

Software Engineering

Übungsblatt 2

11. Mai 2018

Aufgabe 1: Kano-Modell

a)

1) Basismerkmale

Beschreibung: Mindestanforderungen, die als vorausgesetzt gelten und den User nicht unbedingt beeindrucken, aber bei Fehlen schnell zu Unzufriedenheit führen.

Beispieleigenschaften: Touchbedienung

2) Leistungsmerkmale

Beschreibung: Eigenschaften, die dem User bewusst sind und ihn zufrieden stellen, wenn diese Eigenschaften (in großem Ausmaß) vorhanden sind.

Beispieleigenschaften: hohe Akkulaufzeit

3) Begeisterungsmerkmale

Beschreibung: Unerwartete Zusätze, die den User begeistern.

Beispielsmerkmale: Fingerprint-Sensor

4) Unerhebliche Merkmale

Beschreibung: Für den User triviale Eigenschaften, die (im generellen) keine Zufriedenheit/Emporung auslösen.

Beispielsmerkmale: Eingebauter Barcode-Scanner

5) Rückweisungs-Merkmale

Beschreibung: Eigenschaften, die bei Vorhandensein zu Unzufriedenheit und bei Fehlen aber meistens nie zu Zufriedenheit führen.

Beispielsmerkmale: schnelles Erhitzen

b)

dysfunktionale Frage: "Was würden Sie davon halten, wenn unser Smartphone keinen Fingerprint-Sensor eingebaut hätte?"

c)

funktionale Frage: "Was würden Sie davon halten, wenn sich unser Smartphone schnell erhitzen würde?"

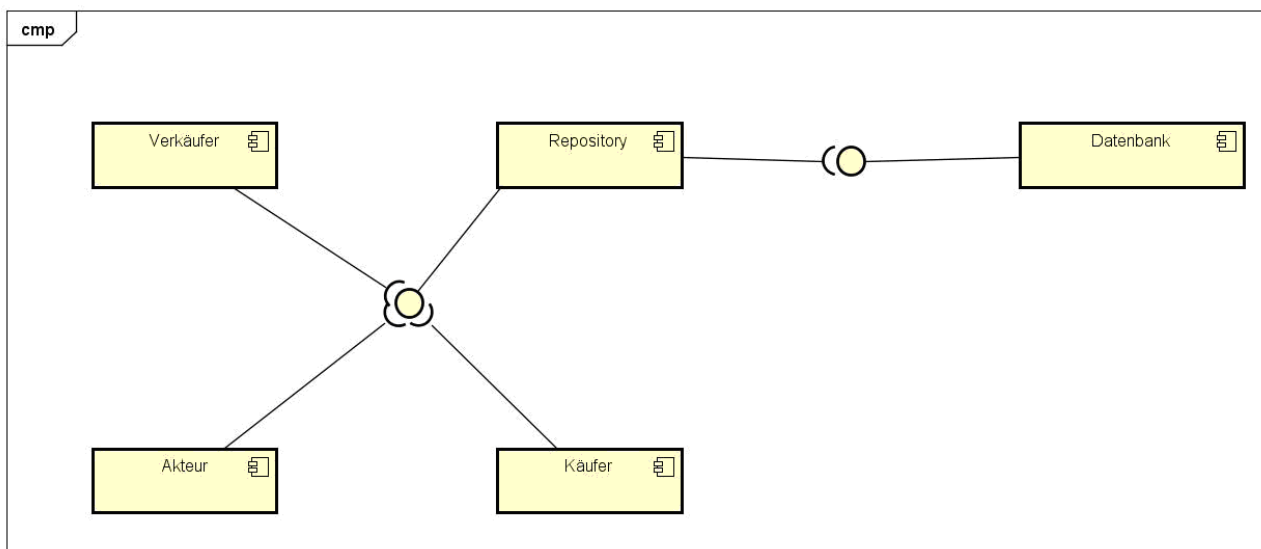
d)

Leistungs- und Begeisterungsmerkmale können im Laufe der Zeit zu Basismerkmalen werden, da sich die User daran gewöhnt haben und sich somit diese Eigenschaften in der Gesellschaft etabliert haben.

