

HFN

HOCHSCHULE HEILBRONN

TECHNIK

WIRTSCHAFT

INFORMATIK

Opti4Apps – Optimierung von mobilen Applikationen durch einen Feedback-basierten Qualitätssicherungsansatz

Michael Bauer, DIVIGATION 2017

Agenda

- ▶ Einführung
 - ▶ Motivation
 - ▶ Ziele
 - ▶ Konsortium

- ▶ Ansatz
 - ▶ Überblick
 - ▶ Grundlagen
 - ▶ Prozess

- ▶ Datenanalyse
 - ▶ Nutzungsdaten
 - ▶ Nutzerfeedback
 - ▶ Werkzeugkettendaten

Einführung

Motivation, Ziele, Konsortium;

Motivation

- ▶ Kurze, iterative Produktentwicklungszyklen prägen den Markt mobiler Applikationen
- ▶ Unternehmen gehen zunehmend dazu über Minimal Viable Products (MVPs) einzusetzen um innovative Geschäftsideen zur schnellen Marktreife zu führen
- ▶ Die Einbeziehung echter Nutzer in den Entwicklungsprozess ist aufwändig

Ziele

- ▶ Kurze Entwicklungszyklen durch die Entwicklung von mobilen Applikationen als MVP
- ▶ Automatisierte Auswertung und Analyse von Benutzer-Feedback und Nutzungsdaten
- ▶ Fokussierte Qualitätssicherung anhand auftretender Fehlermuster

Konsortium

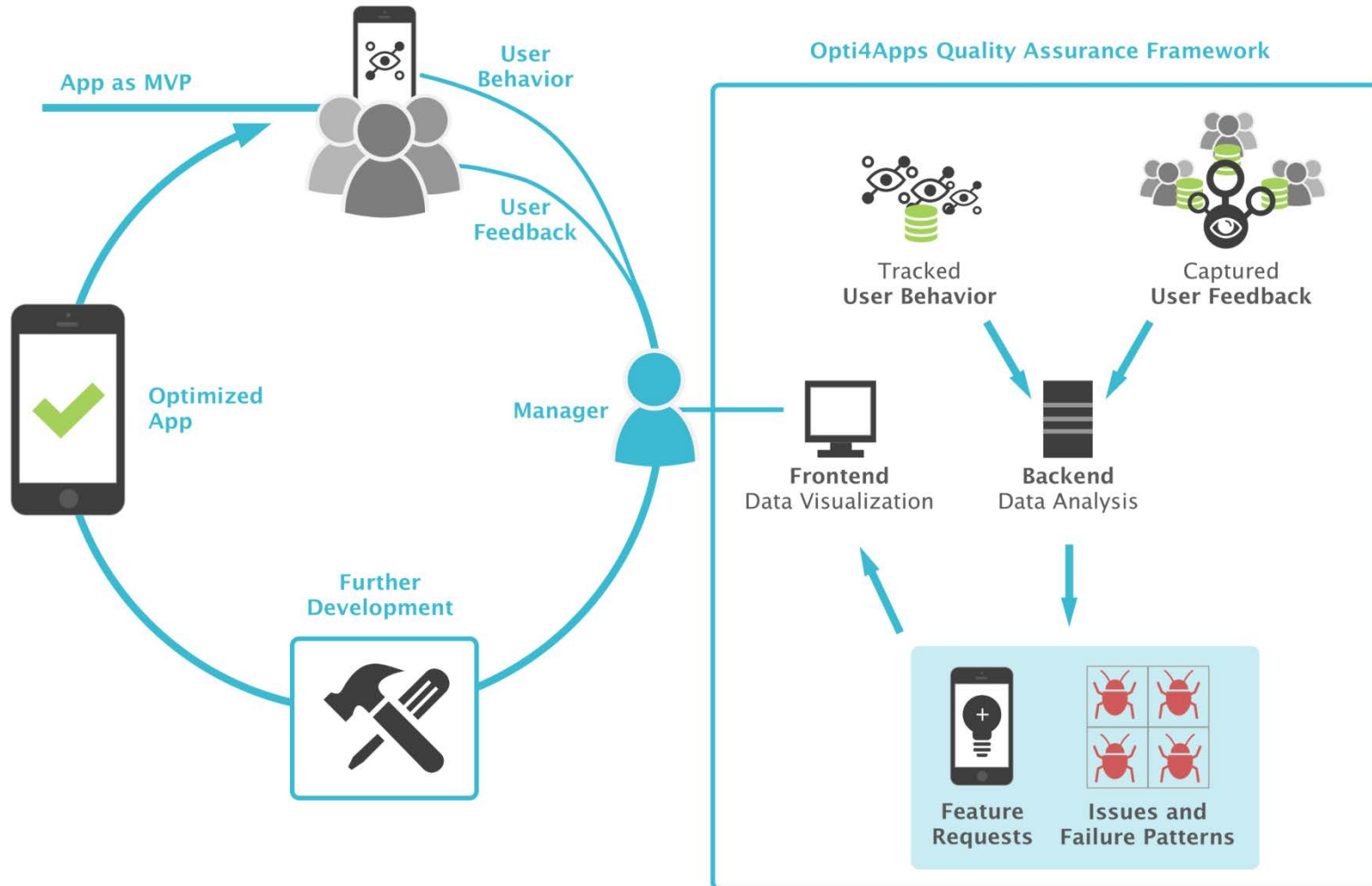
- ▶ Centigrade
 - ▶ Opti4Apps Nutzeranalysemethodik
 - ▶ Opti4Apps Entwicklungsprozess
- ▶ Heidelberg Mobil
 - ▶ Opti4Apps Framework
 - ▶ Opti4Apps Realisierung
- ▶ Fraunhofer IESE
 - ▶ Opti4Apps Konzeptentwicklung
 - ▶ Fokus: Qualitätssicherung / Framework
- ▶ Hochschule Heilbronn
 - ▶ Opti4Apps Konzeptentwicklung
 - ▶ Fokus: Data Mining / Evaluation



