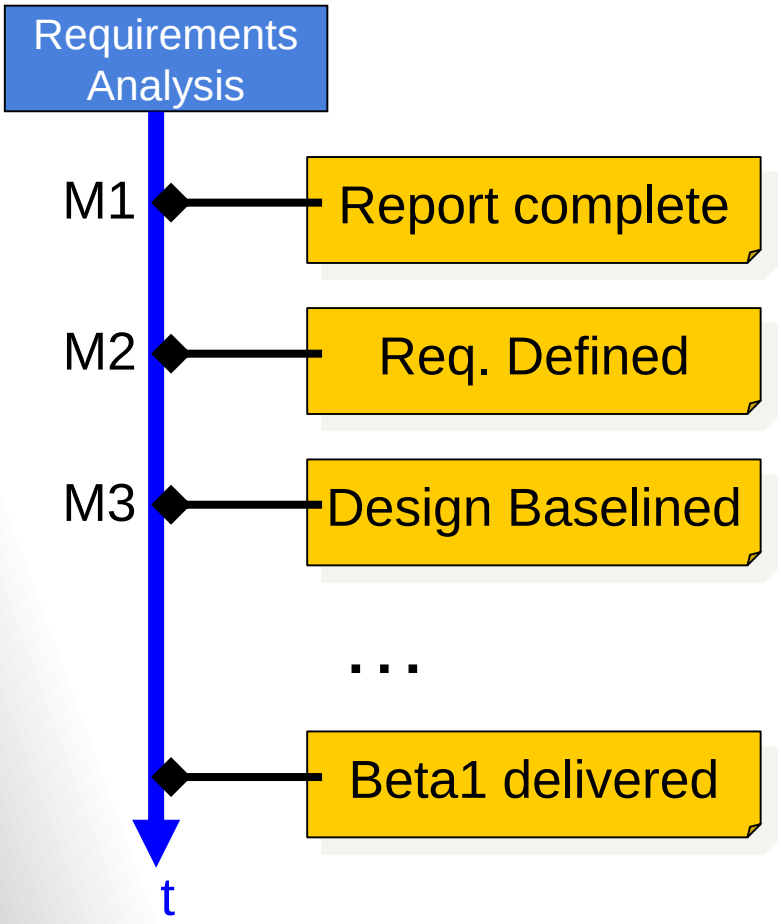
	Task Name	Duration	Start	Finish	Predecessors	Resource Names
26	Define critical success	2 days	Fri 13.02.04	Mon 16.02.04		Product management
27	Define success metrics	1 day	Tue 17.02.04	Tue 17.02.04	26	Product management
28	Define major deliverable	3 days	Wed 18.02.04	Fri 20.02.04	27	Product management
29	Draft vision/scope	3 days	Mon 23.02.04	Wed 25.02.04	28	Program management
30	Review vision/scope	2 days	Thu 26.02.04	Fri 27.02.04	29	Program management
31	Update vision/scope	3 days	Mon 01.03.04	Wed 03.03.04	30	Product management
32	Buffer	4 days	Thu 04.03.04	Tue 09.03.04	31	
33	Perform milestone revie	1 day	Wed 10.03.04	Wed 10.03.04	31;32	Program management
34	Vision approved	0 days	Wed 10.03.04	Wed 10.03.04	33;28;17;10	Program management
35	**Planning**	**59 days**	**Thu 11.03.04**	**Tue 01.06.04**	**34**	
36	Update risk assessment	1 day	Thu 11.03.04	Thu 11.03.04	34	Program management
37	Conduct user interviews	10 days	Thu 11.03.04	Wed 24.03.04	34	Product management
38	**Functional Specification C**	**31 days**	**Fri 12.03.04**	**Fri 23.04.04**		
39	Develop functional spec	5 days	Fri 12.03.04	Thu 18.03.04	37FS-90%	Program management
40	Develop functional spec	5 days	Fri 19.03.04	Thu 25.03.04	39	Program management
41	Develop functional spec	5 days	Fri 26.03.04	Thu 01.04.04	40	Program management
42	Develop functional spec	5 days	Fri 02.04.04	Thu 08.04.04	41	Program management
43	Functional specification	0 days	Fri 23.04.04	Fri 23.04.04	42;48	Program management
44	**Development Plan**	**28,25 days**	**Wed 17.03.04**	**Mon 26.04.04**		
45	**Create Development**	**28 days**	**Wed 17.03.04**	**Fri 23.04.04**		
46	Conduct conceptua	10 days	Wed 17.03.04	Tue 30.03.04	39SS+3 days	Development[80%];P
47	Conduct logical des	15 days	Tue 23.03.04	Mon 12.04.04	46SS+4 days	Development[50%];P
48	Conduct physical d	19 days	Tue 30.03.04	Fri 23.04.04	47SS+5 days	Development[80%];T
49	Develop development s	5 days	Mon 19.04.04	Mon 26.04.04	48SS+75%	Development;Testing
50	**Test Plan**	**35 days**	**Wed 17.03.04**	**Tue 04.05.04**		**Testing**
51	Develop test plan	30 days	Wed 17.03.04	Tue 27.04.04	39SS+3 days	Testing;Program mar
52	Develop test schedule	5 days	Wed 28.04.04	Tue 04.05.04	51;43	Testing;Program mar
53	**User Education Plan**	**36 days**	**Wed 17.03.04**	**Wed 05.05.04**		**User education**
54	Develop user education	30 days	Wed 17.03.04	Tue 27.04.04	39SS+3 days	User education;Testi
55	Develop user education	6 days	Wed 28.04.04	Wed 05.05.04	54	User education;Prog
56	**Logistics Plan**	**48 days**	**Thu 11.03.04**	**Mon 17.05.04**		**Logistics manage**
57	**Develop Logistics Pla**	**43 days**	**Thu 11.03.04**	**Mon 10.05.04**		
58	Perform infrastruct	15 days	Thu 11.03.04	Wed 31.03.04		
59	Develop security p	2 days	Fri 19.03.04	Mon 22.03.04	39	Logistics manageme
60	Develop deploymer	27 days	Fri 02.04.04	Mon 10.05.04	39FS-3 days;59;	Logistics manageme
61	Order components	15 days	Fri 16.04.04	Thu 06.05.04	60SS+10 days	
62	Logistics plan com	0 days	Thu 06.05.04	Thu 06.05.04	59;61	
63	Create logistics schedul	7 days	Fri 07.05.04	Mon 17.05.04	62	
64	**Product Management Pla**	**18 days**	**Thu 11.03.04**	**Mon 05.04.04**		
65	Develop product manag	14 days	Wed 17.03.04	Mon 05.04.04	39SS+3 days	
66	Develop product manag	5 days	Thu 11.03.04	Wed 17.03.04		Product management
67	**Program Management Pla**	**41 days**	**Wed 17.03.04**	**Wed 12.05.04**	**39SS+3 days**	

PM-Tool: Deliverables



- Requirements Analysis:
  - Feasibility study
- Requirements Definition:
  - Req. specification
- Design:
  - Design specification
- Implementation:
  - Web Application code
- Testing:
  - Test result report
- Delivery:
  - Acceptance test / final system
- Operation:
  - Usage report, feedback

PM-Tool: Milestones



- **Milestones** – Significant event in the project, usually completion of a major deliverable
  - Important checkpoints or interim goals for a project to be met at a given ***date***
  - Used to catch scheduling problems early
  - Name by noun-verb form, e.g. "report due", "prototype complete"
- Milestone "Rules of thumb"
  - Too many milestones are useless
  - Focus on hard-results (Not: 80% of Site finished)

