anderScore
trust in competence

Achieving usable security in a project driven industry

2018-03-19, J.Lühr

Agenda

About Me

anderScore trust in competence

Jan Lühr

- B. Sc., Computer Science
- Senior Software Engineer & Architect
- anderScore since 2007
- Focus
  - Software Development
  - Pragmatic Architect
  - Build- and Deployment engineering
  - Network- and Security Techniques
  - IT-Trainer
  - Java, JavaScript, Ruby
- Misc.

CRYPTO PARTY

Introducing the company I work for

1. WHAT ARE WE DOING AT ANDERSCORE?

1. anderScore: Facts & Focus

anderScore trust in competence

Facts

- Founded: 2006
- Owner-managed (GmbH)
- Office: Cologne, Germany
- Employees
  - ~ 20 internal
  - ~ 70 external (contacts; freelancers)
- > 10 partner companies
- > 20 major customers (enterprises)

Focus

- Customer specific software development
- Pragmatic, agile approach
- Mobile apps & web applications
- Technique: Java
- Business process automation
- Software and systems migration
- Software and security audits

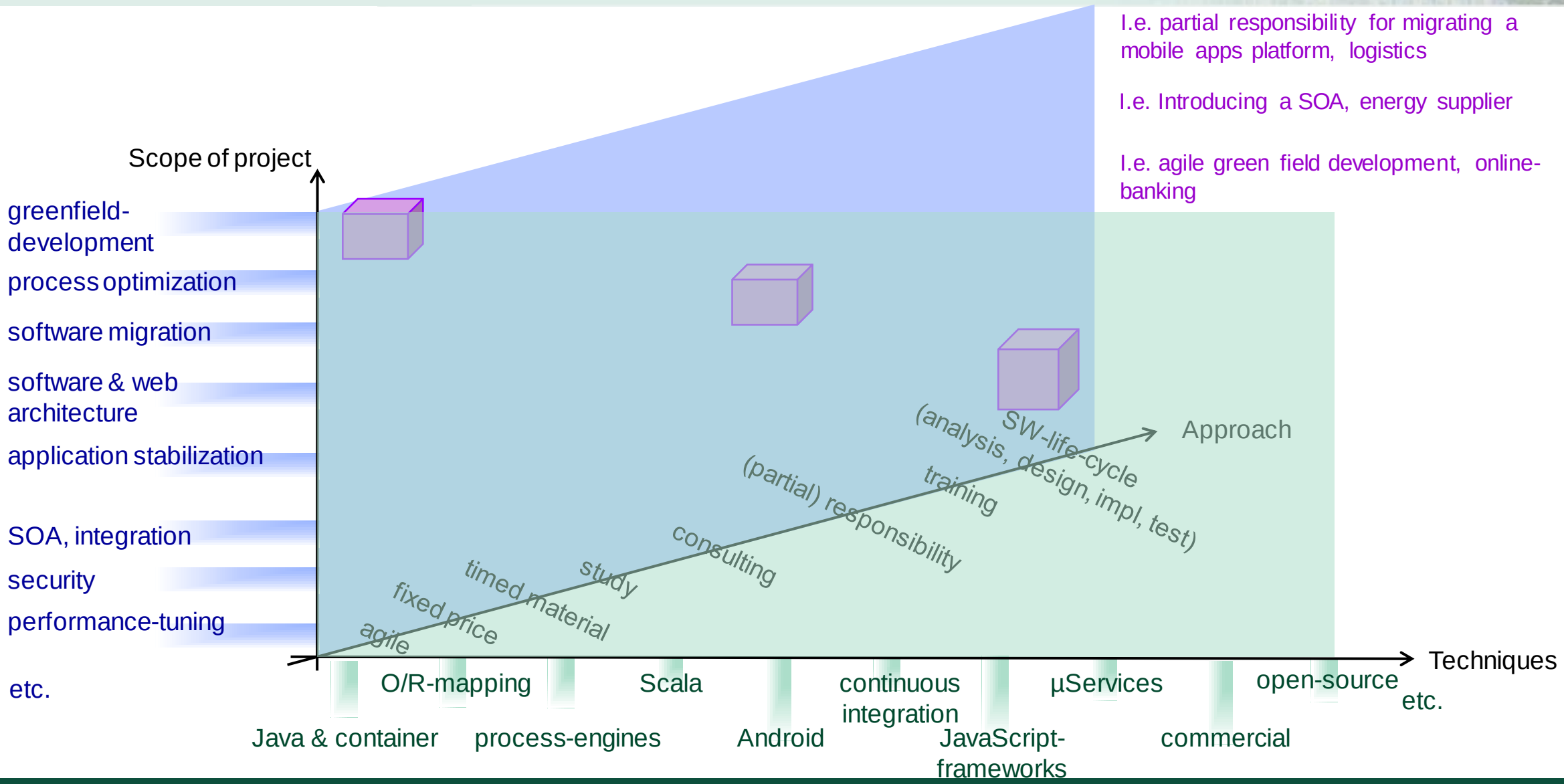
1. Our Service Offering

anderScore trust in competence

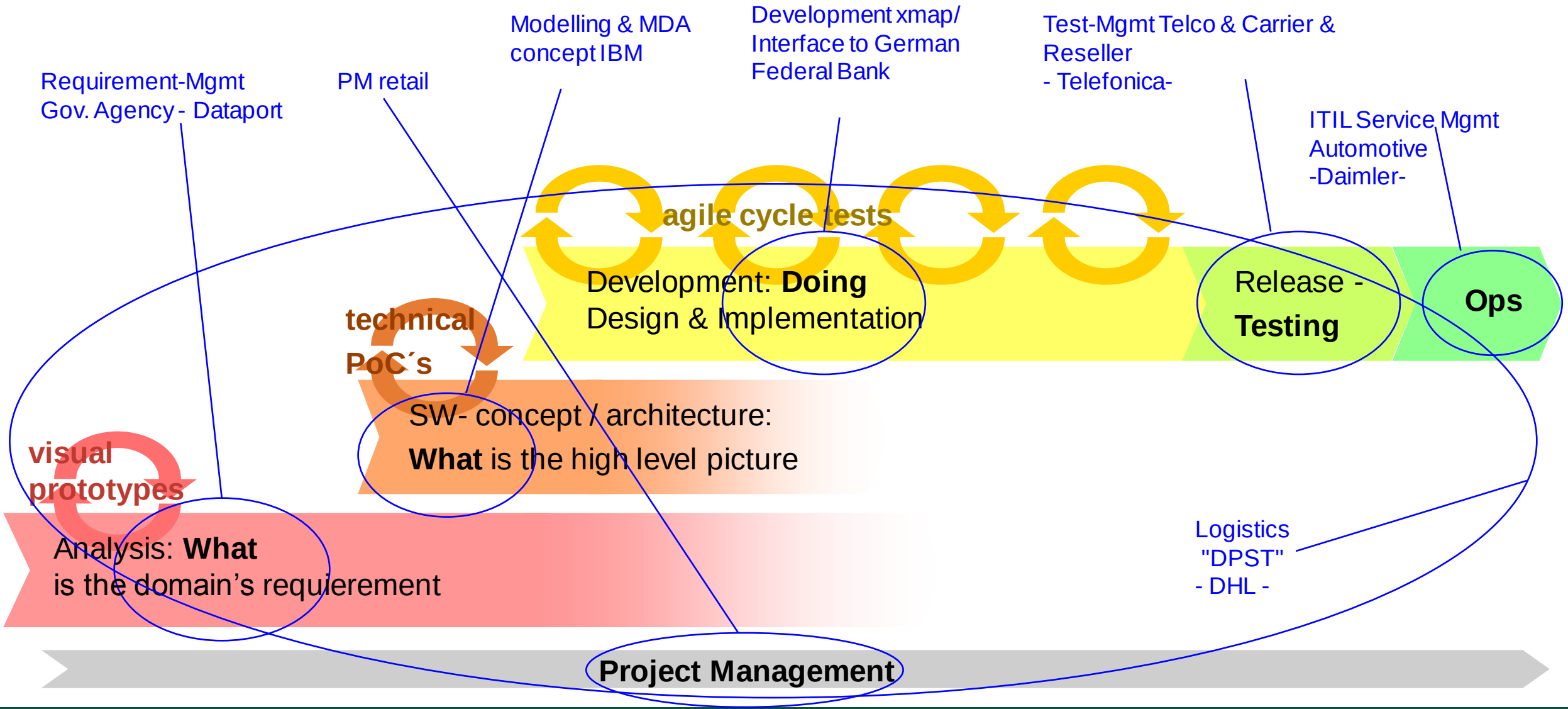
I.e. partial responsibility for migrating a mobile apps platform, logistics

