Entwicklung Vertrauenswürdiger Web-APIs

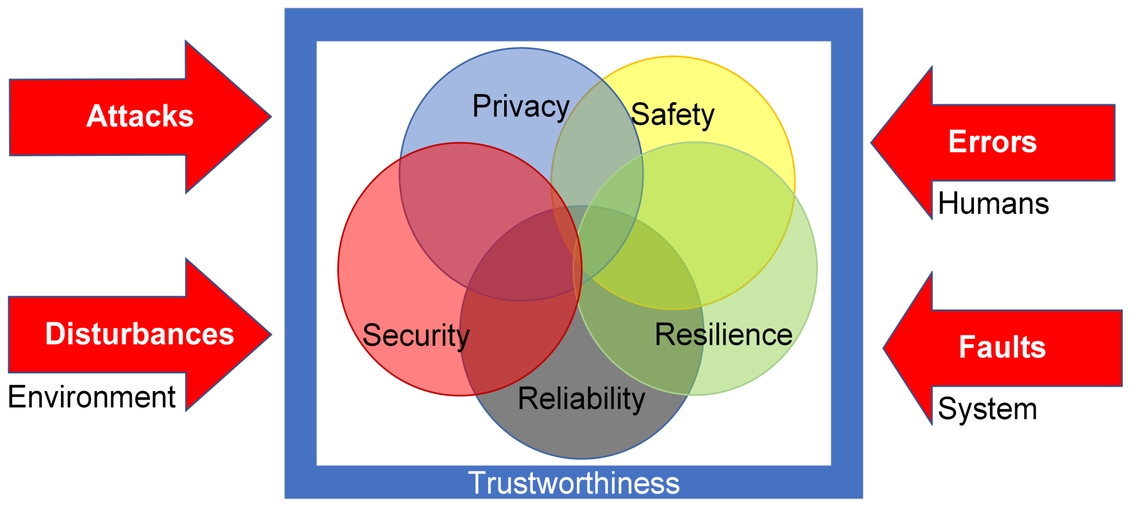
Hochschule für
Wirtschaft und Recht Berlin
Berlin School of Economics and Law

Sandro Hartenstein, M. Sc.
sandro.hartenstein@hwr-berlin.de
www.hwr-berlin.de 3.11.2020

Einleitung

Die Digitalisierung unserer Gesellschaft bewirkt den hohen Einsatz von Software. Da zunehmend auch sensible Bereiche, z.B. Gesundheitsdienstleistungen, adressiert werden, ist die Entwicklung von vertrauenswürdigen Anwendungen notwendig. Neue digitalisierte Dienste bieten nicht nur die Vorteile der orts- und zeitunabhängigen Verfügbarkeit, sowie eine schnelle Prozessbearbeitung, sondern es existieren auch neue Gefahren, wie beispielsweise der unberechtigte Datenabfluss und die Manipulation durch Dritte. Diese Risiken adäquat zu adressieren, erfordert eine vertrauenswürdige Softwareentwicklung. Diese Ausarbeitung zeigt einen Ansatz, das iterative Scrum-Modell für API-basierten Softwareprojekte im Kontext der vertrauenswürdigen Entwicklung zu optimieren. Sie stellt zudem Ausgangspunkt einer empirischen Untersuchung zur Wirksamkeit und Effizienz von Prozessveränderungen dar.[1]

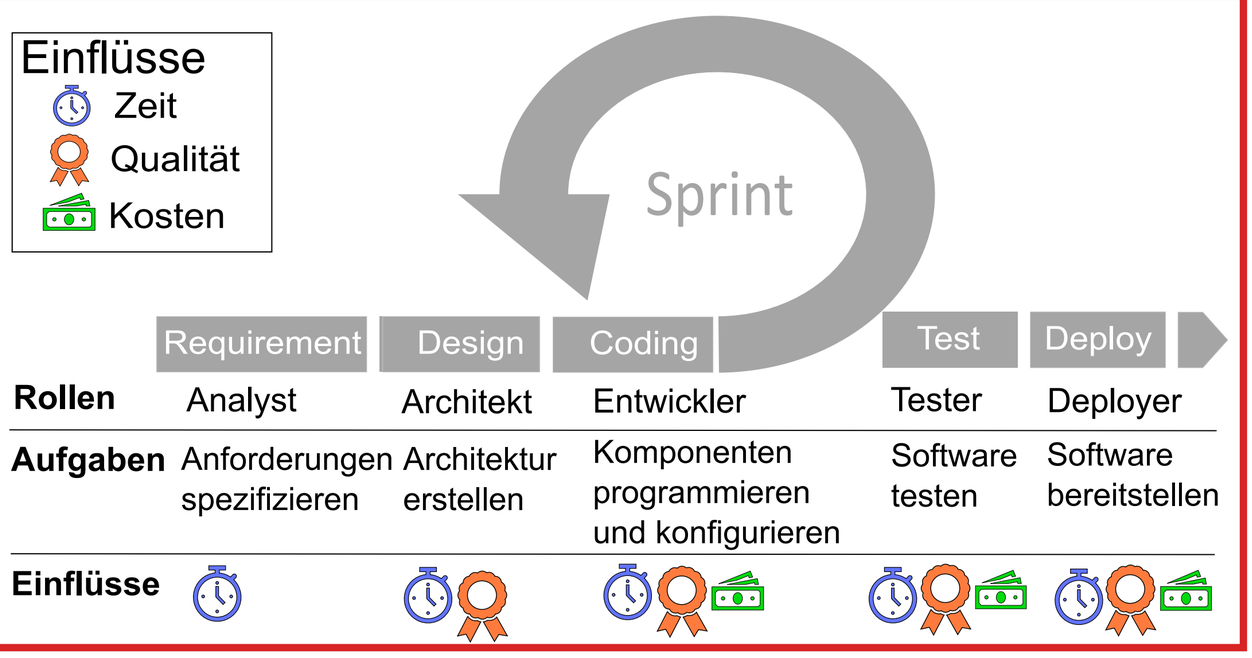
Problemstellung



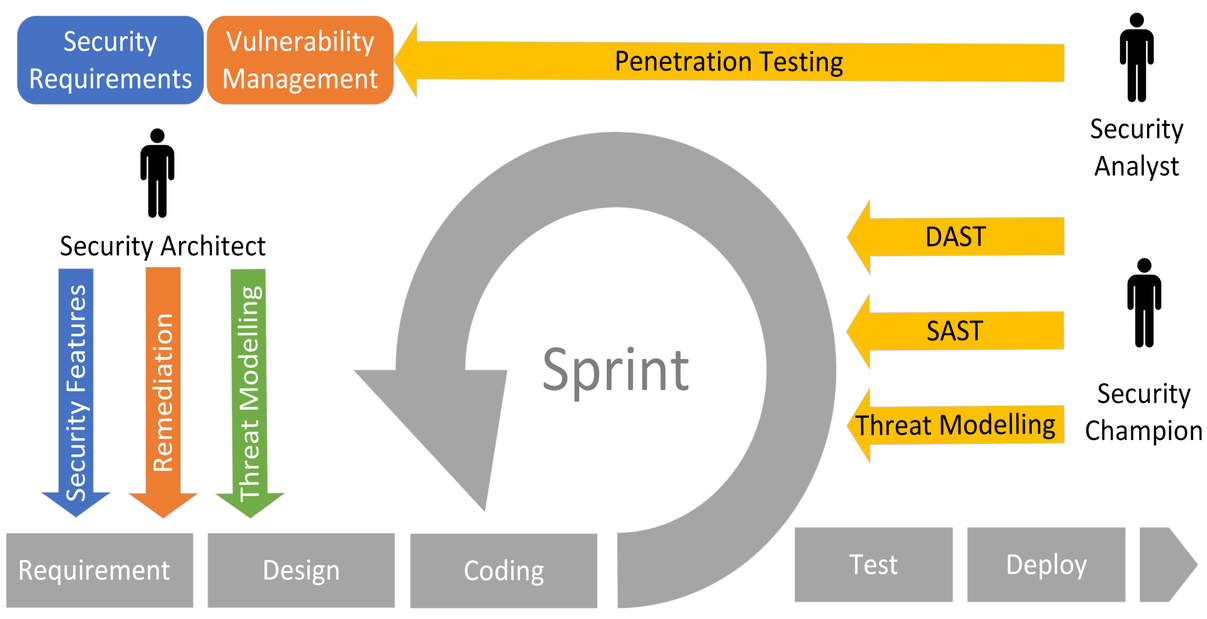
Die Vertrauenswürdigkeit ist eine wichtige Eigenschaft von Web-APIs und ist durch die Attribute Sicherheit, Privatheit, Safety, Widerstandsfähigkeit und Zuverlässigkeit geprägt. Bekannte Einflüsse auf diese Attribute sind Fehler und Angriffe auf das System, die negative Auswirkungen haben können und Risiken darstellen. Diese Risiken müssen durch geiegnete Anforderungen, Architekturen und Softwareentwicklungsverfahren adressiert werden.[2]

Forschungsstand

Die Zeit ist eine wichtige Dimension im Softwareprojekt. Jede Änderung in jeder Phase kann zu Verzögerungen der führen [3] S. 108. Die Qualität, insbesondere die Eigenschaft Vertrauenswürdigkeit, ist besonders von Anpassungen in den Implementierungsphasen (Design, Coding, Test und Deploy) betroffen [4] S. 22. Der Einfluss auf die Kosten ist in der Coding-, Test- und Deploymentphase im Fokus, da die Kosten von Nacharbeiten in den späten Phasen sehr hoch sind. Diese Tätigkeiten sind sehr kostenintensiv [5]



Methode



Der Shift-Left Ansatz sieht vor, Fehler frühestmöglich zu erkennen und zu beheben. Dazu wird die in den frühen Projektphasen priorisiert und in den späteren Phasen automatisiert.. Durch ein Schwachstellenmanagement, soll das Projekt von Fehlern vorheriger Entwicklungen lernen.

Ausblick

Die Wirksamkeit wird mit einer Simulation des optimierten Prozesses ermittelt. Hierbei werden alle Akteure virtuell nachgebildet. Die Auswirkungen der einzelnen Parameter auf die Vertrauenswürdigkeit sind das Hauptziel der Untersuchung.

Literatur

[1] SCHIEFERDECKER, Ina: Responsible Software Engineering. In: GOERICKE, Stephan (Hrsg.): The future of software quality assurance. Cham, Switzer-land : SpringerOpen, 2020, S. 137–146
[2] M. Buchheit, F. Hirsch, S. Rix, M. Hermeling, and B. Martin, Software Trustworthiness Best Practices. [Online].https://www.iiconsortium.org/pdf/Software_Trustworthiness_Best_Practices_Whitepaper_2020_03_23.pdf.
[3] HARISH, Prashanth Southekal: Measurement Framework For Software Projects : A Generic And Practical Goal-Question-Metric(Gqm) Based Approach : Trafford Publishing, 2011
[4] RAMESH, Gopalaswamy: Managing global software projects : How to lead geographically distributed teams, manage processes and use quality mo-dels. New Delhi : Tata McGraw-Hill, 2006
[5] GREGORY, Tassey: The Economic Impacts of Inadequate Infrastructure for software Testing. 2002