Was kommt nach „Security by Design“?

Chancen der Partizipation im Software Engineering

Sven Türpe, Andreas Poller

Fraunhofer für Sichere Informationstechnologie, Darmstadt

Fraunhofer SIT

Thesen

1. Das Konzept „Security by Design“ …
   - …sammelt deduktiv-rationalistische Ansichten
   - …missversteht damit das Konzept von „Design“,
   - …ignoriert Software-Emergenzprozesse, und…
   - …Software-Entwicklung als kollaborative, interaktive Praxis in Institutionen.

Thesen

2. Wir benötigen „Security Design Research“, das heißt, z.B., …

- …die Rolle von „Sicherheit“ und „Sicherheitsingenieuren“ in der Softwareentwicklung reflektieren,
- …Ursachenforschung für Erfolg oder Misserfolg von „Security Designs“ oder „Security-Nicht-Designs“ betreiben, und
- …ein Methodik von „Security Design Thinking“ entwickeln.

Security by Design - Eine Bestandsaufnahme

Security by Design

Secure by design

From Wikipedia, the free encyclopedia

This article includes a list of references, related reading or external links, **but its sources remain unclear because it lacks inline citations**. Please help to improve this article by introducing more precise citations.
(December 2010) (Learn how and when to remove this template message)

It has been suggested that this article be merged into *Defensive programming*. (Discuss) *Proposed since January 2014.*

Secure by design, in software engineering, means that the software has been designed from the ground up to be secure. Malicious practices are taken for granted and care is taken to minimize impact when a security vulnerability is discovered or on invalid user input.

Generally, designs that work well do not rely on being secret. It is not mandatory, but proper security usually means that everyone is allowed to know and understand the design *because it is secure*. This has the advantage that many people are looking at the code, and this improves the odds that any flaws will be found sooner (Linus' law). Of course, attackers can also obtain the code, which makes it easier for them to find vulnerabilities as well.

Also, it is very important that everything works with the least amount of privileges possible (principle of least privilege). For example, a Web server that runs as

Quelle: https://en.wikipedia.org/wiki/Secure_by_design

Security by Design

Was es sein soll:

- **Systematisch Softwaresicherheit in allen Teilen von Software-Lebenszyklen berücksichtigen** (Aberdeen-Group 2010; Forrester 2011; Viega and McGraw 2011)
- **Betonung auf langfristige ökonomische Vorteile und sozialen Vorteilen** (Davis 2006; Allen et al. 2012; CIO 2007; Kasal et al. 2011; Microsoft 2010; Tassey 2002; Bodden et al. 2014; Boehm 1981)

Security by Design

Zwei Grundströmungen

- Build Security In
  (Cigital, DHS, Microsoft)

- Betonung der systematischen Berücksichtigung von Sicherheit ab der Entwurfsphase einer Software

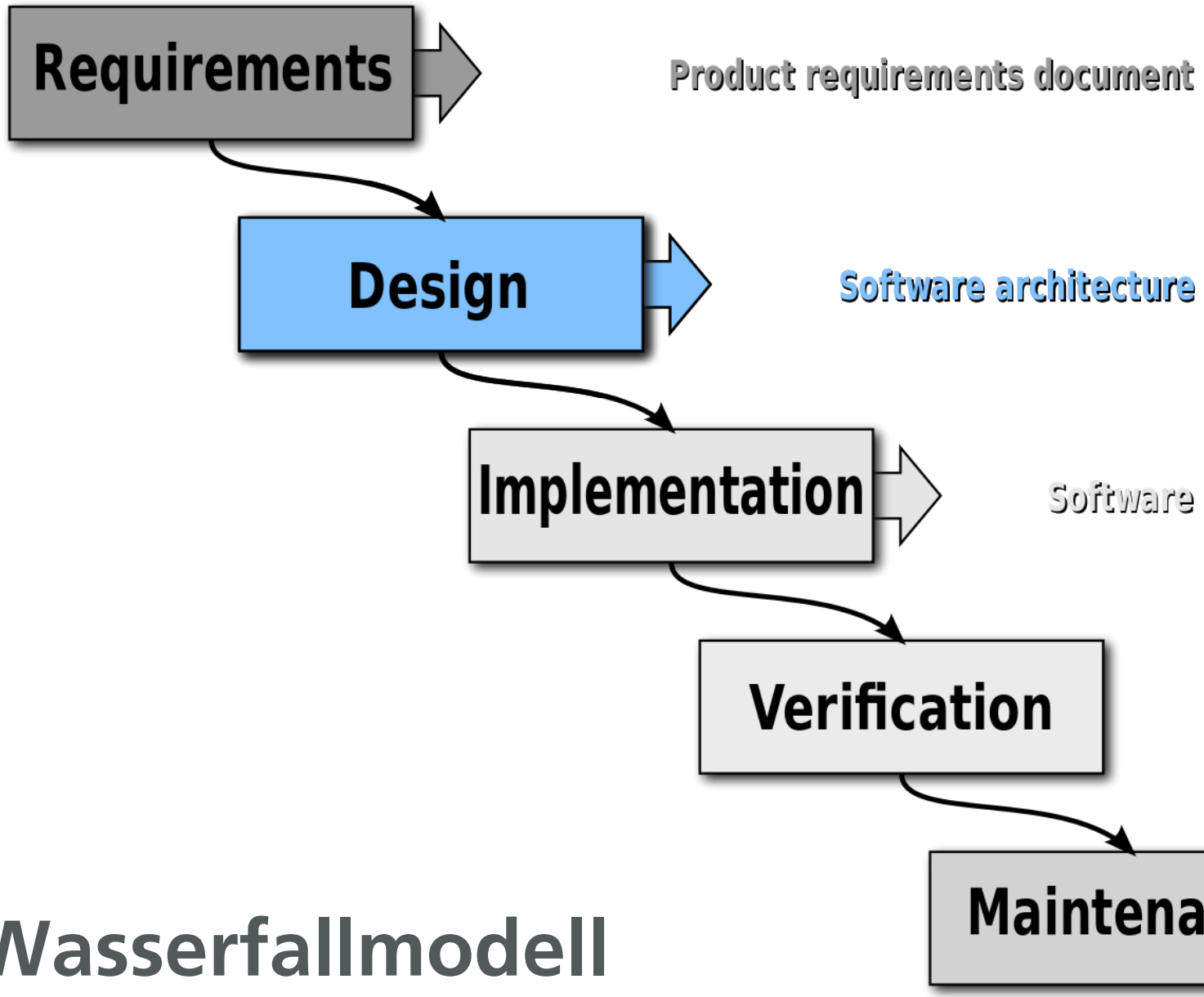
Security by Design

Was wir tatsächlich wissen:

- **Ökonomischer Vorteil,**
  und
  Praxis von Security by Design
  nicht umfassend empirisch untersucht.