I.e. Introducing a SOA, energy supplier

I.e. agile green field development, online-banking



1. Service Offering: Software Life Cycle



How do we do it - our methodology

2. PROJECT DRIVEN SOFTWARE ENGINEERING

2. Software Engineering Project 101

anderScore trust in competence

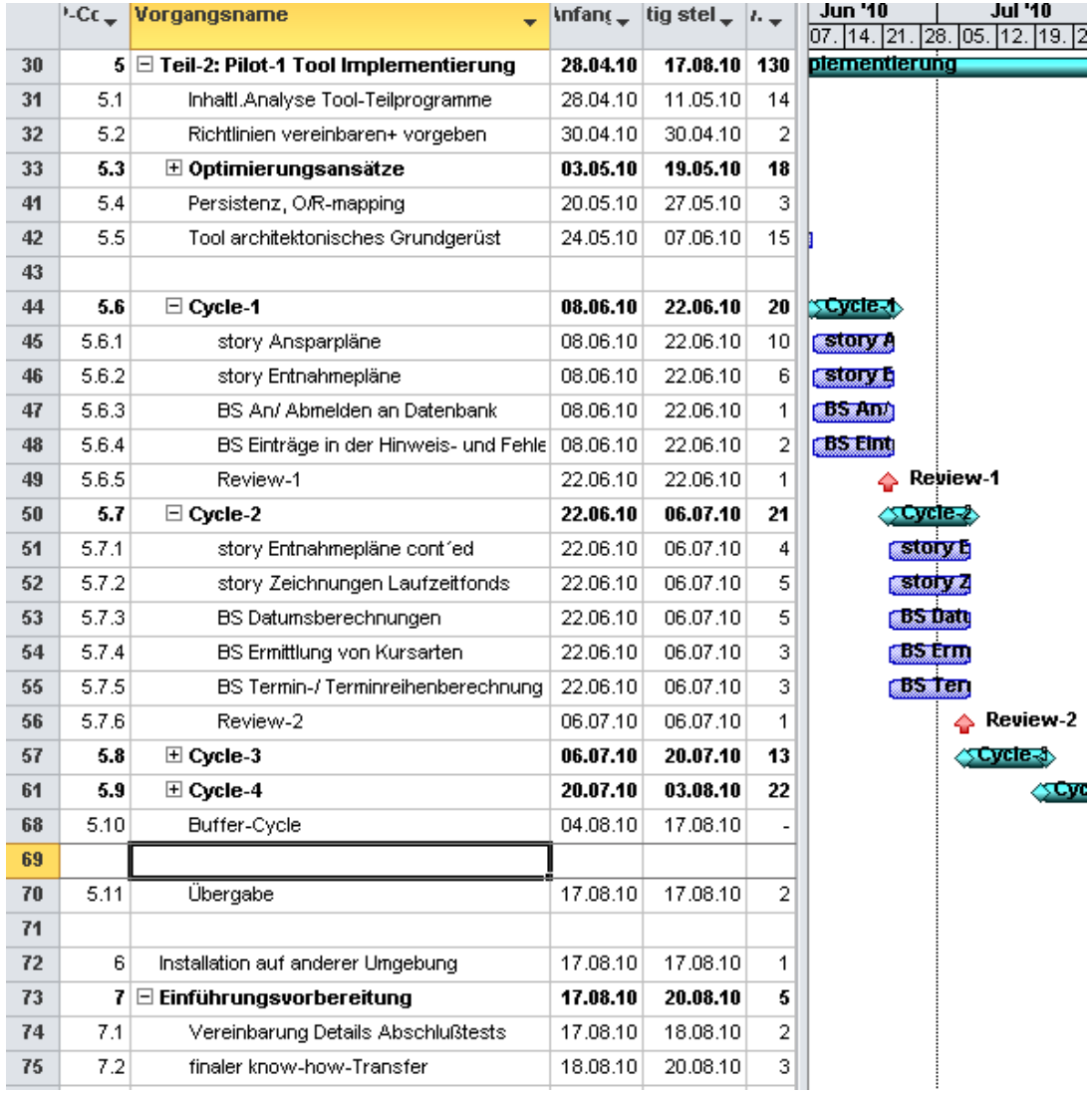
1. We‘re **not selling a box** / software product / platform

2. Software Engineers: hired to **implement a business case**.
   Domain people know what’s needed
   – but can hardly specify it on their own

3. Projects have **end-dates**, a **predefined scope**: 2-12 people working as an ad-hoc team for 3-36 months (typically)

4. Functionality: given by **formal requirements** - split into tasks (0.5 – 5 person-days),

5. A **plan** defines deadlines, task order, milestones, cycles, …



	-Cc	Vorgangsname	Anfang	tig stel	
30	5	**Teil-2: Pilot-1 Tool Implementierung**	**28.04.10**	**17.08.10**	**130**
31	5.1	Inhaltl.Analyse Tool-Teilprogramme	28.04.10	11.05.10	14
32	5.2	Richtlinien vereinbaren+ vorgeben	30.04.10	30.04.10	2
33	**5.3**	**Optimierungsansätze**	**03.05.10**	**19.05.10**	**18**
41	5.4	Persistenz, O/R-mapping	20.05.10	27.05.10	3
42	5.5	Tool architektonisches Grundgerüst	24.05.10	07.06.10	15
43					
44	**5.6**	**Cycle-1**	**08.06.10**	**22.06.10**	**20**
45	5.6.1	story Ansparpläne	08.06.10	22.06.10	10
46	5.6.2	story Entnahmepläne	08.06.10	22.06.10	6
47	5.6.3	BS An/ Abmelden an Datenbank	08.06.10	22.06.10	1
48	5.6.4	BS Einträge in der Hinweis- und Fehle	08.06.10	22.06.10	2
49	5.6.5	Review-1	22.06.10	22.06.10	1
50	**5.7**	**Cycle-2**	**22.06.10**	**06.07.10**	**21**
51	5.7.1	story Entnahmepläne cont´ed	22.06.10	06.07.10	4
52	5.7.2	story Zeichnungen Laufzeitfonds	22.06.10	06.07.10	5
53	5.7.3	BS Datumsberechnungen	22.06.10	06.07.10	5
54	5.7.4	BS Ermittlung von Kursarten	22.06.10	06.07.10	3
55	5.7.5	BS Termin-/ Terminreihenberechnung	22.06.10	06.07.10	3
56	5.7.6	Review-2	06.07.10	06.07.10	1
57	**5.8**	**Cycle-3**	**06.07.10**	**20.07.10**	**13**
61	**5.9**	**Cycle-4**	**20.07.10**	**03.08.10**	**22**
68	5.10	Buffer-Cycle	04.08.10	17.08.10	-
69					
70	5.11	Übergabe	17.08.10	17.08.10	2
71					
72	6	Installation auf anderer Umgebung	17.08.10	17.08.10	1
73	**7**	**Einführungsvorbereitung**	**17.08.10**	**20.08.10**	**5**
74	7.1	Vereinbarung Details Abschlußtests	17.08.10	18.08.10	2
75	7.2	finaler know-how-Transfer	18.08.10	20.08.10	3

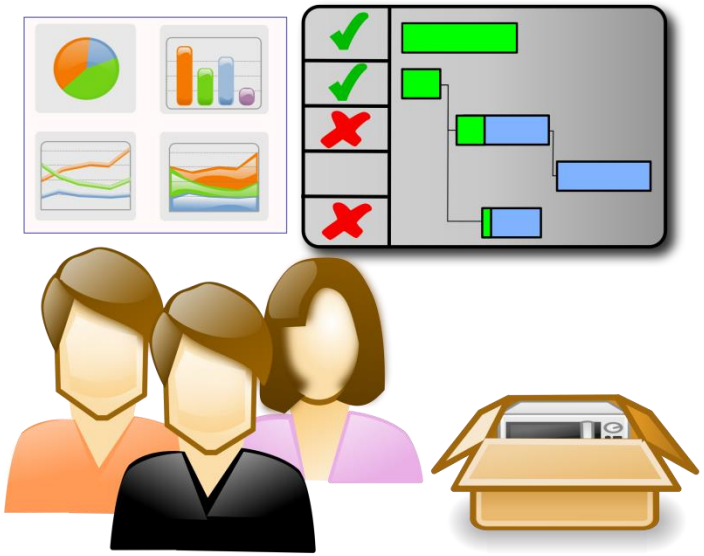
2. SWE Projects Characteristics: Observations & Lessons Learned

anderScore trust in competence

1. **Requirements change** over time
   - Iterative re-definition is essential
   - Process needed (cycles, meetings)

2. **Users must participate actively**
   - Design, Reviews, Tests
   - Key to understanding requirements
   - Needed for user acceptance
   - User-research: low hanging fruit

3. **Ad-hoc teams** require **methodology**
   - Formal proceedings (Scrum, Crystal, Kanban, Lean-IT, …)
   - Security: out-of-scope (usually)
   - Roles (business analyst, developer, architect, tester, …)



2. SWE Project: Roles…

Problems, pitfalls & views I've encounter in software engineering projects

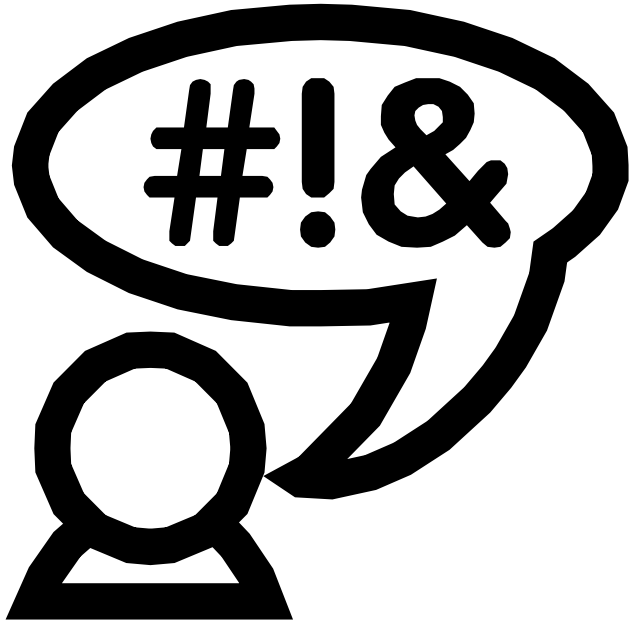
3. SECURITY, USABILITY

3. Definitions

- **Security** (BSI – Bundesamt für Sicherheit in der Informationstechnik):
  Confidentiality, availability and **integrity**

- **Usability** (ISO 9241):
  The **effectiveness**, **efficiency** and **satisfaction**
  with which specified **users** achieve specified **goals**
  in particular **environments**.

- **A system is usable secure** if expected users (~ Whitten, Tygar):
  - Are reliably **made aware** of the security tasks they [..] perform
  - Are able to **figure out how to** successfully perform those [..]
  - **Don't make dangerous errors**
  - Are **sufficiently comfortable** with the interface to **continue using** it.



3. Secure Systems – common understanding

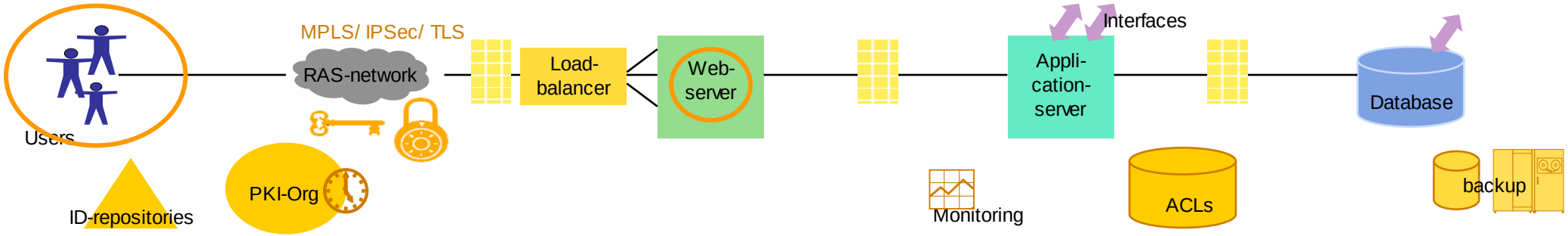
Huge diversity of different aspects:

- ? Signatures, certificates, PKI
- ? IAM, user management, authentication / PKI, authorization
- ? transfer encryption / PKI
- ? timestamping PKI
- ? VPN´s
- ? remote access / reverse-proxy-handling/ firewalls / zones (DMZ´s)/ intrusion-detection/ prevention
- ? Reliability (HW-redundency, automatic failover, LB, backup & restore, monitoring)
- ? Physical security: constructional, organizational, personnel security layers