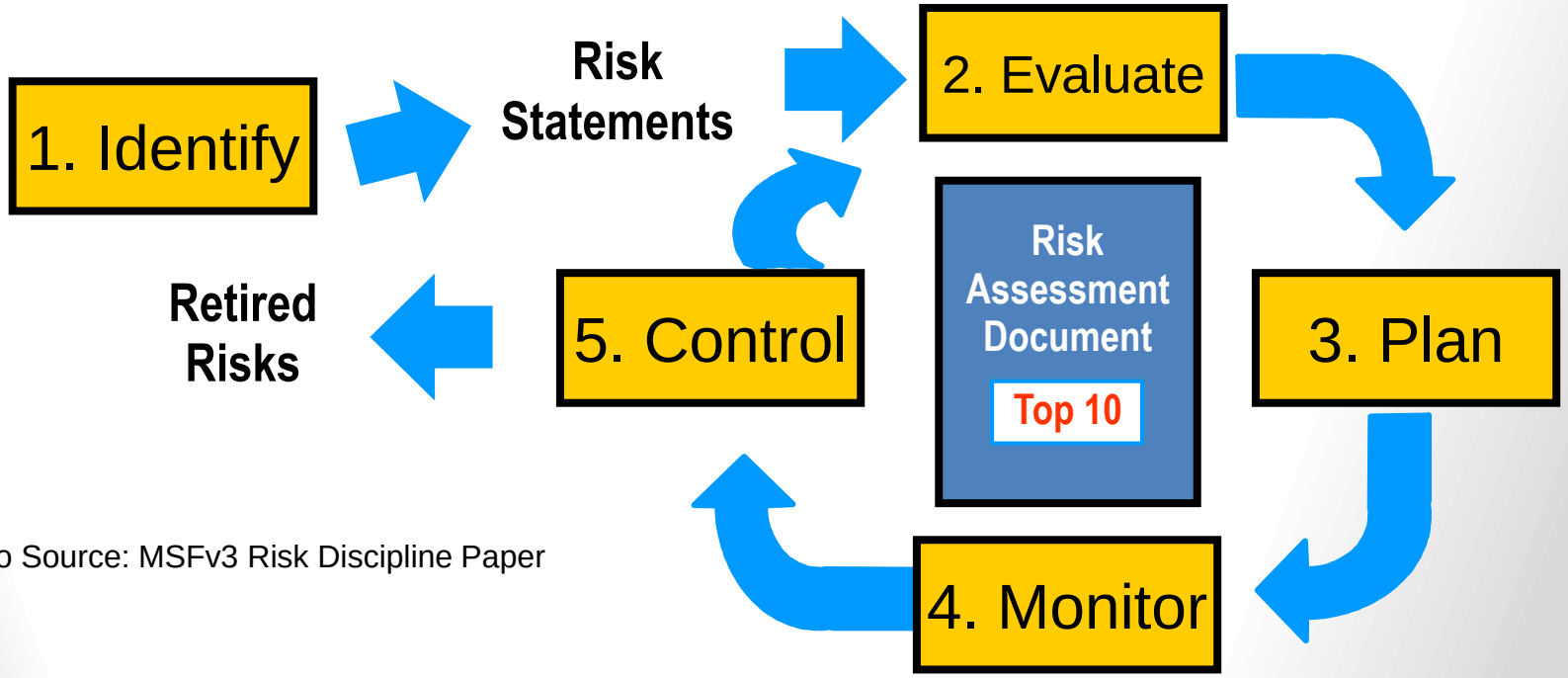
Risks & Potential Sources

- Potential sources:
  - **People** – e.g. Customer, Team
  - **Process** – e.g. Schedule, Requirements, Design
  - **Technology** – e.g. Availability, Security
  - **Environmental** – e.g. Legal, Business

- Risk Management is a recurring process throughout the whole project
  - There is no excuse for not doing it…

Risk Management Process

- **Risk** – Any event that could potentially have a negative impact on the project
  - Remember: Risk is not a problem as refers to the future!
  - Evaluate: Quality analysis (prioritize effect on project objective) & Quantity analysis (probability and estimating implications)



Related to Source: MSFv3 Risk Discipline Paper

Risk Assessment (simplified)

Risk	Probability (1-7)	Impact (1-7)	Total Risk (P x I)
Lead Developer leaves team	2	6	**12**
Regulation §1-3	2	7	**14**
WS not available	4	5	**20**

Organization Structure

Will not be covered by this lecture

Some notes:

- Few (if any) Web Engineering Research
- Cf. standard Software Engineering literature
- Regarding evolution of content
  - ▶ Workflows respectively process models for publishing content and people involved
  - ▶ Cf. Content Management Systems literature
- Cf. E.g. www.holacracy.org

Costs Estimation

Will not be covered by this lecture

Some notes:

- Few Web Engineering Research
- Cf. standard Software Engineering literature
- Issues to look at
  - Costs for Marketing (search engines, advertising)
  - Return on investment (ROI) have to be considered, e.g. advertisements, integration of other Web Applications (Marketplaces, Web Application Production Lines)
  - Computer & Network costs
  - Political influences

CHAPTER://6

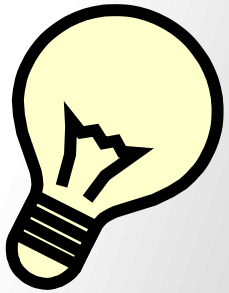
- Web Software Process Models

Goals of Web Engineering

- Develop (high quality) Web Applications
  - ▶ Effective
  - ▶ Efficient
  - ▶ Achieve desired application
  - ▶ in a Predictable Way
- **Maintain and Evolve**
  - ▶ **Plan for change** (Solution may change the problem!!!)
- …using systematic, disciplined and quantifiable Approaches: Process Models

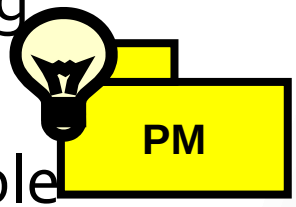
Production Process

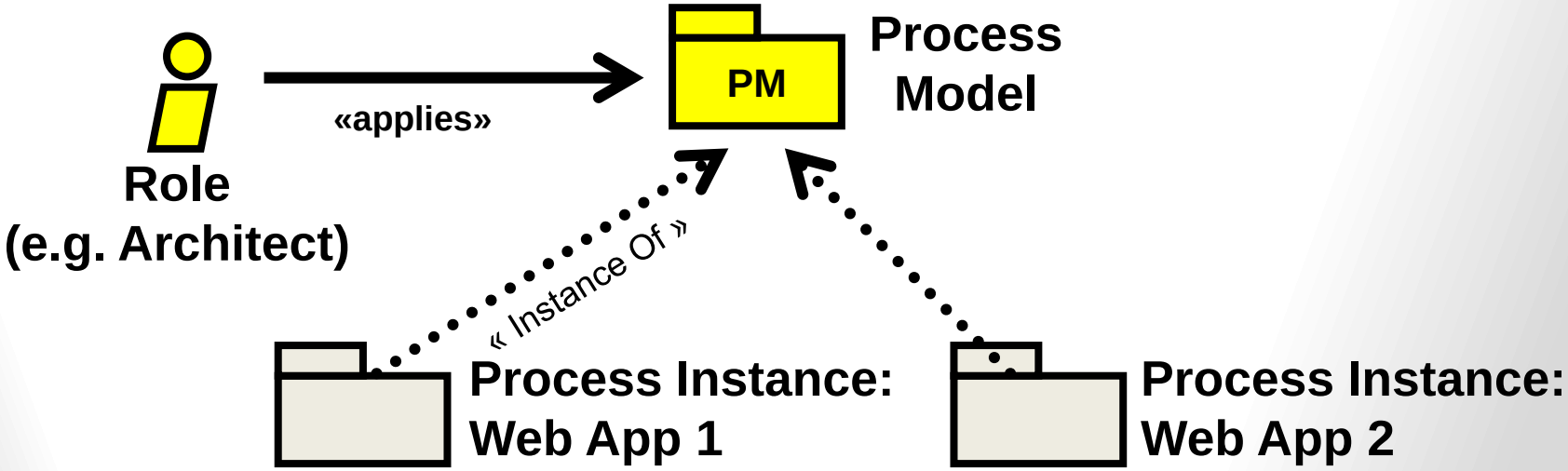
- "Idea" during "software crisis"
  - Apply methods to software development
  - Describe the process of software development with Process Model
- "Visible" development process
  - Important for Project Management
- Lessons learned in Software Engineering:
  - Process is complex and variable
  - Detailed process Models – still research
- Different Processes appropriate for different classes of problems
- Lessons Learnt: Applying a process adds "6% success factor"



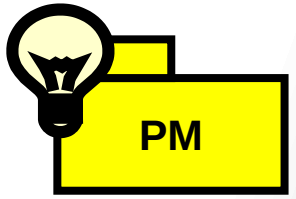
Process Model

- Applying a process model to a Web Engineering specific problem
- Focus on systematic, disciplined and quantifiable development and evolution