Security by Design

- Meisten Studien sind Whitepaper oder technische Berichte (Aberdeen-Group 2010; Cigital 2003; IBM 2013), oder
- anektdotische Einzelschilderungen, oder
- Konzeptpapiere (Cavoukian and Chanliau 2013), oder
- eng fokussierte Einzelstudien (Baca et al. 2008, 2013), oder ihre Ergebnisse mit sehr begrenzter Aussagekraft (Tassey 2002)



Wasserfallmodell

https://commons.wikimedia.org/wiki/File:Waterfall_model.svg

Worum geht es eigentlich bei „Design“?

Design is to redesign

Jan Michl

Design is what designers do

Clive Dilnot

Design is to design a design to produce a design. John Heskett

Everyone designs who devises courses of action aimed at changing existing situations into preferred ones.

Herbert Simon

Worum geht es eigentlich bei „Design“?

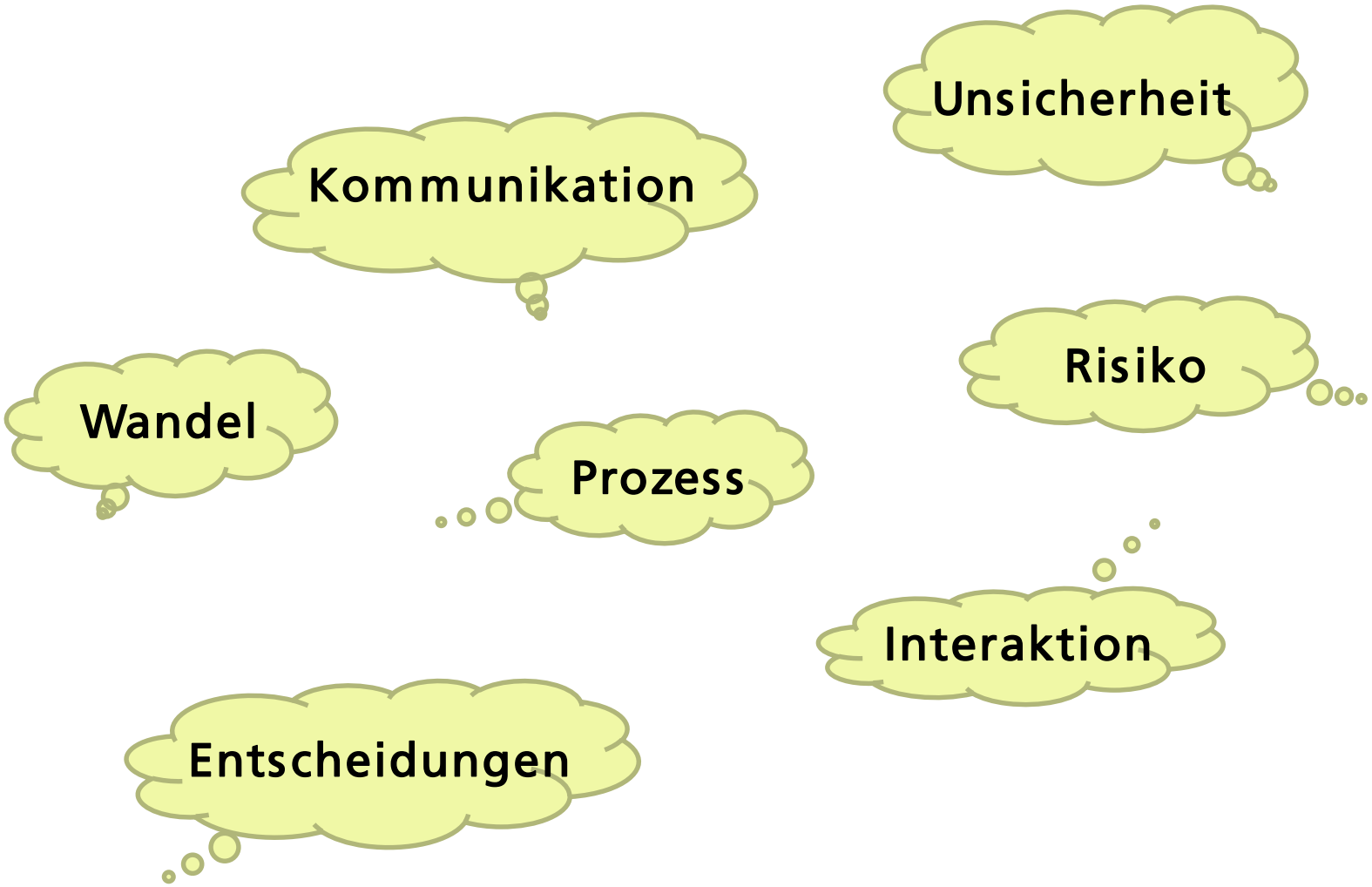
Design is about making decisions, often in the face of uncertainty, It's like running a race where the course keeps splitting. Each fork is a decision.

Joseph Zinter

Good design is also an act of communication between the designer and the user

Donald Normann

Wichtige Aspekte von Design



„Design“ und Softwareentwicklung?