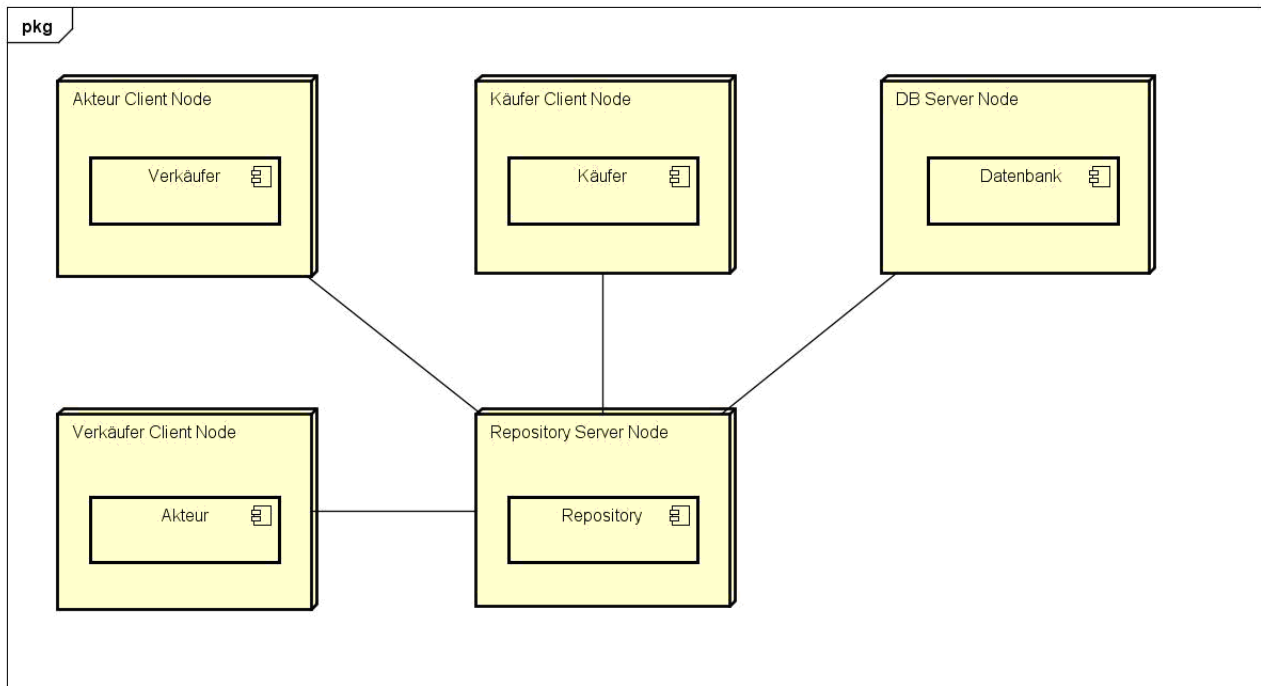
Die Touchbedienung zum Beispiel war zu Release ein Begeisterungsmerkmal, da Handys auch ohne auskommen können, aber die Bedienung dadurch stark vereinfacht wird. Heutzutage ist dieses Feature kaum noch mehr wegzudenken und ist somit zu einem Basismerkmal geworden.

Aufgabe 2: Softwarearchitektur

a)



b)



c)

Da sowohl Verkäufer, Akteure, als auch Käufer auf den Repository-Server zugreifen, kann es zu erheblichen Verlangsamungen im System kommen.

Aufgabe 3: Softwarearchitektur

a)

Programmierer 3 ist der Softwarearchitekt. Er ist ein Entwurfsspezialist, da er selbst ein Programmierer ist und somit über genug praktische Kenntnisse verfügt, die auch von Programmierer 2 bestätigt wurden. Daher ist er in der Lage Programmierer 1 zu leiten. Insgesamt hat er auch eine Übersicht über das Projekt, da er weiß, was dem Projekt noch fehlt und wie lange die Umsetzung dauert.

Außerdem wird der Architekt in diesem Fall als Voodoo-Priester wahrgenommen, da es den anderen Programmierern an Know-How fehlt und er somit als "Wunderheiler" gesehen wird.

b)

Es handelt sich um eine Applikationsarchitektur. Im Fokus steht eine einzige Applikation, nämlich die Tablet-App, die alle benötigten Daten zur Infrastruktur anzeigen kann. Auch handelt es sich um ein einziges Projekt mit nur einem Programmiererteam. Die Anweisungen des Architekten sind sehr detailliert.

c)

- klare Anforderungen
- wartbare Applikation
- gute Infrastruktur durch Microservice
- modularer Aufbau
- unabhängige Entwicklung durch verschiedene Sprachen
- unabhängige Verteil- und Betreibbarkeit
- dezentralisierte Datenbank
- keine Transaktionen zwischen den Services

d)

- Es wird erst auf Probleme reagiert, nach dem Eintreten (siehe fehlender Drehstromanschluss)
- Programmierer 3 bald in Rente (kein klarer Nachfolger bekannt)
- keine klare Zeitplanung
- kleine Anzahl von Beschäftigten
- Programmierer 1 fehlt womöglich know-how

e)

Programmierer 3 scheint alles im Griff zu haben, aber die oben genannten Mängel könnten womöglich dazu führen, dass sich die Fertigung verzögert.

f)

- klare Aufgabenverteilung
- klare Zeitplanung (z.B. durch ein Gantt-Diagramm)
- mehr Angestellte
- Fortbildungskurse für fehlendes Know-How
- Ersatz für Programmierer 3 suchen