Ansatz

Überblick, Grundlagen, Prozess;

Überblick



Grundlagen – Minimum viable product (MVP)

“The minimum viable product (MVP) is that version of a new product which allows a team to collect the **maximum amount of validated learning** about customers with the **least effort**”

(Eric Rieß)

Grundlagen – Nutzungsverhalten und Nutzerfeedback

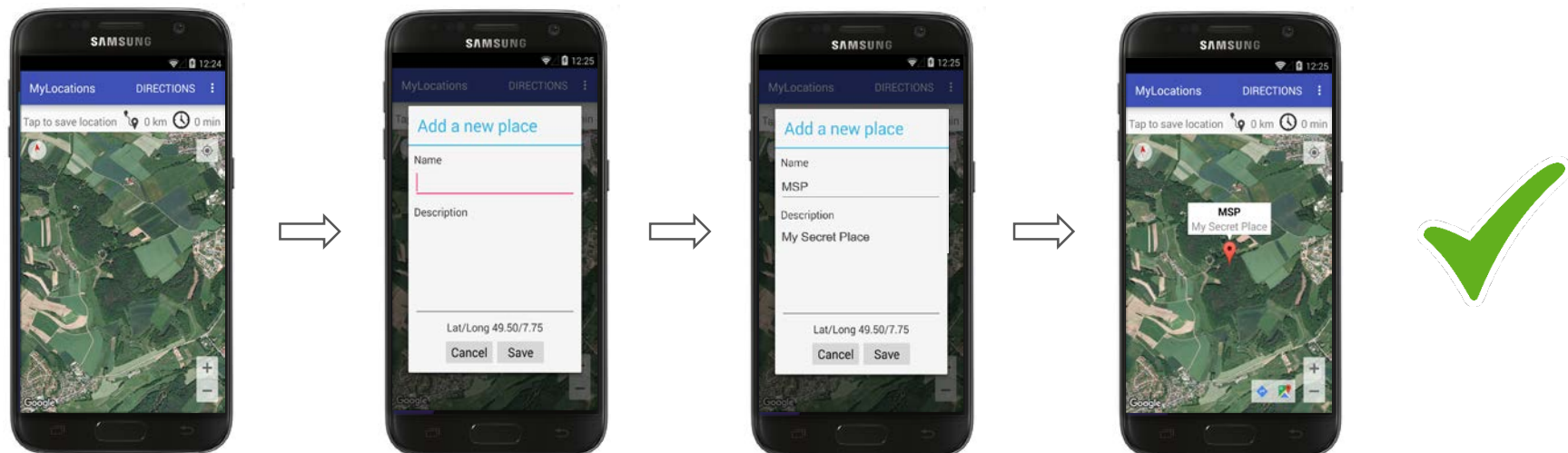
- ▶ Nutzungsverhalten – Das Verhalten des Nutzers während der Interaktion mit dem MVP
 - ▶ Aktionen wie Scrollen oder Klicken
 - ▶ Verweilzeiten auf Screens