➔ **Infrastructure driven**

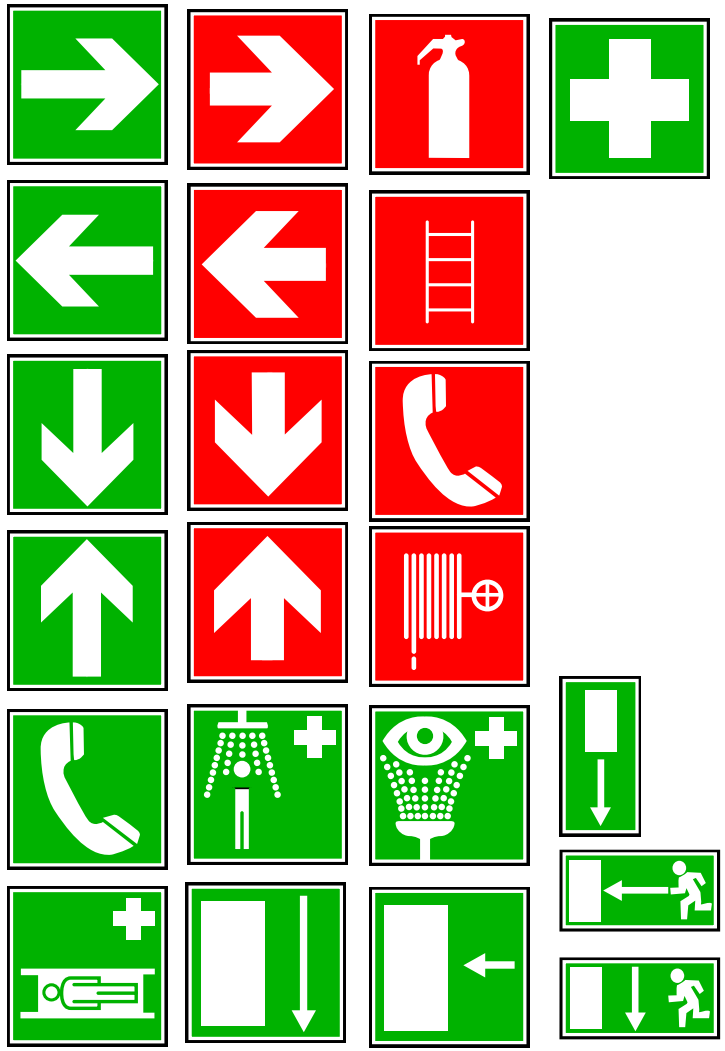
Goals (BSI GS):

- **Availability**
- **Integrity**
- **Confidentiality**



3. Security in SWE – Problematic approach

- Demand: “Has to be secure”
  - Compliance from the security (sometimes: penetration-testing)
  - No detailed non-functional requirements (usually)
- Business Analyst’s / Project managers' perspective:
  - Overarching focus on domain & functional requirements (Scrum, …)
  - *”Security must not hinder business operations“*
- Developer’s / Architect’s perspective:
  - Automate pen-testing, dependency-checking, etc.
  - Include tools in build / delivery pipelines (continuous integration)
- **Consequences**
  - **No plus** from achieving better security (more than compliance)
  - **Usability aspects** are not taken into account



3. User Experience in SWE-Projects

anderScore trust in competence

- Cooperate design / identity:
  - Determines look & feel
  - Provided in advance
  - Re-Useable components (libraries, tools, …)
- User experience people: not included in teams
  - unless a domain requirement exists (rare; competition, end-user acceptance,…)
- Needed in migration projects
  - Old system used efficiently – user got used to it
  - Ergonomics needed for acceptance
- **But: User participation is easy (often)**

What do we do at anderScore to build secure systems?

4. ACHIEVING USABLE SECURITY

4. Usable security enterprise projects

anderScore trust in competence

A multi-dimensional challenge

1. **Methodology dimension**
   How to proceed in a project?
   What roles / steps / procedures are needed?
   Usable security compliance / requirement?

2. **Classical usability / end-user dimension**
   Do end-users securely perform all tasks of a business process? Are security functions usable?

3. **Software engineering dimension**
   Software-Engineers as end-user:
   Usability of APIs / libraries

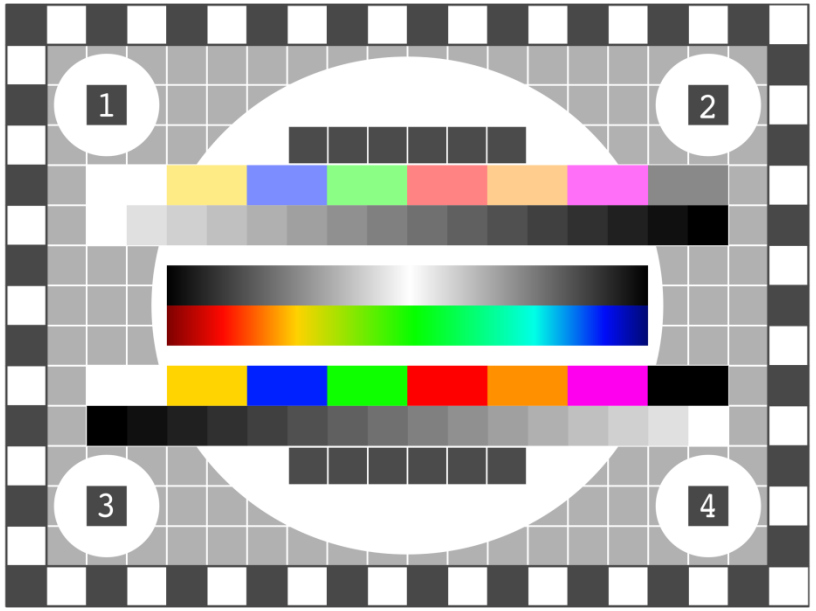
4. **Social-Engineering dimension**
   How to support users during an attack?
   How to design and provide an resilient user experience?
   How do different business processes influence each other?

4.1 Project Methodology

- Challenge
  - How to proceed in a project?
  - What roles / steps / procedures are needed?
  - Usable security compliance / requirement?

- An iterative approach is suitable
  - Users participate in cycle meetings (review)
  - Feedback: Early (i.e. mocks) & often
  - Usable security problems: noticed immediately

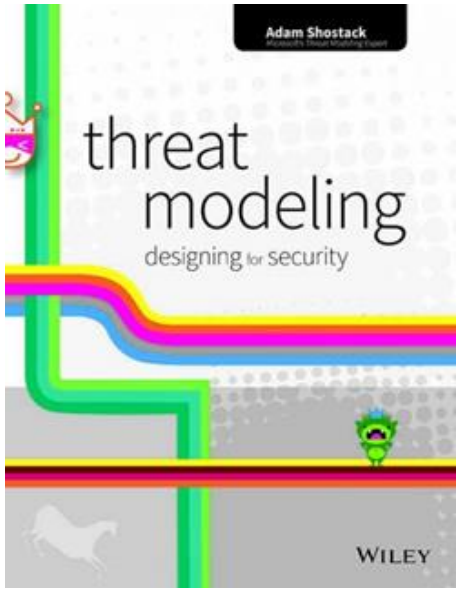
- User Acceptance Testing (UAT)
  - Verifying, that the system works for the user
  - Complete setup, pre-production environment
  - Verification of functional and non-functional requirements
  - User-errors can be detected



4.2 Classical Usability / User Experience Dimension

anderScore trust in competence

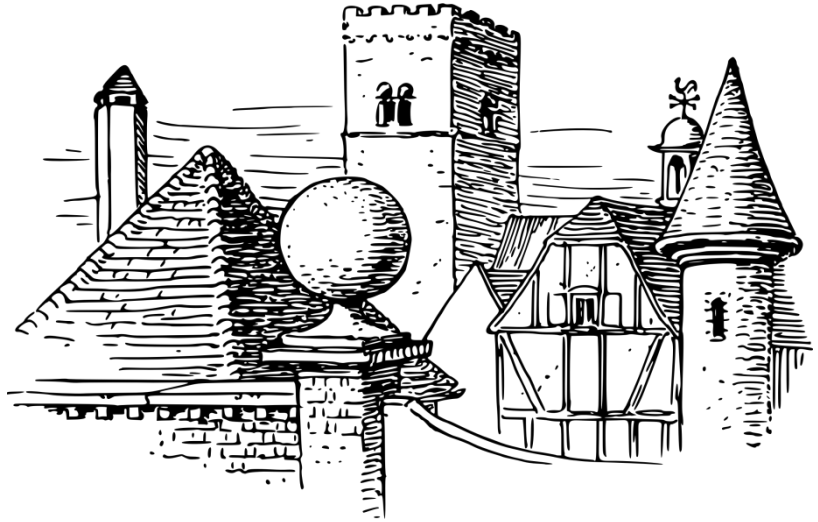
- Challenge
  - Do end-users securely perform all tasks of a business process?
  - Are security functions usable?
- Including a product discovery phase
  - Cognitive Walkthrough & Think aloud
  - Including security experts: Check users' perception
- Utilizing Design personas
  - Threat modeling
- Lean Usability Testing
  - Less than 5 users, often & early



4.3 Software Engineering Dimension

- Architecture
  - Definition of tools / techniques / components / source code structure framing a system
  - Fitness: Effort for implementing a new requirement / fixing a bug

- Usable security – also: architectural fitness.
  - Aware of the security tasks?
  - How to successfully perform those [..]
  - Dangerous errors



- Mind user and goal (ISO 9241)
  - Trainings: realistic
  - Understands technical details
  - Goal: Implement functional requirements efficiently.