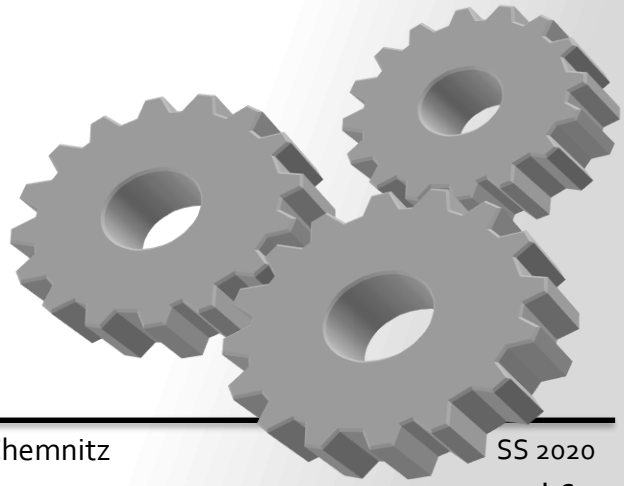




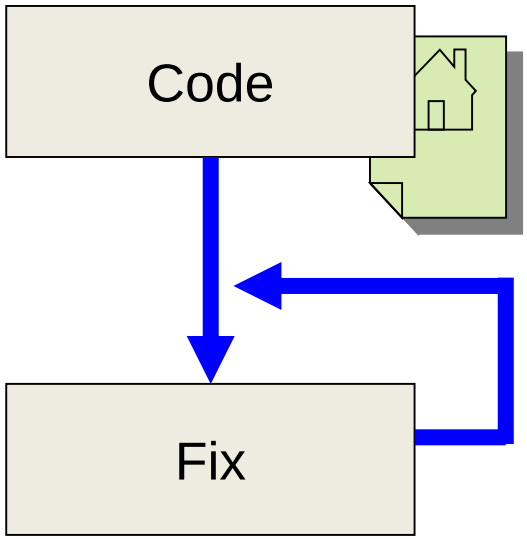
Process Models and WebE



- Code-and-fix model (ad hoc development)
- Classical style
  - **The Waterfall model**
  - Prototyping model
  - V-Model
  - Evolutionary Development model
  - **Spiral model**
  - **Rational Unified Process model**
  - **MSFv3 Process model**
- Agile style
  - **Agile Processes**
  - **Reuse-Oriented Approaches**
  - **WebComposition Process model**
  - **Agile Processes, XP, Scrum**
- And many, many more…