- ▶ Nutzerfeedback – Feedback das der Nutzer aufgrund seiner Erfahrungen bei der Nutzung des MVP gibt
 - ▶ Kommentare
 - ▶ Bewertungen



Grundlagen – Fehlermuster

- ▶ Fehlermuster definieren die Verbindung zwischen typischen Fehlerursachen und deren üblichen Auswirkungen

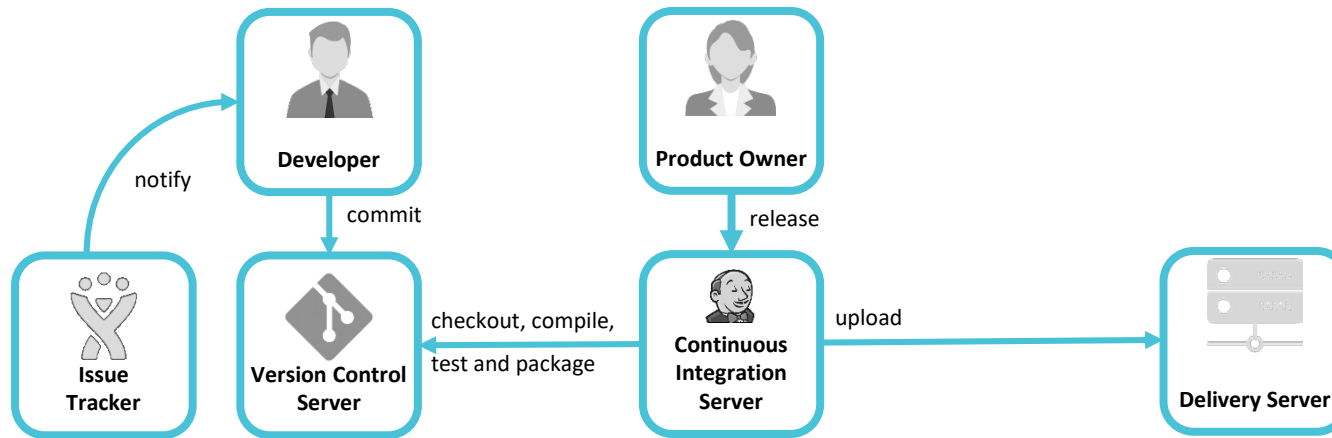


Grundlagen – Fehlermuster

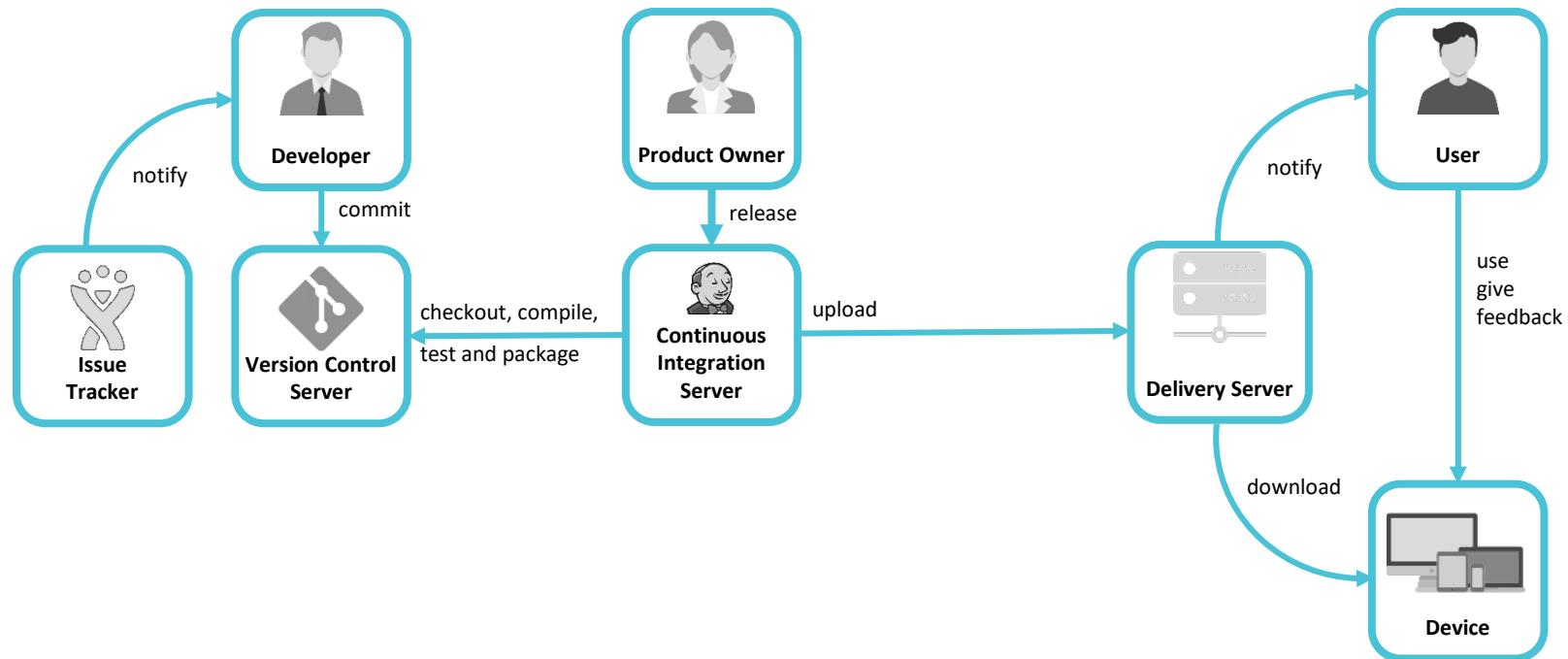
- ▶ Fehlermuster definieren die Verbindung zwischen typischen Fehlerursachen und deren üblichen Auswirkungen