4.4. Social Engineering

- Social engineering:
  - Users‘ are made cooperative (stress, deception, …)
  - Problematic assumptions: exploitable (sender address, …)
  - https://www.youtube.com/watch?v=lc7scxvKQOo
- Challenge
  - How to support users during an attack?
  - How to design and provide an resilient user experience?
  - How do different business processes influence each other?
- In business processes: Scripted communication
  - Fixed set of questions and answers
  - Communication definable by flow chart
- Supporting the user – helpful user experience:
  - Provide compliant responses with legal backing
  - Wire the flow-chart
  - Create a workflow minimizing stress (decoupling responsibilities / actions / etc.)


Von Nick Mathewson
Betreff [tor-talk] Tor 0.3.1.6-rc is released! 2017-09-05 16:36
An tor-talk@lists.torproject.org <tor-talk@lists.torproject.org>

Hi, all!

There's a new Tor release candidate available! The source is available from the "download" page on the website on the website, and packages should be available before long. The Tor Browser team expects to get a release out later this month.

This is a release candidate; please help find bugs in it! If we don't find any new critical problems, we'll be calling this release series "stable" soon.

- **Example**: Scanning Liquor at a grocery store
  - ID verification required
  - Screen shows youngest day of birth (no need to calculate)
  - Different beep: mandatory check

My personal view

5. RESEARCH ./. ENGINEERING QUESTIONS & OUTLOOK

5. The GnuPG case

- Is GnuPG / PGP usable?
- „*If it's "like PGP," it's wrong. PGP is our guide for what not to do.*"
  (Signal-Android Contributor Guidelines)
- https://www.youtube.com/watch?v=aJPaR9Cfmk

"*I've traveled to a few countries in South America and also Mexico to do PGP trainings [..], threat modeling, [..] for both activists and journalists. If people need to protect their email communications, it's the best tool out there. I think that if you understand how to use [..] it's solid and it's a good way to communicate and to be confident that your communications are private.*" (Lisa Wright, EFF)

- PGP / GnuPG's usability is under dispute (Johnny papers) – tool: user-studies
- Results: Not transferable to a corporate context:
  - Users differ – education, background, trainings
  - ISO 9241: specified *users* achieve specified *goals*
  - **Statistical soundness & significance: useless**

5. Research in Usable Security – How does it help me?

- Very helpful for understanding the problem / challenge
  (i.e. Whitten / Tygar – Why Johnny can‘t encrypt)

- User-studies
  - Hard to apply, if users and context match to some extent
  - Not applicable, if specification is vague / non-transferable
    (i.e. Amazon‘s mechanical turk, college billboard recruiting)

- Helpful
  - Surveys on tools and methods
    (i.e. S.Fahl, et.al. “Rethinking SSL development in an appified world”)
  - Qualitative / non-quantitative studies

5. Open Questions from a Software Engineer

anderScore trust in competence

1. What is the economic value of usable security?
2. How to proceed in a project to achieve usable security?
   - Interdisciplinary teams?
   - I.e. constructing a scrum like approach?
3. How to require usable security?
   (non-functional requirements, compliance)
4. Post mortem analysis
   - Case study: Contribution of usability to a specific incident?
   - Methodology: How to evaluate the contribution in a structured way?
5. Social engineering
   - How to secure scripted communication in a usable way?
   - How do perform security features under stress / an attack?

Summary

1. Enterprise software (online banking,…) is engineered by small teams using a defined methodology.

2. Security: important for almost any enterprise
   - Integrated into software systems – it‘s not limited to security software such as PGP.
   - Strong focus on compliance and infrastructure.

3. Four dimensions (ux, methodology, swe, social engineering)
   - Project methodology and social engineering underrepresented in research?
   - Current methodology (scrum / cycles) support basic usable security

4. Usable security: Not just about *putting-lipstick-on-a-pig* anymore*:*
   - Promising strategy for hardening against social engineering attacks
   - Design strategy: Mitigating issued not addressed by classical security patterns (i.e. sign / encrypt / verify)

Thanks for your time

QUESTIONS?