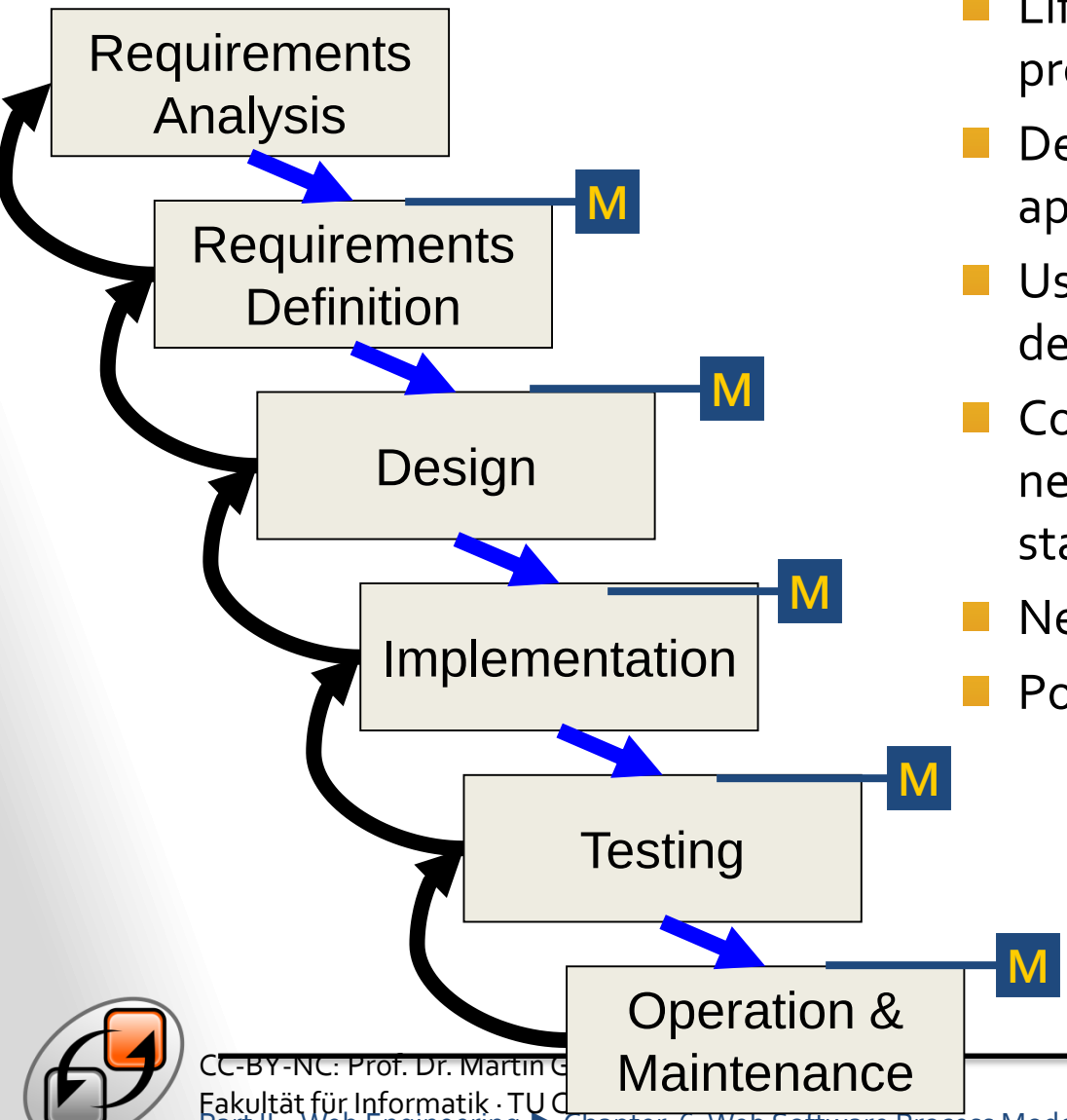


In the early beginning: Code-and-fix Model



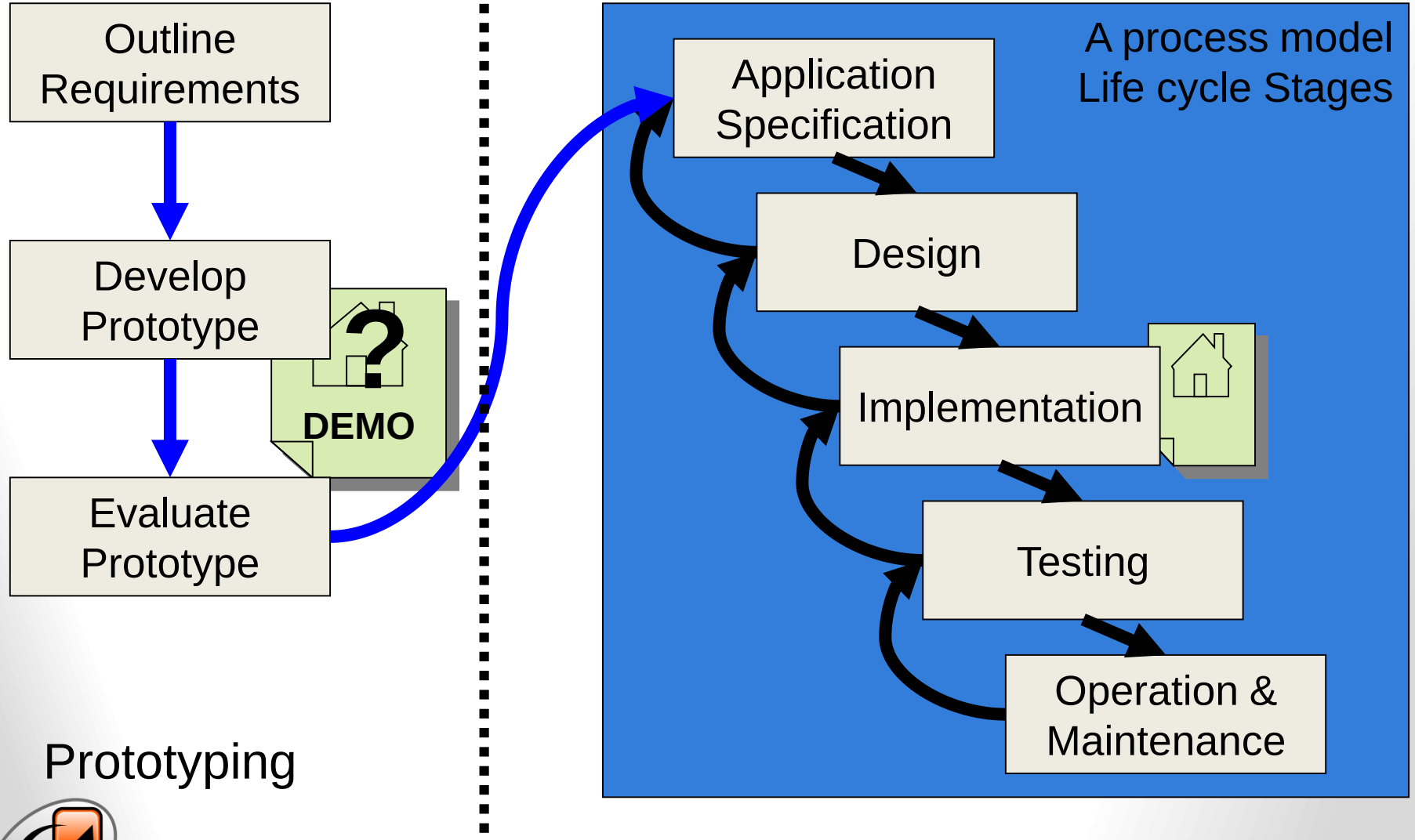
- Oldest Model – and still in use
  - Works "pretty well" for small projects in the early beginning
  - Test phases usually very long and unpredictable
  - Susceptible to Spaghetti-Code & -Linking
- User requirements often neglected
- Fixing bugs expensive
- Unsuitable for team work
- Unsuitable for most Web projects

The Waterfall Model



- Life Cycle Model – software process as number of stages
- Derived from other engineering approaches (late sixties)
- Used for practical Web Application development in several variants
- Completion of stage / milestone needed to proceed to following stage (M)
- Negative: Problems in early stages
- Popular for management

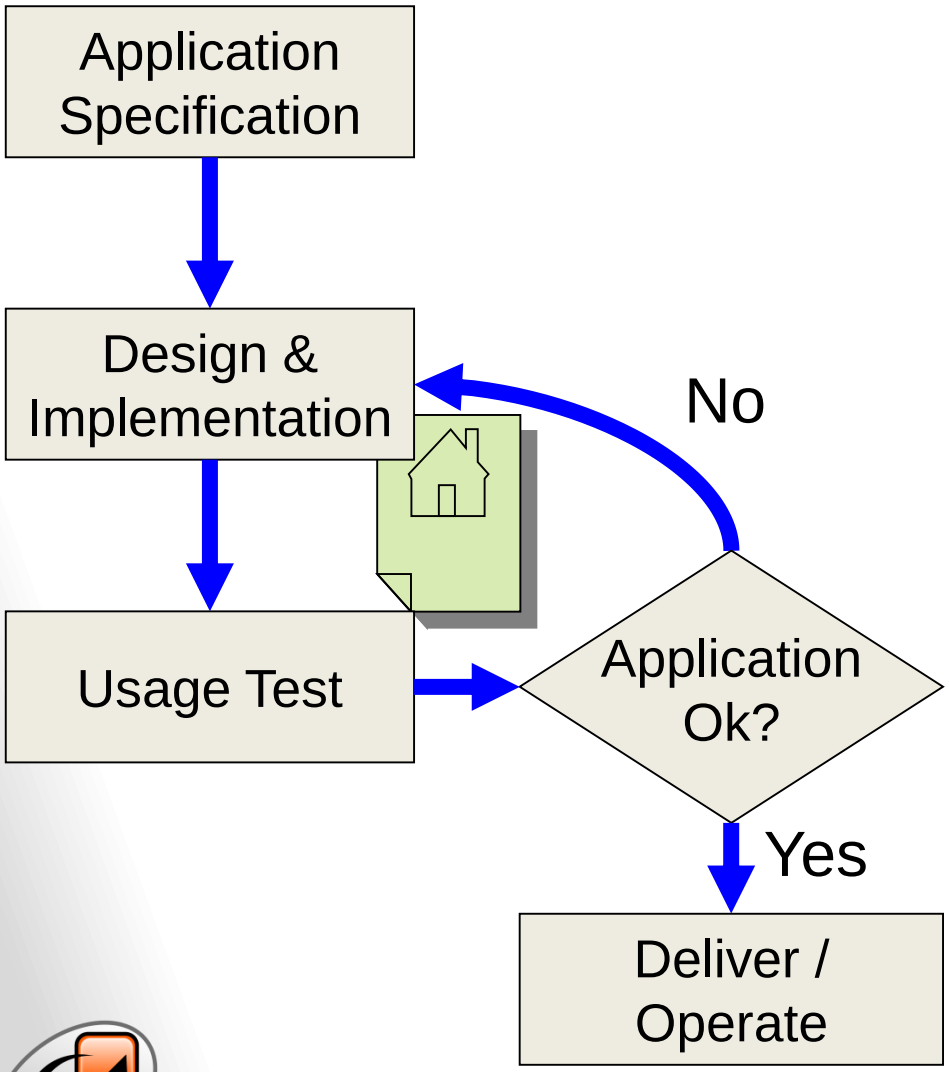
Prototyping Model



Prototyping Model II

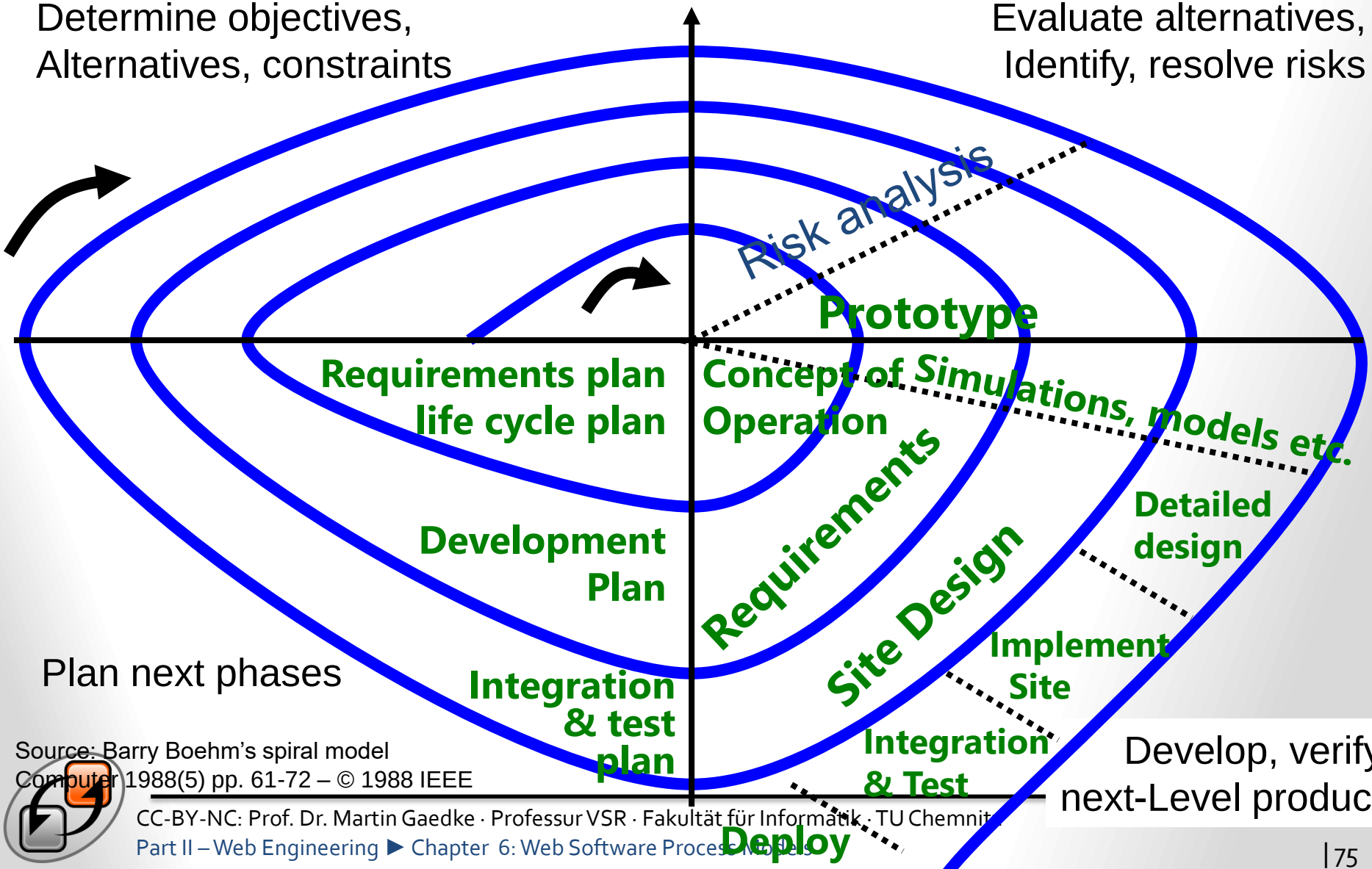
- Prototype – *only* responsible for defining system requirement
- Suitable if system requirements can not described completely in the beginning
- Applicable for reuse approaches
- Open Process Model – use of any process model
- Further Issues
  - ▶ Good for motivation of team
  - ▶ Increases trust of customer

Evolutionary Development



- Successive Versions Model, Exploratory Evolutionary Development
- Aims at fast product implementation
- Delivery if product corresponds to subjective requirements
- Monolithic view hinders reuse
- Measuring progress difficult

Spiral Model (Risk-Driven)

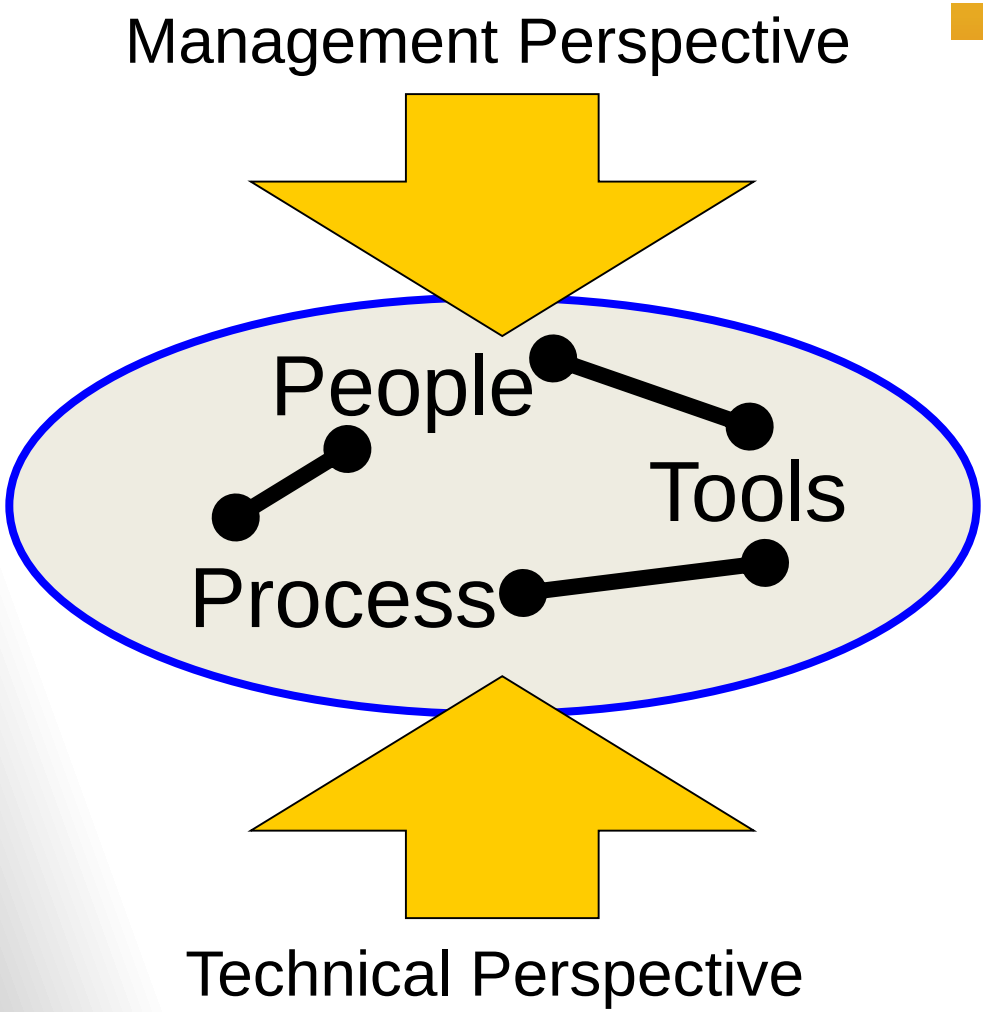


Source: Barry Boehm's spiral model
Computer 1988(5) pp. 61-72 – © 1988 IEEE

Spiral Model Template

Project, Date, …	
Objectives	Improve O1 and O2
Constraints	Within t, costs, …
Alternatives	Buy A1
Risks	Integration of X
Risk resolution	Develop Prototype, Product Survey
Results	Prototype works, flexible with A3
Plans	Develop Product and integrate A3
Commitment	Fund further 6 month

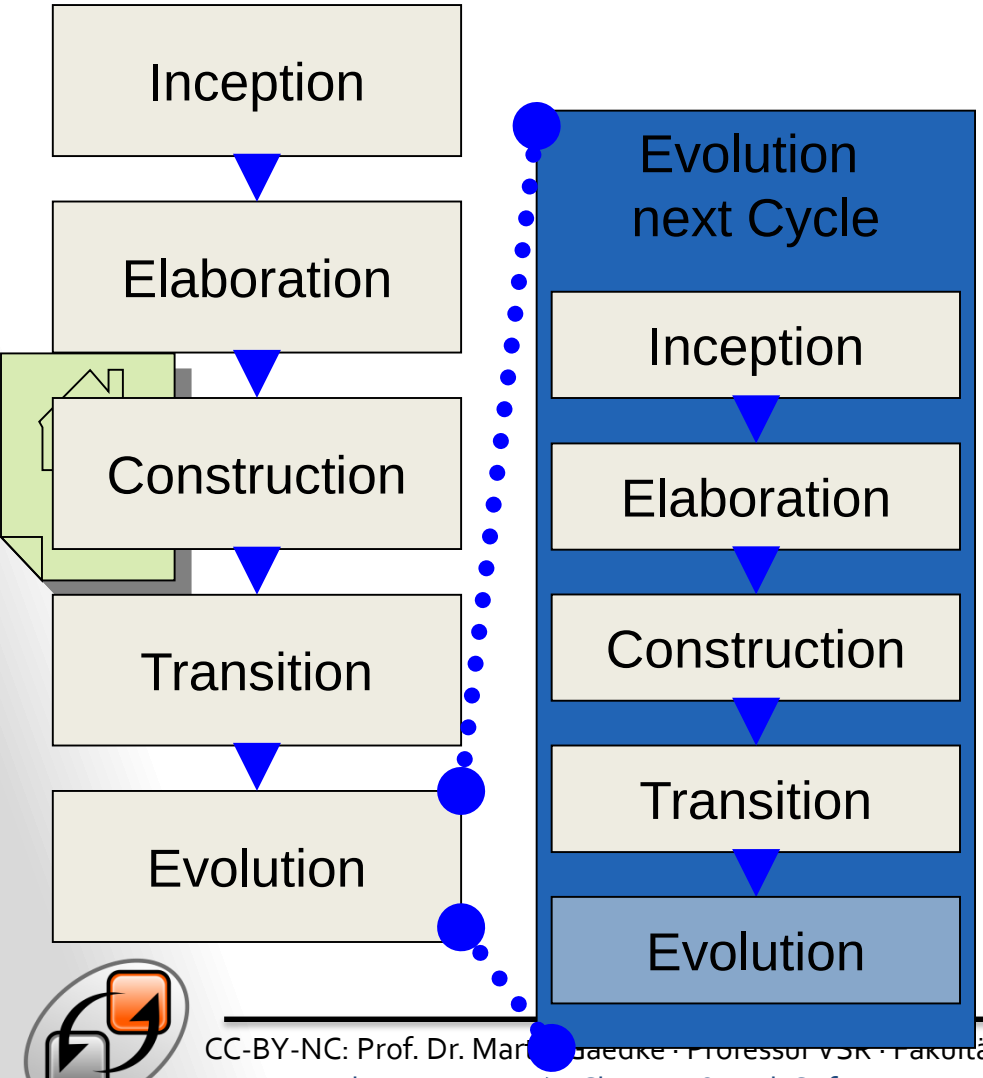
Rational Unified Process®



- RUP
  - Unified Software Development Process
  - Jacobson, Booch und Rumbaugh

Source: Philippe Kruchten, *A Rational Development Process*, Crosstalk, 9(7), July 1996

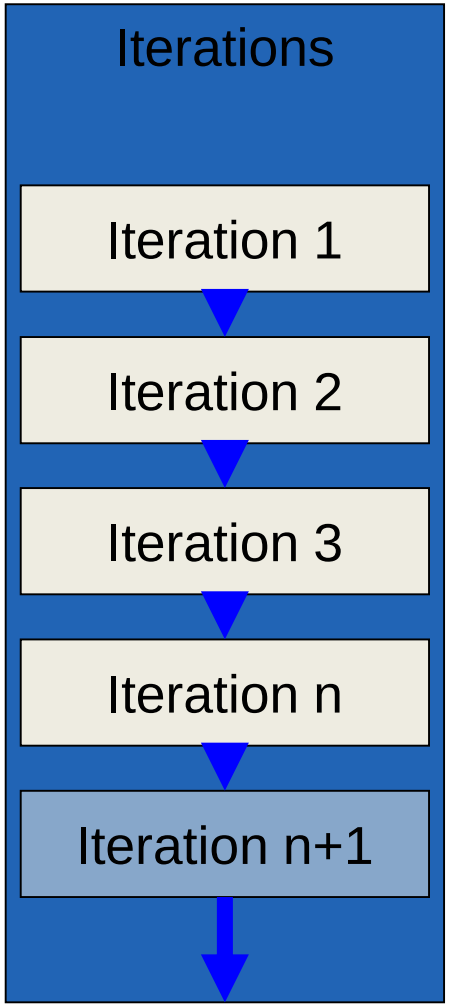
Management Perspective



- Development Life *Cycle*
  - Inception:
    - vision
    - business case
    - scope of the project
  - Elaboration: Planning
  - Construction
  - Transition: Deliver, Training
- Evolution
  - Life of software after initial development cycle

Source: Philippe Kruchten, *A Rational Development Process*, Crosstalk, 9(7), July 1996

Technical Perspective



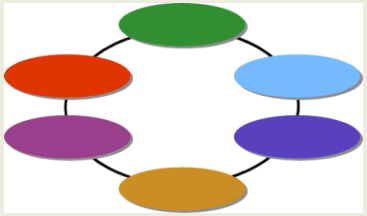
- Iteration Activities
  - Planning
  - Analysis
  - Design
  - Implementation
  - Testing
- Iteration is intellectual activity – not a stage for certain task

Source: Philippe Kruchten, *A Rational Development Process*, Crosstalk, 9(7), July 1996

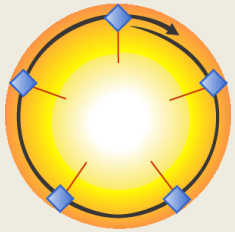
Key MSF Components (MSF v3)

Models

Team Model

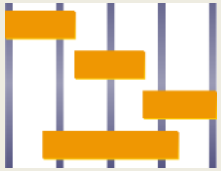


Process Model



Disciplines

Project Management Discipline

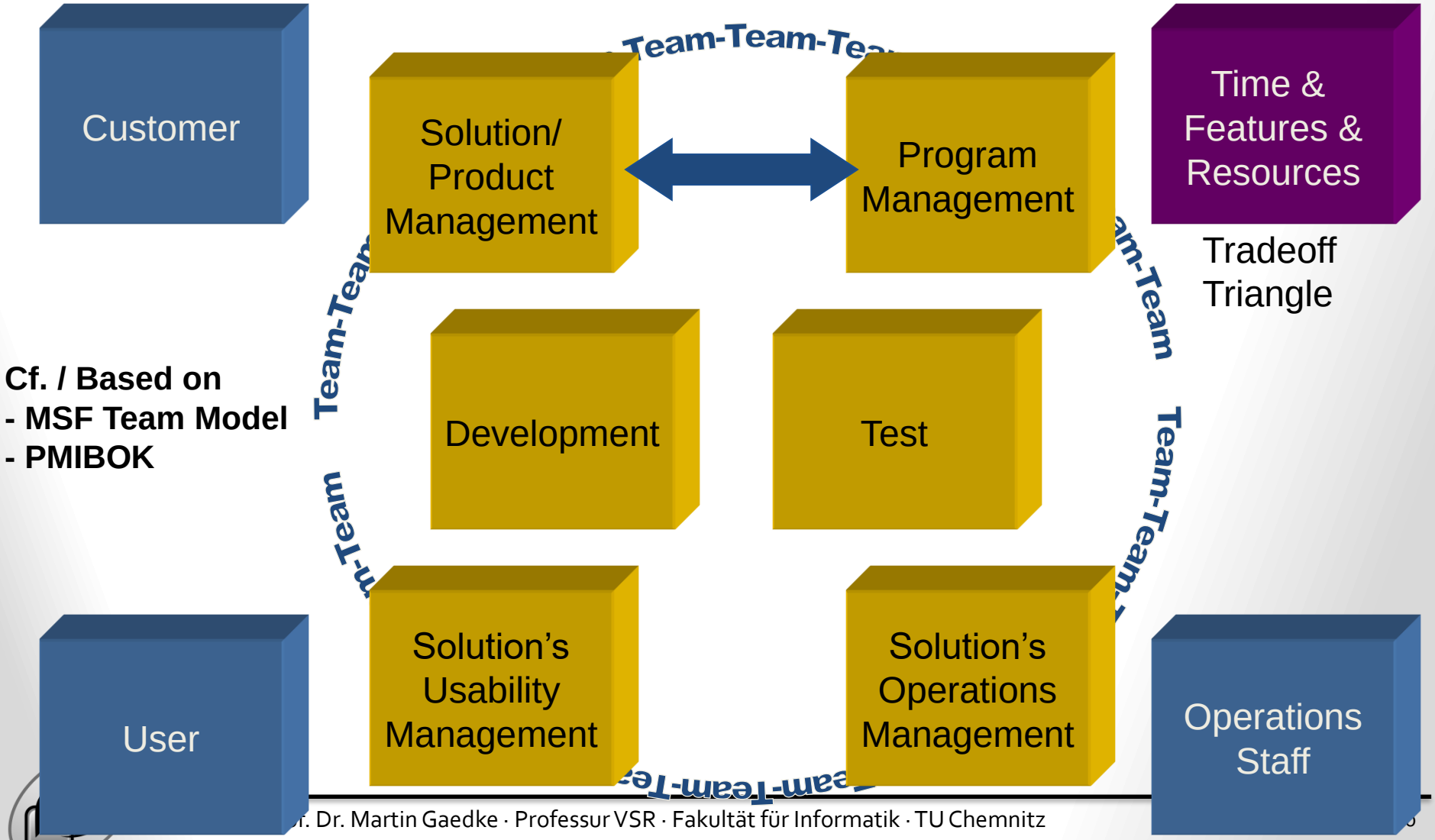


Risk Management Discipline

Readiness Management Discipline

Source: http://www.microsoft.com/technet/itsolutions/techguide/msf/default.mspx

MSF & Typical Team Approach



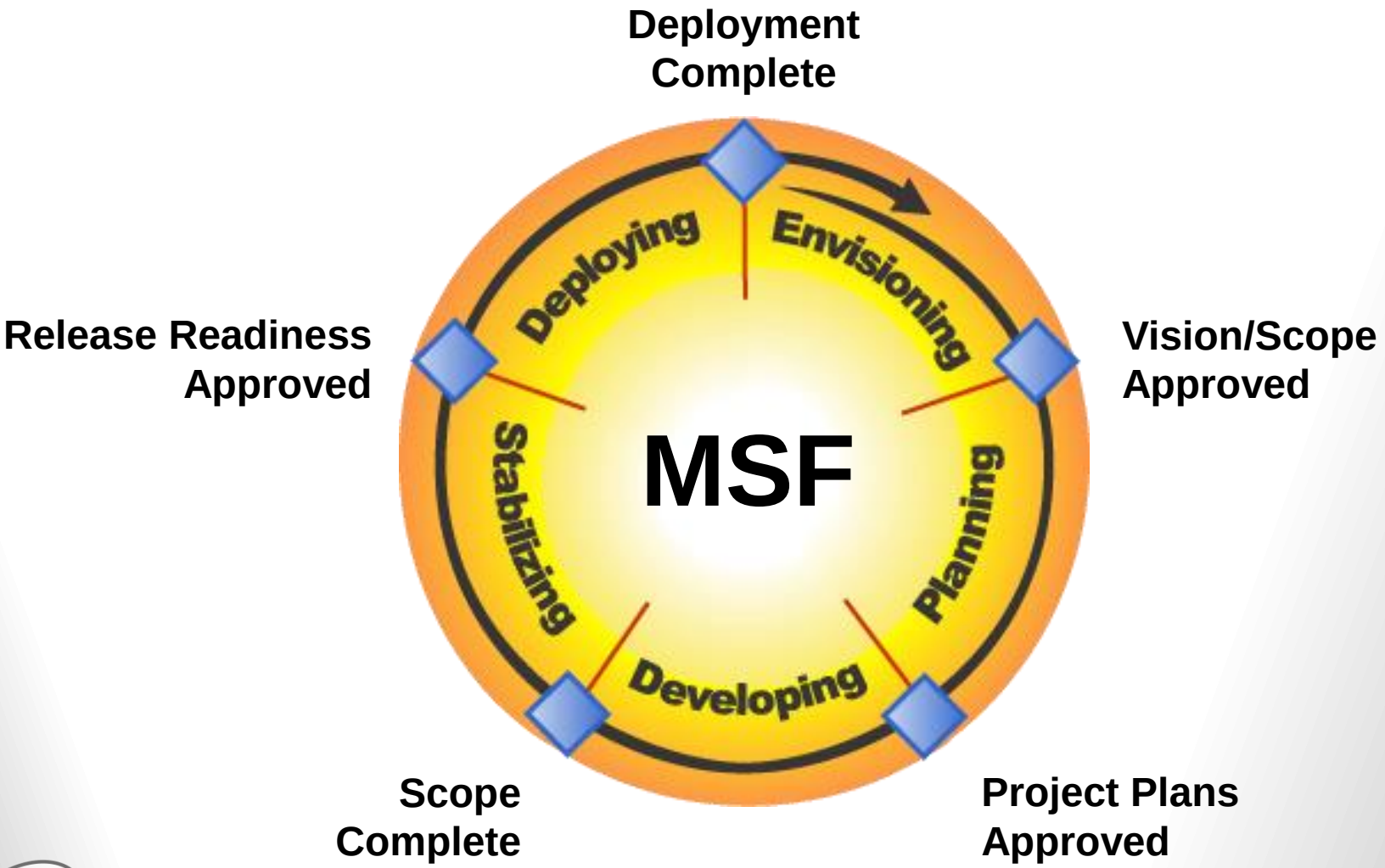
Cf. / Based on
- MSF Team Model
- PMIBOK

Scaling MSF Team-Model

- You can combine some roles to teams as small as 3 people
  - ▶ Do not combine some (like Product and Program Manager, or anything with Developer)
- You can scale the teams by using two general methods

- Functional Teams
  - ▶ Many people for one role
- Feature Teams
  - ▶ Sub-teams for each feature

MSF Process Model (MSF v3)



Design Process Overview (MSF v3)

Conceptual Design

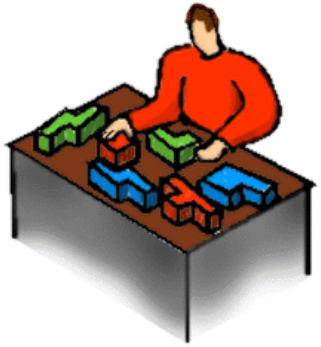
Scenarios

Logical Design

Objects and Services, User Interface, and Logical Database

Physical Design

Components, User Interface, and Physical Database

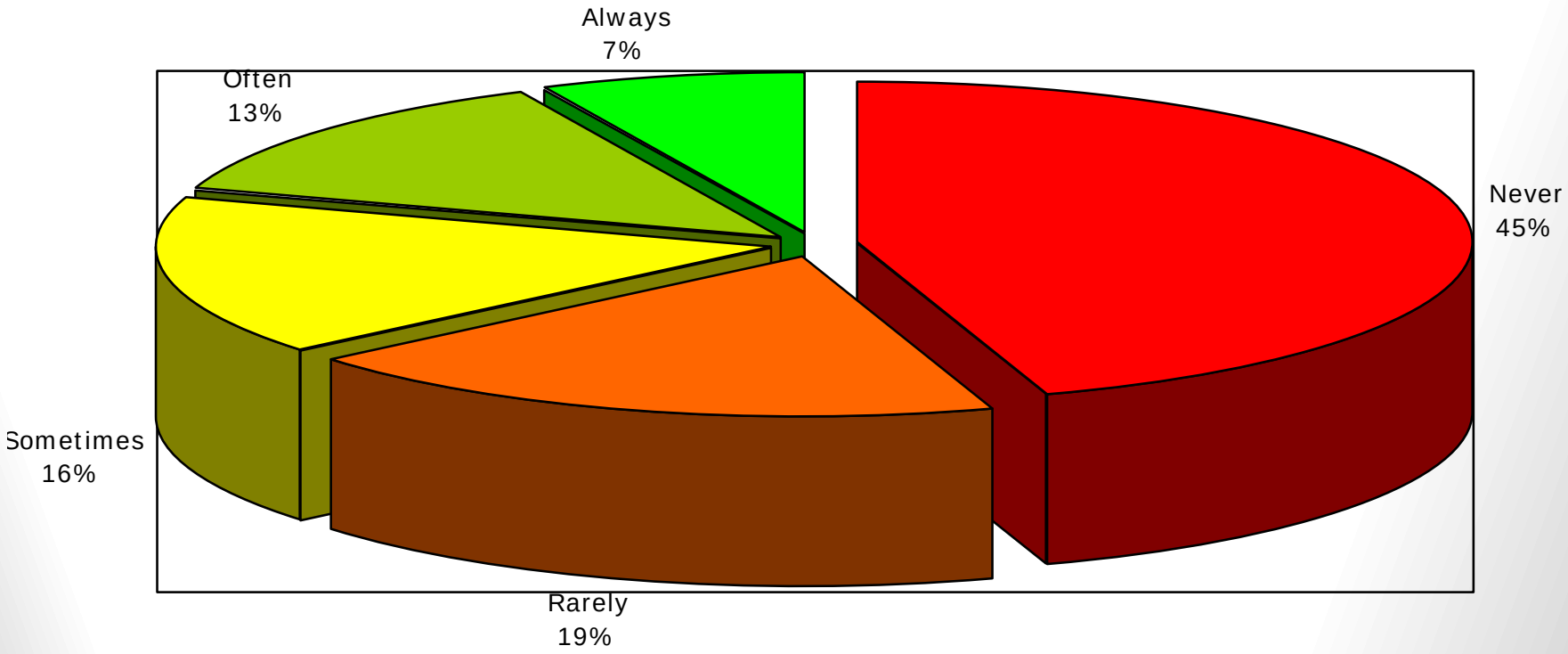


More on Process Models

- OPEN Process
  - OPEN Consortium
  - Contract-oriented stages
  - Coordination with other processes
- Web Engineering Process Models
  - Hot research topic – no standards yet
  - In most cases: derived from Spiral and focus on object oriented/based design models
- Many others are available – in most cases modifications of the Process Models presented here

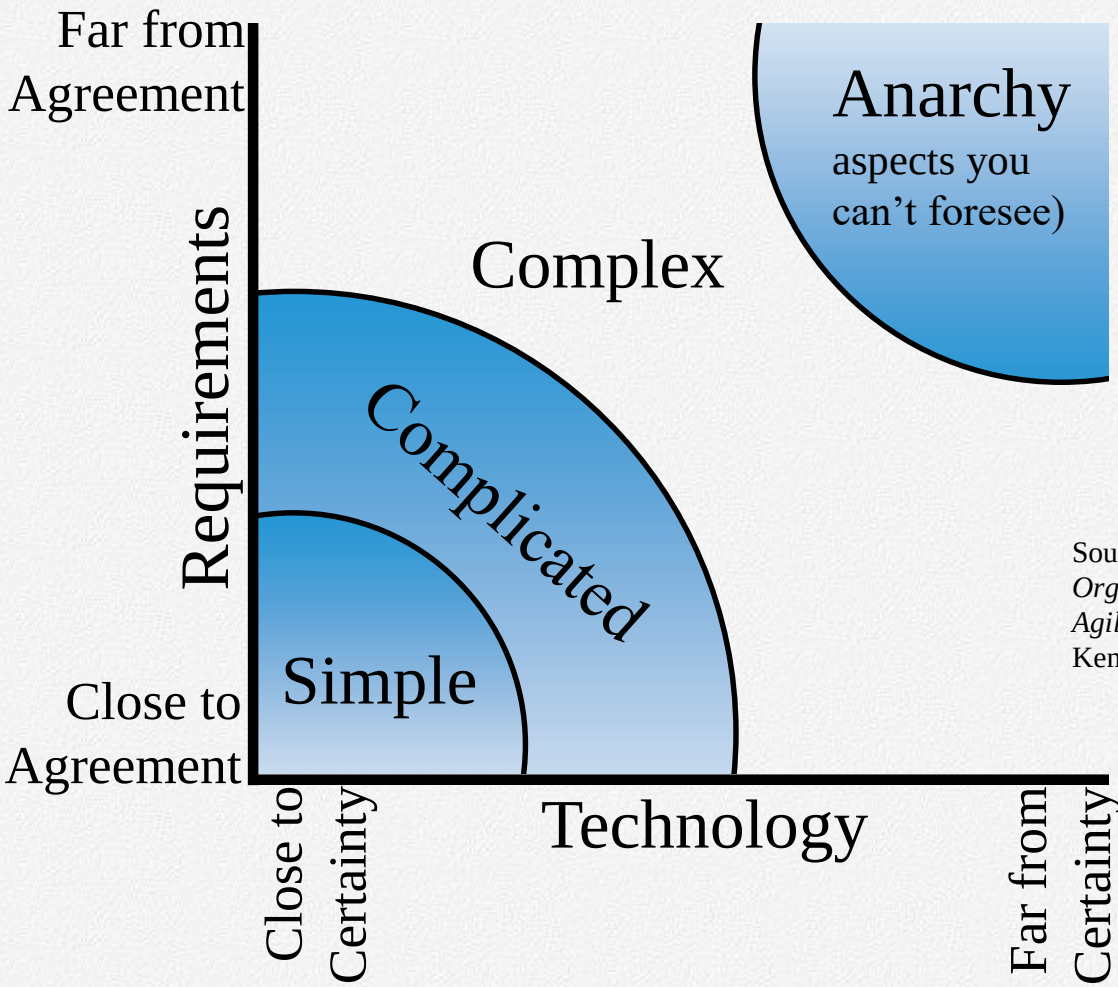
Planning: The Cost of Traditional BRUF

"Successful" Projects Still Have Significant Waste



Source: Jim Johnson of the Standish Group, Keynote Speech XP

We focus on Complex Problems

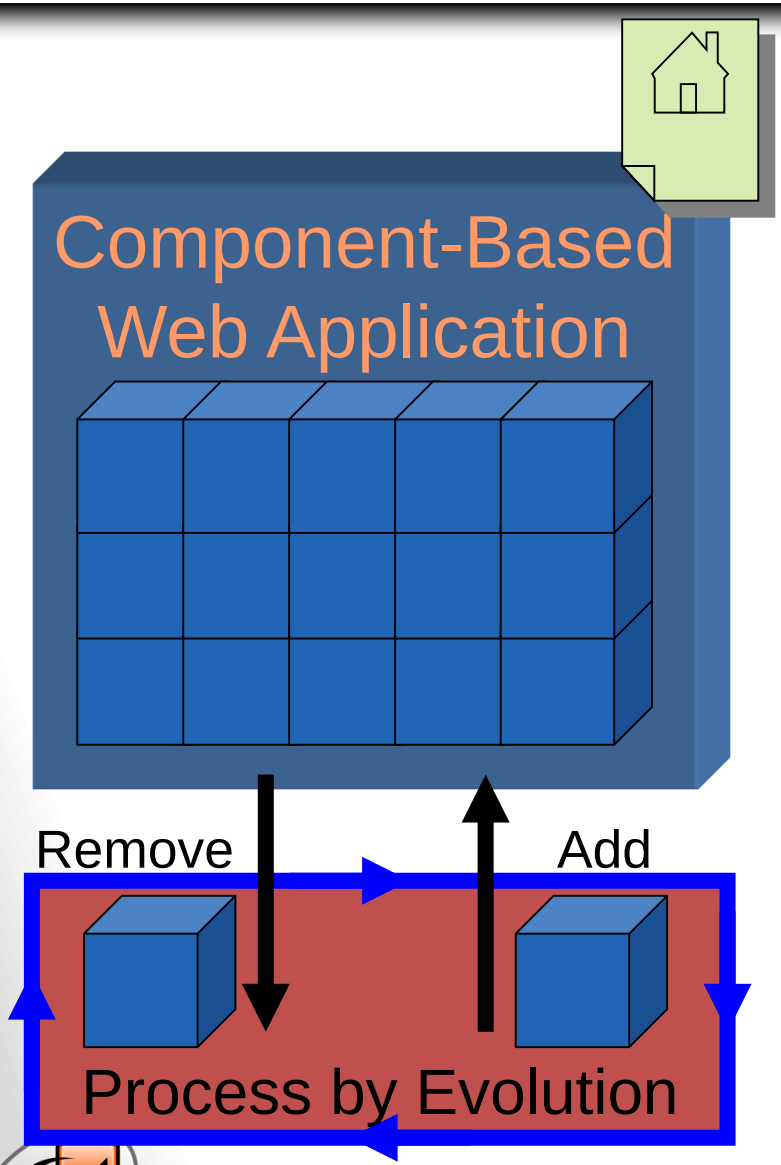


Source: *Strategic Management and Organizational Dynamics* by Ralph Stacey in *Agile Software Development with Scrum* by Ken Schwaber and Mike Beedle.

Idea: Agile Processes

- Reaction to the "bureaucratic" process models
  - Lightweight methodologies (now agile methodologies)
  - Try to answer Too much process vs. no process
- Apply an iterative and evolutionary approach to development
- Examples
  - Reuse-oriented approaches, e.g. WebComposition
  - Scrum
  - Kanban

Reuse-Oriented Approaches



- Web Engineering in context of Reuse
- Product is assembly from **reusable components**
  - Idea: All needed Components exist
- These Approaches focus on being **agile** in the context of:
  - **Producer Reuse**
  - **Consumer Reuse**

Model for Producer Reuse

- Develop (**Components**) **for Reuse**
- Domain Engineering
  - Process Model for Production of Domain Components
- Process Model
  - Domain Analysis
  - Develop Components
  - Deploy Components in Reuse-Repository
  - Make them available via Registries

Model for Consumer Reuse

- Develop **with Reuse** (of *Components*)
- Orthogonal Process Model
- Process Model
  - Accessing
  - Understanding
  - Adapting