Prozess – Continuous Delivery Toolchain

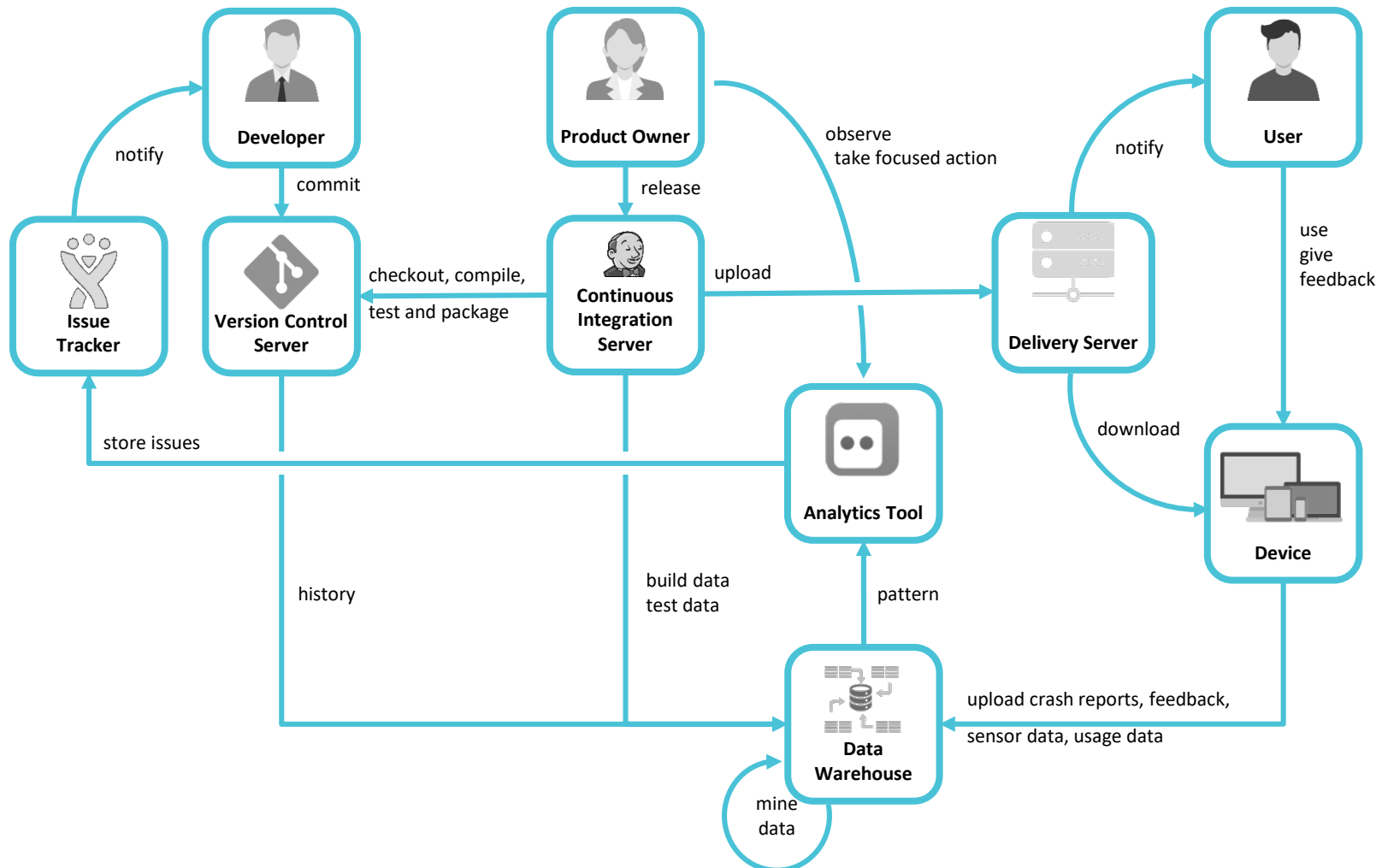


Prozess – Integration von Nutzerfeedback



(Krusche & Alperowitz)

Prozess – Integration der Datenanalyse