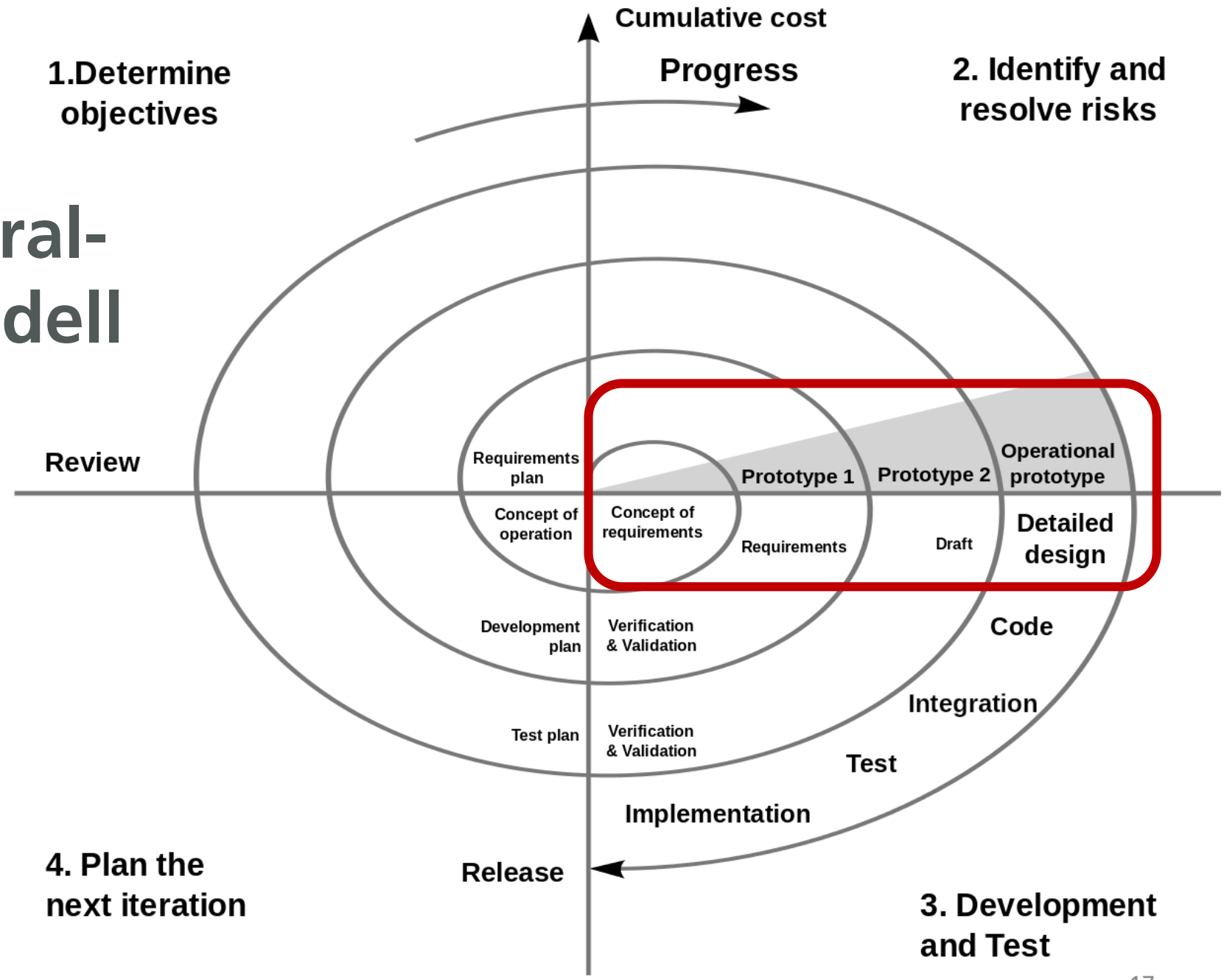
[E]mergent technologies portray a fundamentally different process from the traditional one. The idea that system requirements can be inscribed into the artefact thanks to analytical and problemsolving logic and that development can be broke down into ‘self-containing’, linear, and goal-oriented phases is replaced by an emergent process. [...] Designing is no longer an easily identifiable activity confined within clear boundaries and stated goals.

Giacomo Poderi

„Design“ und Softwareentwicklung?

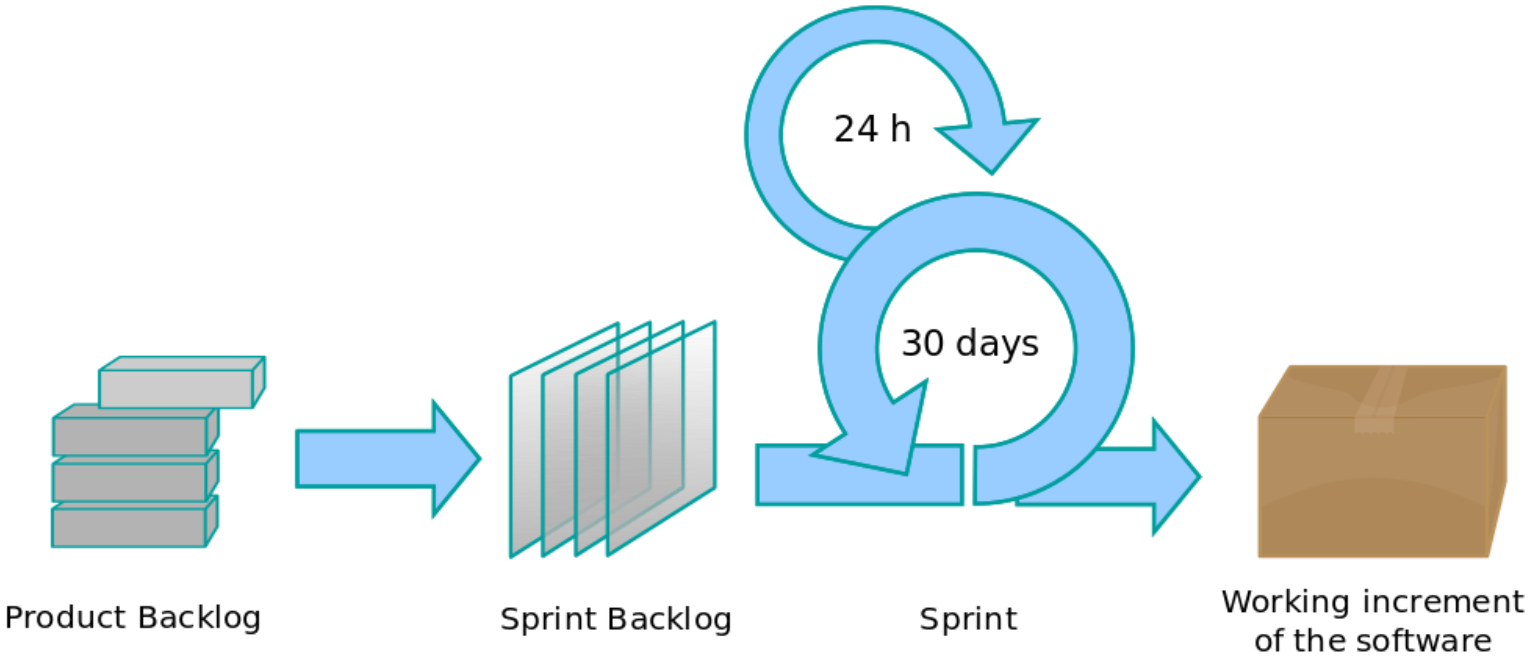
Design as a continuing process that goes on after the formal end of the software development project is, of course, ‘old news’. [...] The ‘new news’ is, that this is where much of the action is today, and it is a much more complex and diverse scene than it was ten years ago. Yvonne Dittrich

Spiral-modell



https://en.wikipedia.org/wiki/File:Spiral_model_(Boehm,_1988).svg

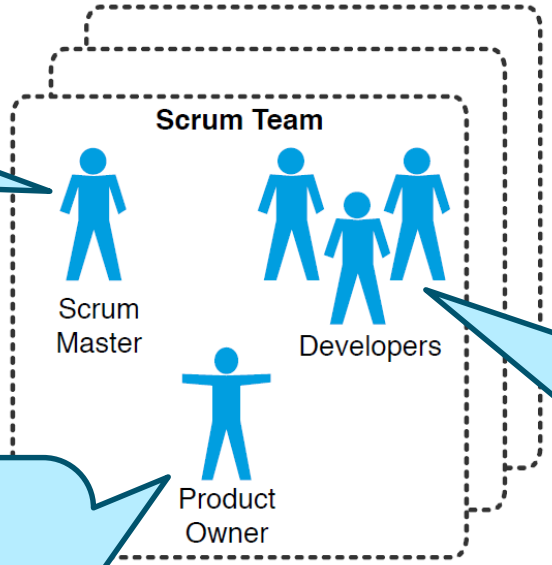
Agile Entwicklung: Scrum



Source: https://en.wikipedia.org/wiki/File:Scrum_process.svg by Lakeworks, CC-BY-SA