Datenanalyse

Nutzungsdaten, Nutzerfeedback, Werkzeugkettendaten

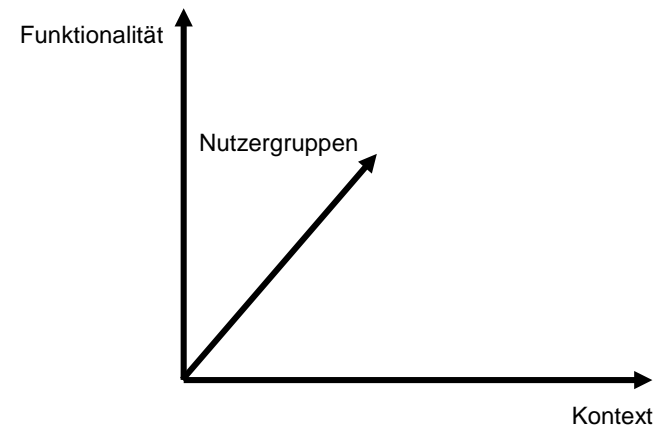
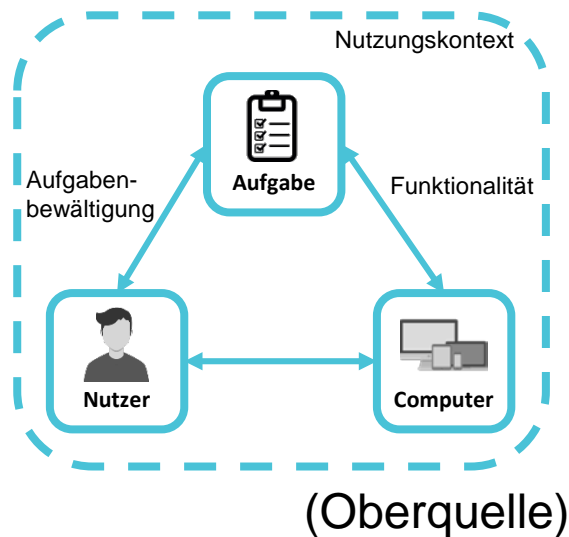
Nutzungsdaten

- ▶ Nutzungsdaten enthalten Verhaltensmuster der Nutzer in der Interaktion mit der Applikation
- ▶ Klickpfade
 - ▶ Welches UI-Element wurde zu welchem Zeitpunkt mit welcher Geste benutzt?
 - ▶ Gibt es nutzergruppenspezifische Nutzungsmuster?
- ▶ Applikation
 - ▶ Welche Funktionalitäten werden wie oft aufgerufen oder verursachen Fehler?
- ▶ Endgerät
 - ▶ In welchem Zustand befindet sich das Endgerät während der Nutzung?
- ▶ Sensordaten
 - ▶ In welchem Kontext wird die Applikation genutzt?



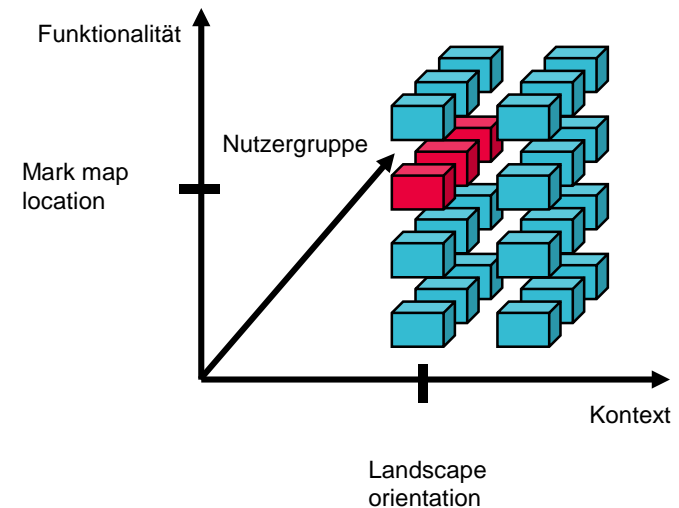
Nutzungsdaten

- ▶ Nutzungsdaten enthalten Verhaltensmuster der Nutzer in der Interaktion mit der Applikation und Daten zum Kontext der Nutzung



Nutzungsdaten

- Nutzungsdaten enthalten Verhaltensmuster der Nutzer in der Interaktion mit der Applikation und Daten zum Kontext der Nutzung



Nutzerfeedback

- ▶ Nutzerfeedback enthält Meinungen, Erfahrungen und Bewertungen der Nutzer
- ▶ App Store
 - ▶ Welches Sterne Rating hat die Applikation?
 - ▶ Welche im Bezug auf die Softwarequalität relevanten Aussagen wurden in den Kommentaren getroffen?
 - ▶ Welche Aussagen über die Qualität führen zu schlechten Bewertungen?
- ▶ Web & Social Media
 - ▶ Welche im Bezug auf die Softwarequalität relevanten Aussagen wurden in den Kommentaren getroffen?