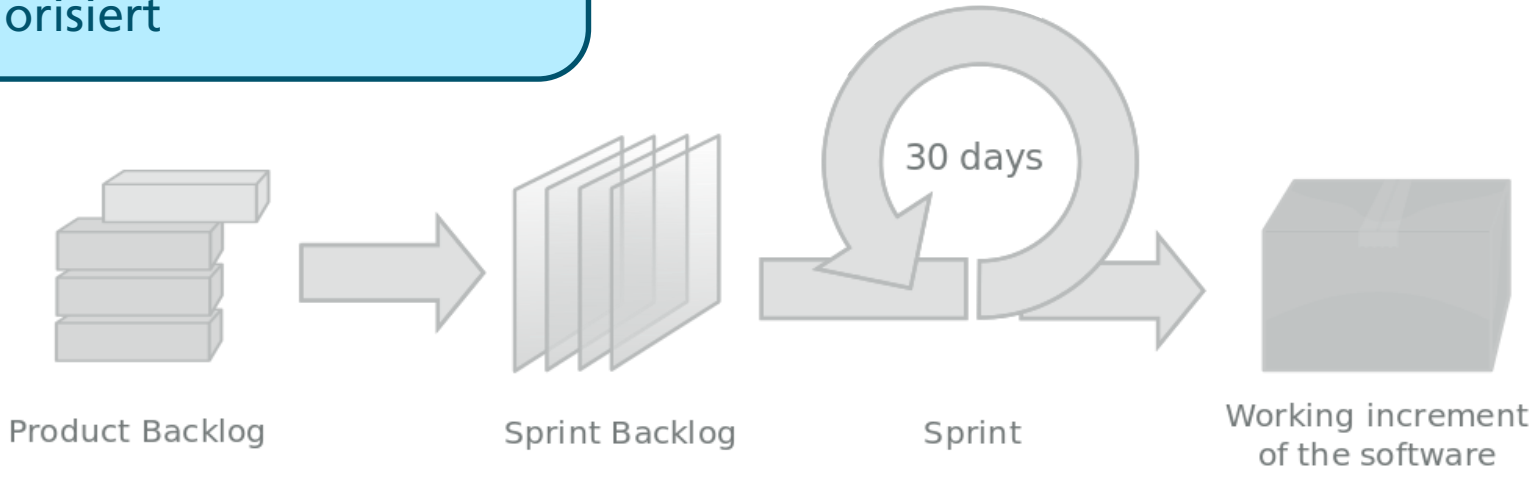
Scrum – Rollen

Scrum Master:
- Organisiert
- Moderiert
- Unterstützt



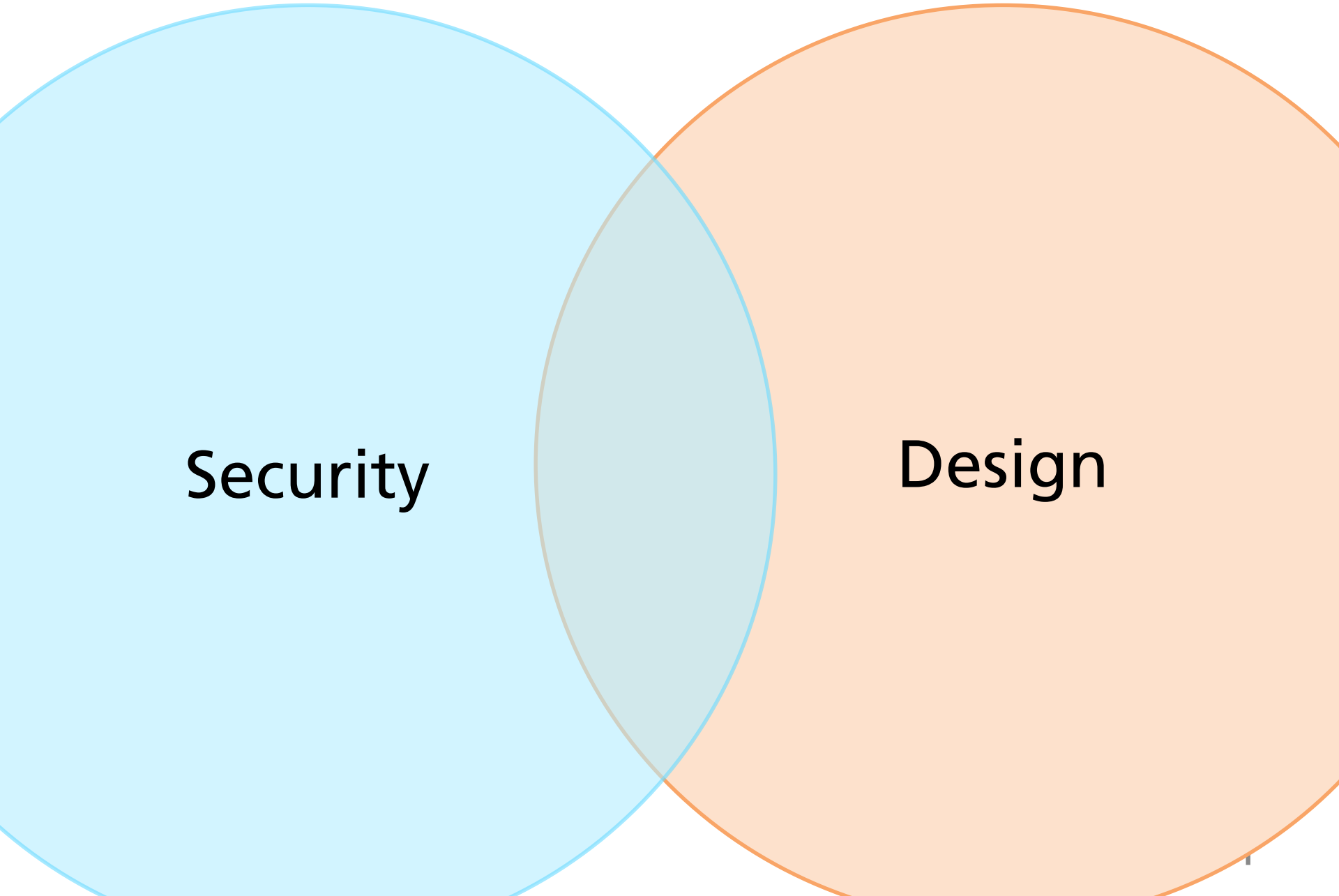
Development team:
- Design
- Implementieren
- Selbstorganisierend
- Crossfunktional

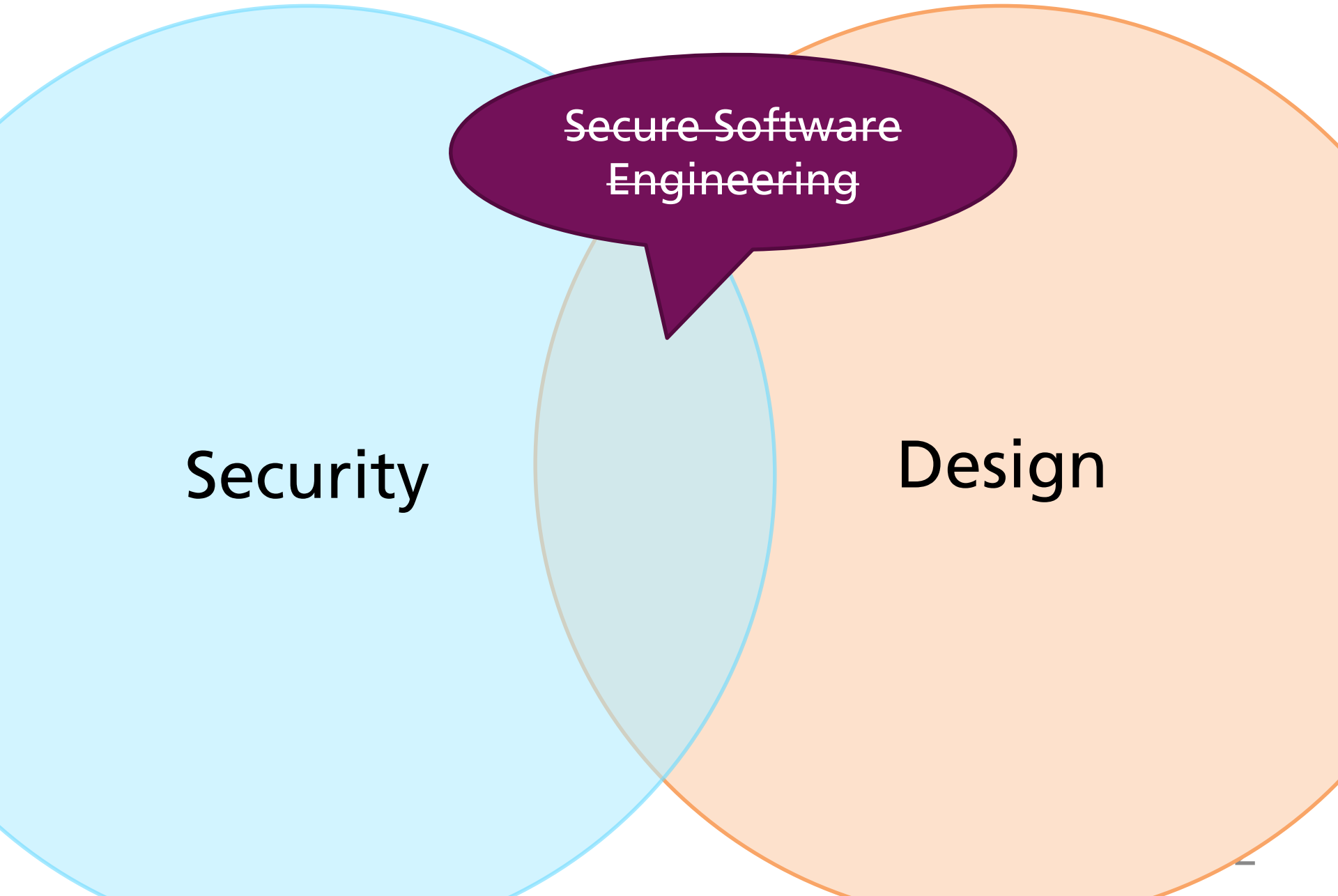
Product Owner:
- Formuliert Anforderungen
- Vertritt Stakeholder
- Priorisiert

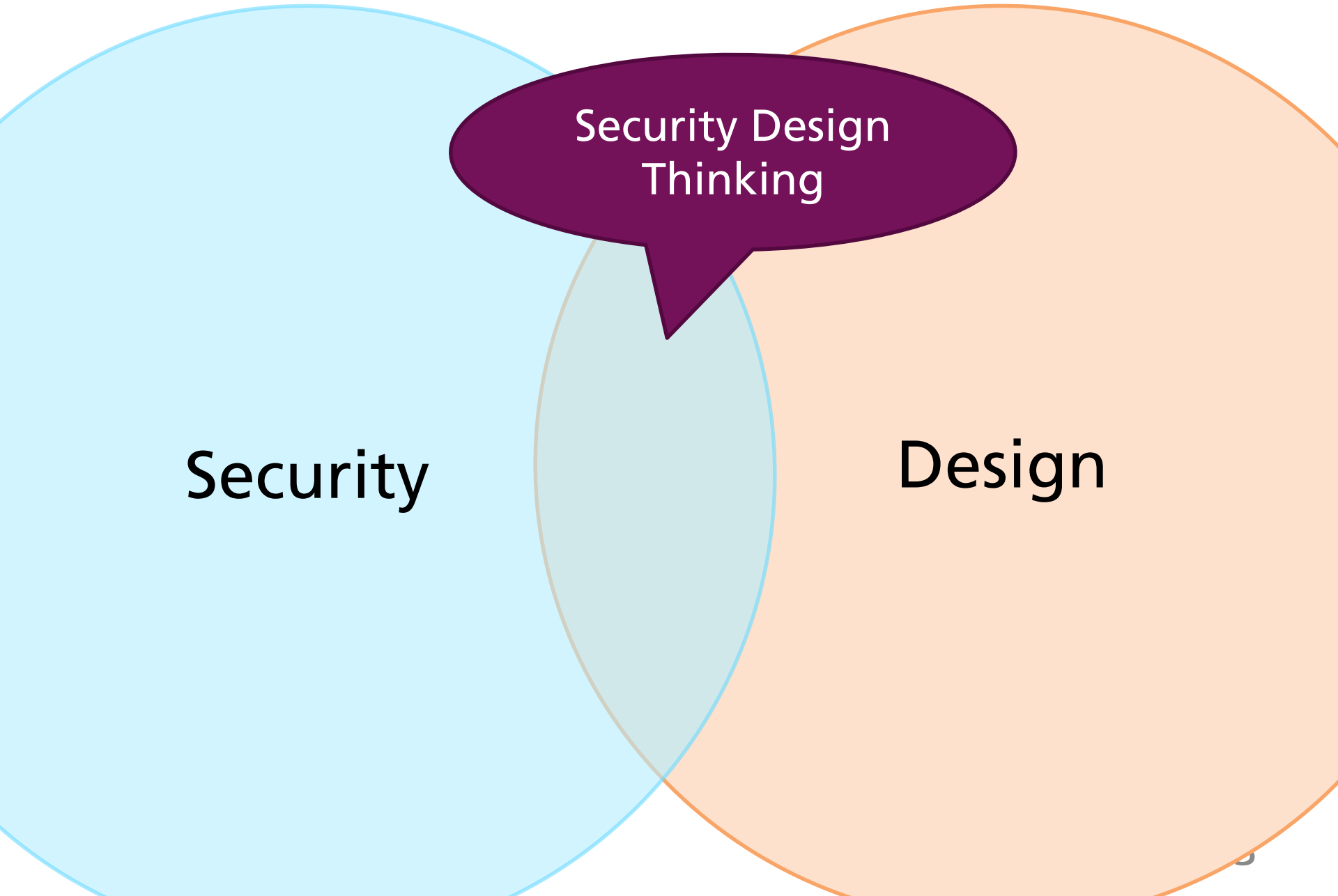


Source: https://en.wikipedia.org/wiki/File:Scrum_process.svg by Lakeworks, CC-BY-SA

Beschäftigt sich „Security by Design“ eigentlich mit diesen Perspektiven?


Security
Design

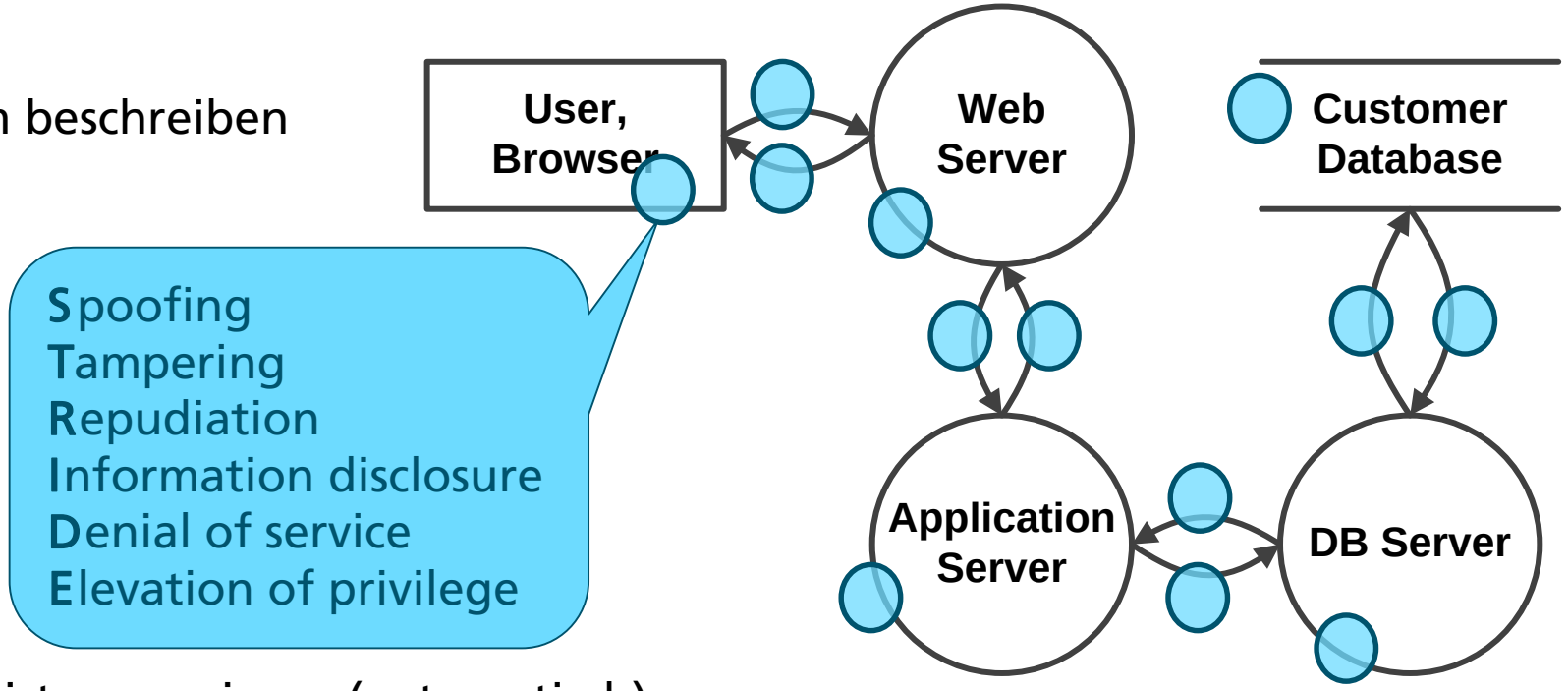

~~Secure Software Engineering~~
Security
Design


Security Design Thinking
Security
Design

“Organizations which design systems ... are constrained to produce designs which are copies of the communication structures of these organizations”

Bedrohungsmodellierung

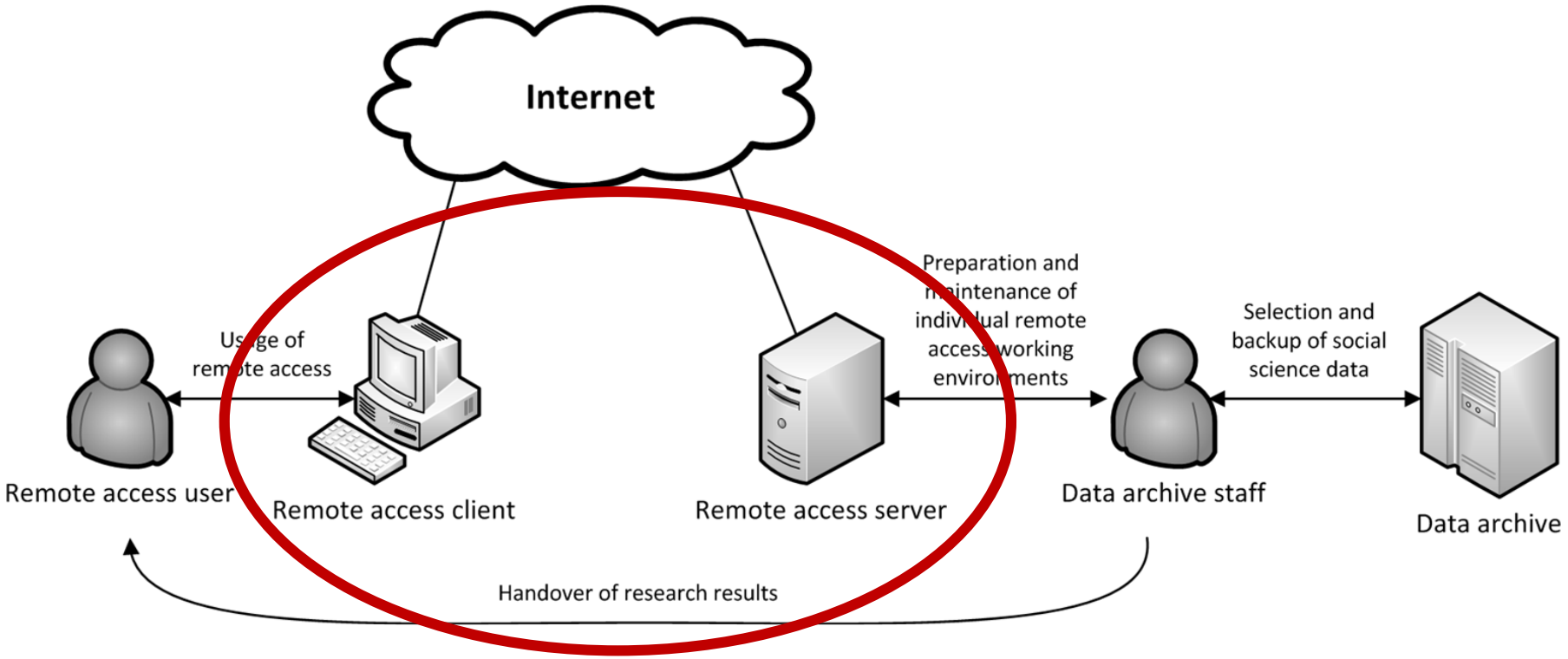
1. System beschreiben



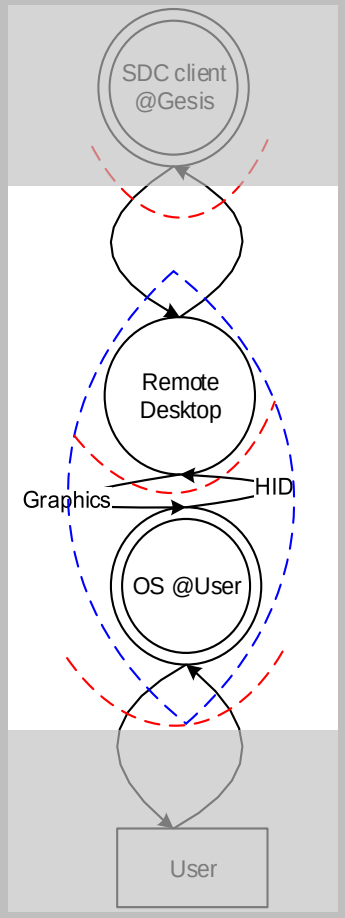
2. Checkliste generieren (automatisch)

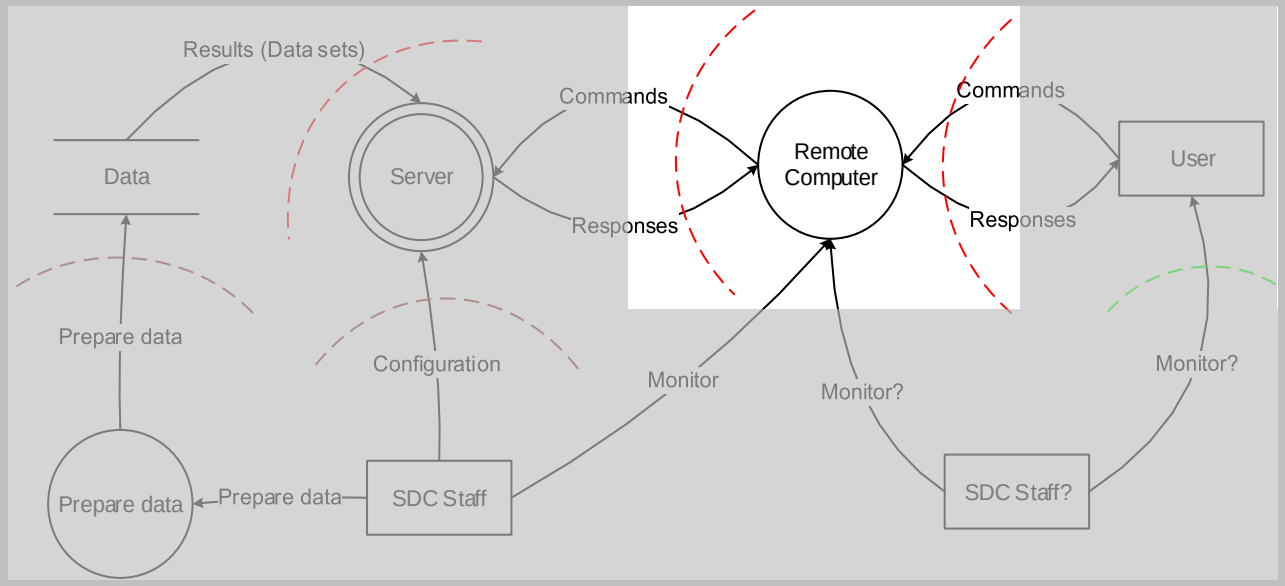
3. Probleme bewerten und lösen

Beispiel:
Fernzugriff auf ein Datenarchiv

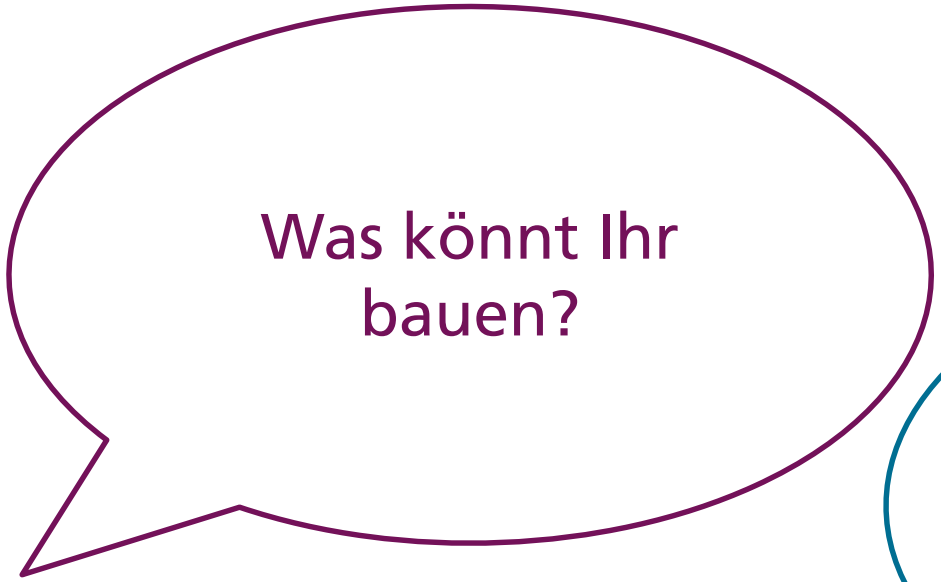


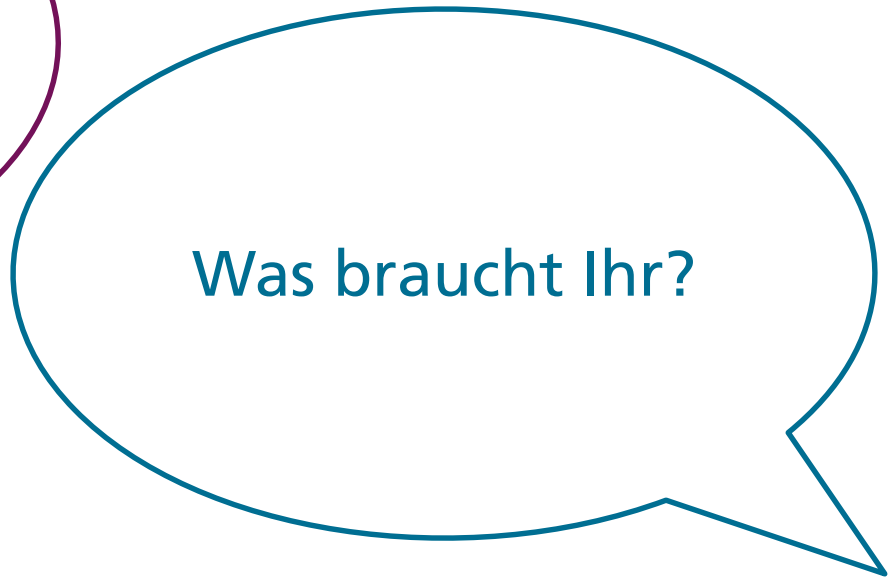
Individuelle Perspektiven





Entwurfsentscheidungen – Henne und Ei





CAST-Workshop:
„Sichere Software entwickeln“
10. November 2016

http://www.cast-forum.de/workshops/infos/227

http://testlab.sit.fraunhofer.de

Andreas.Poller@sit.fraunhofer.de

Sven.Tuerpe@sit.fraunhofer.de

Veröffentlichungen

- Andreas Poller, Laura Kocksch, Sven Türpe, Felix Epp, Katharina Kinder-Kurlanda: Can Security Become a Routine? A Study of Organizational Change in an Agile Software Development Group. CSCW 2017 (*to appear)*
- Jim Whitmore, Sven Türpe, Stefan Triller, Andreas Poller, Christina Carlson: Threat analysis in the software development lifecycle. IBM Journal of Research and Development 58(1) (2014)
- Andreas Poller, Sven Türpe, Katharina Kinder-Kurlanda: An Asset to Security Modeling?: Analyzing Stakeholder Collaborations Instead of Threats to Assets. NSPW 2014: 69-82
- Sven Türpe. "Idea: Usable Platforms for Secure Programming–Mining Unix for Insight and Guidelines." International Symposium on Engineering Secure Software and Systems. Springer International Publishing, 2016.

Referenzen

- Michael Davis (2011). Will software engineering ever be engineering?. Commun. ACM 54, 11 (November 2011), 32-34.
- Aberdeen Group (2010). “Security and the Software Development Lifecycle: Secure at the Source.” Avail-able: http://www.microsoft.com/en-us/download/confirmation.aspx?id=6968 [Accessed 2014-11-27].
- Forrester (2011). “Software Integrity Risk Report”. Available: http://www.coverity.com/library/pdf/Soft-ware_Integrity_Risk_Report.pdf [Accessed 2014-11-19]
- Viega, J. and McGraw, G. (2011). Building Secure Software: How to Avoid Security Problems the Right Way (Paperback). Addison-Wesley Professional.

Referenzen

- Davis, N. (2006). “Secure software development life cycle processes”. Technical Report, CMU/SEI-2005-TN-024, Software Engineering Institute.
- Allen, J.; Alberts, C. and Stoddard, R. (2012). Deriving Software Security Measures from Information Se-curity Standards of Practice. Carnegie Mellon University.
- CIO - Custom Solution Group (2007) “Executive Downloads: A CISO’s Guide to Application Security prepared for Fortify”.
- Kasal, K.; Heurix, J. and Neubauer, T. (2011). “Model-driven development meets security: An evaluation of current approaches.” In: 44th Hawaii International Conference on System Sciences (HICSS). p. 1-9.

Referenzen

- Microsoft (2010). “Security Development Lifecycle: Simplified Implementation of the Microsoft SDL”. Microsoft Corporation.
- Eric Bodden, Markus Schneider, Michael Kreutzer, Mira Mezini, Christian Hammer, Andreas Zeller, Dirk Achenbach, Matthias Huber, Daniel Kraschewski (2014). Development of Secure Software with Security by Design. White Paper, Fraunhofer-Verlag.
- Boehm, B. (1981). Software Engineering Economics. 1st Edition. Prentice Hall PTR, Upper Saddle River, NJ, USA.
- Cavoukian, A. and Chanliau, M. (2013). “Privacy and Security by Design: A Convergence of Paradigms”. Ontario, Canada: Office of the Privacy Commissioner. Ontario.
- Conway, Melvin E. "How do committees invent." Datamation 14.4 (1968): 28-31.

Referenzen

- IBM Global Technology Services (2013). “The economics of IT risk and reputation: What business continuity and IT security really mean to your organisation”. IBM Research Report.
- Tassey, G. (2002). “The economic impacts of inadequate infrastructure for software testing”. National Institute of Standards and Technology, RTI Project 7007 (11).
- Stecklein, J. M.; Dabney, J.; Dick, B.; Haskins, B.; Lovell, R. and Moroney, G. (2004). “Error cost escalation through the project life cycle”. NASA Technical Reports Server (NTRS).
- Baca, D.; Carlsson, B. and Lundberg, L. (2008). “Evaluating the cost reduction of static code analysis for software security.” In: Proceedings of the third ACM SIGPLAN workshop on Programming languages and analysis for security. ACM, p. 79-88.

Referenzen

- Poderi, Giacomo. "Innovation Happens Elsewhere, but Where Does Design Happen? Considerations on Design and Participatory Processes in Emerging Information Technologies." TECNOSCIENZA: Italian Journal of Science & Technology Studies 3.1 (2012): 63-72.