Nutzerfeedback

- ▶ Studie von Khalid et al.
 - ▶ Untersuchung der 20 meistgenutzten Applikation mit mindestens 750 Reviews
 - ▶ 12 Beschwerdetypen wurden identifiziert
 - ▶ Die Häufigkeit des Beschwerdetyps und die Wahrscheinlichkeit auf eine schlechte Bewertung wurden eingeordnet

| Complaint type | Description | Example Review | Frequency | Impact on star rating |
|------------------|------------------------------|--------------------------------------------------------------|-----------|-----------------------|
| Functional Error | The problem was app specific | “Not getting notifications unless you actually open the app” | Rank 1 | Rank 7 |
| App Crashing | The app often crashed | “Crashes immediately after starting” | Rank 3 | Rank 4 |
| ... | | | | |

Nutzerfeedback

- ▶ Studie von Khalid et al.
 - ▶ Applikationsspezifische Umsetzung
 - ▶ Reduzierung der benötigten Reviews
 - ▶ Nutzung von Web & Social Media Texten
 - ▶ Neben Sternebewertung wird ein Sentiment Score verwendet

| Complaint type | Description | Example Review | Frequency | Impact on star rating |
|------------------|------------------------------|--------------------------------------------------------------|-----------|-----------------------|
| Functional Error | The problem was app specific | “Not getting notifications unless you actually open the app” | Rank 1 | Rank 7 |
| App Crashing | The app often crashed | “Crashes immediately after starting” | Rank 3 | Rank 4 |
| ... | | | | |

Werkzeugkettendaten

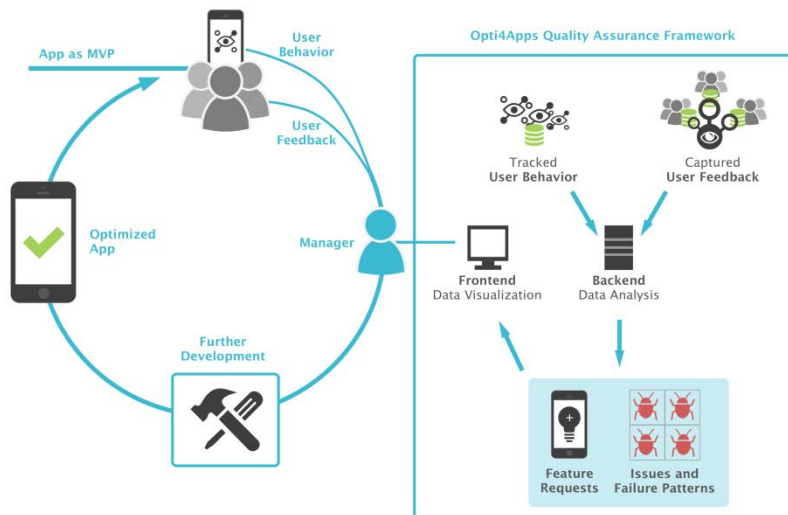
- ▶ Werkzeugkettendaten enthalten
entwicklungsspezifische Informationen
- ▶ Git
 - ▶ Welche Funktionalität (Code) wird welchem Feature-Branch zugeordnet?
 - ▶ Welche Funktionalität unterliegt häufigen Änderungen (durch verschiedene Entwickler)?
 - ▶ Treten Änderungen im Code häufig gemeinsam auf? (Copy Paste)

Werkzeugkettendaten

- ▶ Werkzeugkettendaten enthalten
entwicklungsspezifische Informationen
 - ▶ Automatisierte Identifikation von strukturellen Schwachstellen im Code
 - ▶ Zuordnung der Klickpfade zu Feature-Banches
 - ▶ Lokalisierung von Funktionalität die Probleme bei der Nutzung verursachen

Zusammenfassung

Zusammenfassung



1. Mobile Applikation als MVP entwickeln
2. Nutzerverhalten und Nutzerfeedback sammeln
3. Häufige Fehlermuster erkennen
4. Erkenntnisse (semi-) automatisiert aufbereiten bzw. visualisieren
5. Datenbasiert entscheiden

Vielen Dank für Ihre Aufmerksamkeit!

www.opti4apps.de

Michael Bauer | +49 7131 504 6744 | michael.bauer@hs-heilbronn.de

Funded by:



PTKA
Projekträger Karlsruhe
im Karlsruher Institut für Technologie

In cooperation with:

