Freie Universität Berlin

Qualitätserlebnis statt Qualitätssicherung. Eine Mehrfachfallstudie agiler Teams

12.06.2014, Abschlussvortrag Masterarbeit
Holger Schmeisky

Freie Universität Berlin

Die Forschungsfrage

- Wie und unter welchen Bedingungen funktioniert agile Qualitätssicherung?
- Wenn sie gut funktioniert, warum funktioniert sie?

Freie Universität Berlin

Die Methodik

- Explorative Fallstudie von 3 agilen Teams bei Immoscout24 und SoundCloud
- Beides Webplattformen, um Vergleichbarkeit zu haben

Freie Universität Berlin

Die Methodik

- Datentriangulation
- Entwicklerbegleitung (ca. 3 Tage)
- Interviews (mit Entwickler, PO, Teamleiter, Scrum master,...)
- Analyse durch narrative Form
- Iteratives Vorgehen
- Verlässlichkeit
- Validierung durch Diskussion der Ergebnisse

Freie Universität Berlin

Ergebnisse

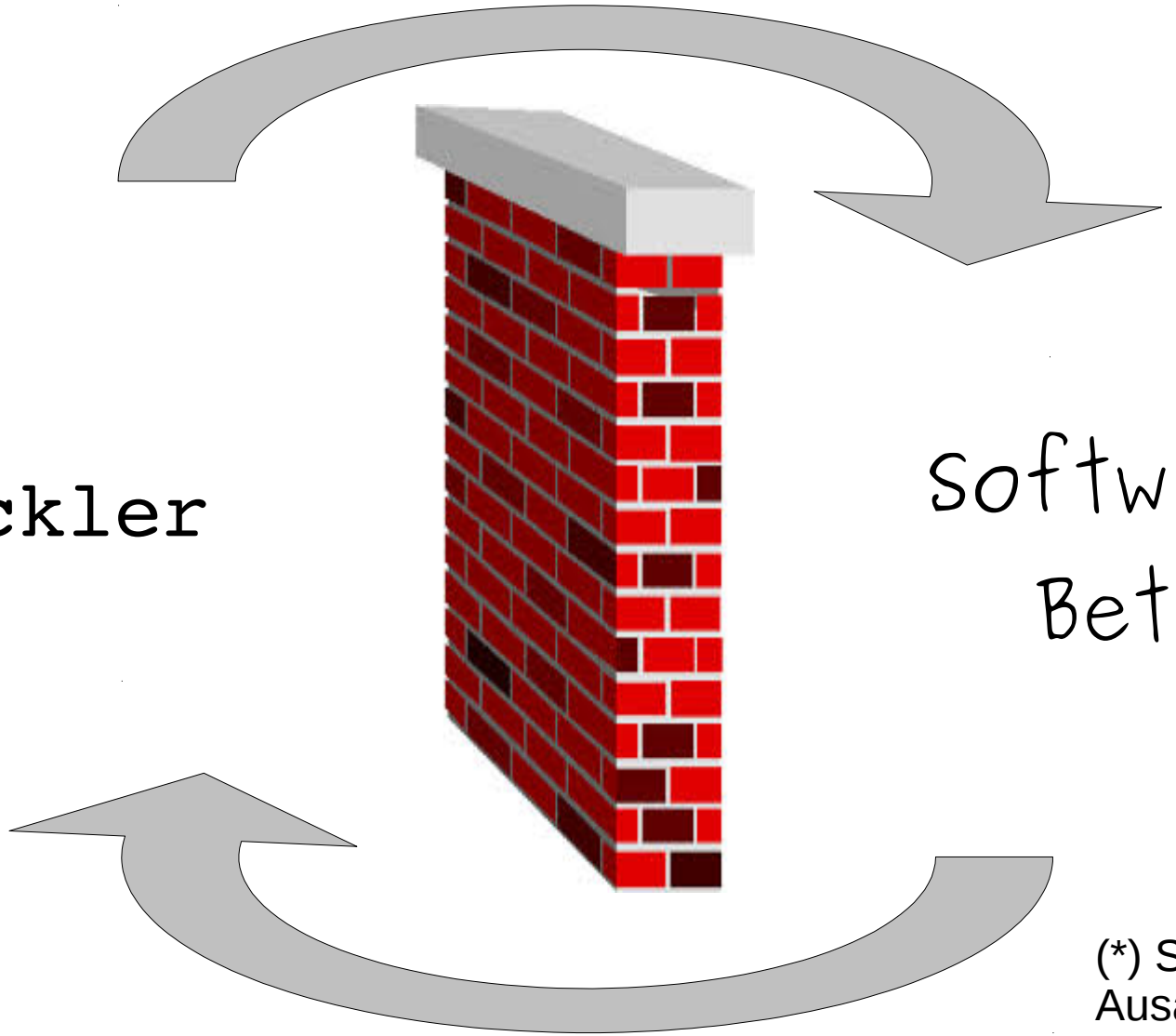
- Team IS24 Anbieten Profi (AP)
  - 1-wöchige Testphase mit festem Deadline-Tag in der Woche
  - 8-15 Tage zwischen Änderung und Veröffentlichung
  - Keine Verantwortung für Betrieb
  - Wenig Gefühl dafür, was die Software gerade macht
- Wenige Faults, Testabteilung findet die meisten

Ergebnisse

- Teams IS24 Online Marketing (OM) und SC Payment (PM)
  - Deployment nach Abnahme durch PO
  - 2-3 Tage zwischen Änderung und Veröffentlichung
  - Überwachen Applikation selber
  - Bei Probleme mit Anwendung sie verantwortlich
  - Payment auf Bereitschaft, Online Marketing nicht, aber verantwortlich für Deployments
  - Gute Qualität, POs zufrieden, keine offenen Faults, schnell gefixt

Freie Universität Berlin

Das Qualitätserlebnis (*)

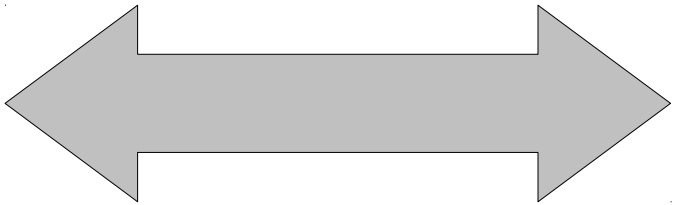


(*) So nicht in der Ausarbeitung enthalten

Freie Universität Berlin

Das Qualitätserlebnis (*)

Entwickler



Software im Betrieb

- Schnell
- Direkt
- Unmittelbar

(*) So nicht in der Ausarbeitung enthalten

Freie Universität Berlin

Das Qualitätserlebnis

- Bei PM ist das Firmenkultur
- *"Der Entwickler ist verantwortlich für das was er macht. Und zu keiner Zeit denkt er, er kann den Code schreiben und sich darauf verlassen, dass jemand anders den fixt." (Entwickler Team PM)*

Freie Universität Berlin

Das Qualitätserlebnis (*)

- Bei Team OM bis vor 2 Jahren ebenfalls „Wand“
- Stückweise abgebaut, jetzt wie bei PM
- „*Wir haben gemerkt, dass das [ohne nachgelagerte Tester, die Arbeit] anders macht. Man hat niemanden zum blamen, wir habens gemacht. Dadurch nimmt man jeden Fehler der draußen aufschlägt persönlich. Dadurch trägt man mehr Verantwortung. Vorher konnte man sich wegducken [...]" (Entwickler Team OM)*

Freie Universität Berlin

Das Qualitätserlebnis (*)

- Führt zu anderer Einstellung und Verhalten der Entwickler

- *"Solange du noch nie erlebt hast, wie es ist sofort damit live zu gehen, [bleibt die Diskussion (um Qualität) theoretisch]. [Sofortiges Deployment] würde der Diskussion eine ganz andere Qualität geben, und man würde über andere Sachen sprechen. Z.B. wieviele Tests brauchen wir eigentlich oder was gehört alles dazu um eine Story fertig zu machen." (Scrum Master Team AP)*

Auswirkungen – Prävention (*)

- Starkes Qualitätserlebnis führt zu mehr Prävention (OM und PM)
- *"Und ich weiß auch, dass unter Umständen wenn es einen Fehler gibt, werde ich Nachts um 2 Uhr geweckt. Das ist auch schon genug Motivation zu sagen: Dann teste ich lieber doppelt-dreifach und stelle sicher, dass ich nicht Nachts geweckt werde." (Entwickler Team PM)*

- Hohe, interne Qualität
- Schnelle, aussagekräftige Tests
- Häufige, kleine Veröffentlichungen

Auswirkungen – Eindämmung (*)

- in beiden Teams 5-Uhr Regel
- Überwachung
- Automatisches Deployment
- Bereitschaftsdienst

- *„Meistens ist es gar nicht das wecken [Nachts], sondern die Implikationen, dass die Leute gar nicht upgraden können. Das wolllen wir ja verhindern. [...] Ich hab ja den Anspruch, dass das System läuft.“ (Entwickler PM*)

Auswirkungen – Motivation (*)

- Verzögerung frustrierend für Entwickler
- *„Ja. Gerade bei Bugs ist es sehr ärgerlich. Mich ärgern die Bugs auch. Ich habe manchmal das Gefühl: 'Was sollen die Kunden denken. [...] Und dann braucht das noch 2 Wochen [bis es live ist]" (Entwickler Team AP)*
- *„Man hat das ja noch im Kopf, was man letzte Woche getan hat, aber du hast nicht mehr im Kopf, was du vor 3 Wochen getan hast. Daran müsste man sich bei einem [Fault] erinnern. [...] Das ist sehr weit weg.“ (Entwickler Team AP)*

Auswirkungen – Motivation (*)

- Motivierend für Entwickler
- *„aus diesen Erkenntnissen heraus [... selber QA zu machen], haben wir gesehen dass wir ein bisschen mehr können. Und durch schnelle Iterationen haben wir ein bisschen Luft. Dadurch [können wir] Ideen formulieren und auch ein bisschen austesten. [Dahinter steht] der Wunsch nicht soviel Routine machen zu müssen“ (Entwickler Team OM)*

Freie Universität Berlin

Literatur

- Nachgelagerte Testphase ist „traditionelles Testen“
- Viele Vor- und Nachteile von parallelem „agilen Testen“ genannt
- Oft dabei, dass Testen niedrigeren Stellenwert als Entwicklern hat
- Konnte ich nicht beobachten
- Kaum Erfolgsfaktoren, nur dass Einstellung der Entwickler wichtig ist
- Kein Bezug zu Veröffentlichung von Änderungen (CD ist relativ neu)

- Hypothese: Qualitätserlebnis mögliche Erklärung für Erfolg- und Nichterfolg

Freie Universität Berlin

Schluss (*)

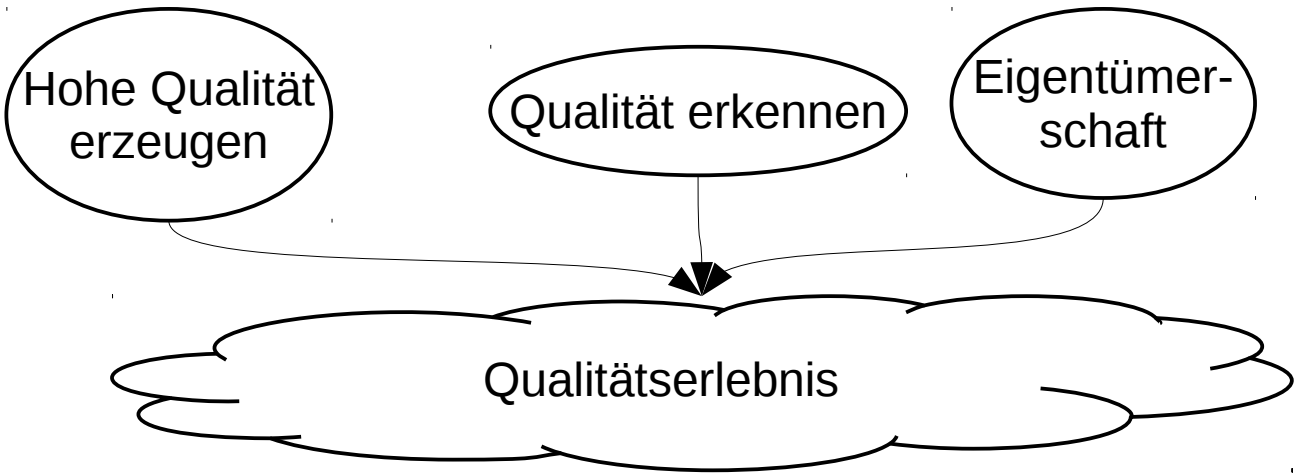
- Qualitätserlebnis durch schnellen, direkten, unmittelbaren Informationsfluss
- Prävention und Eindämmung statt nachgelagertem Testen
- Sehr motivierend für Entwickler
- Deutlich flexibler als Testphase

Freie Universität Berlin

Fragen?

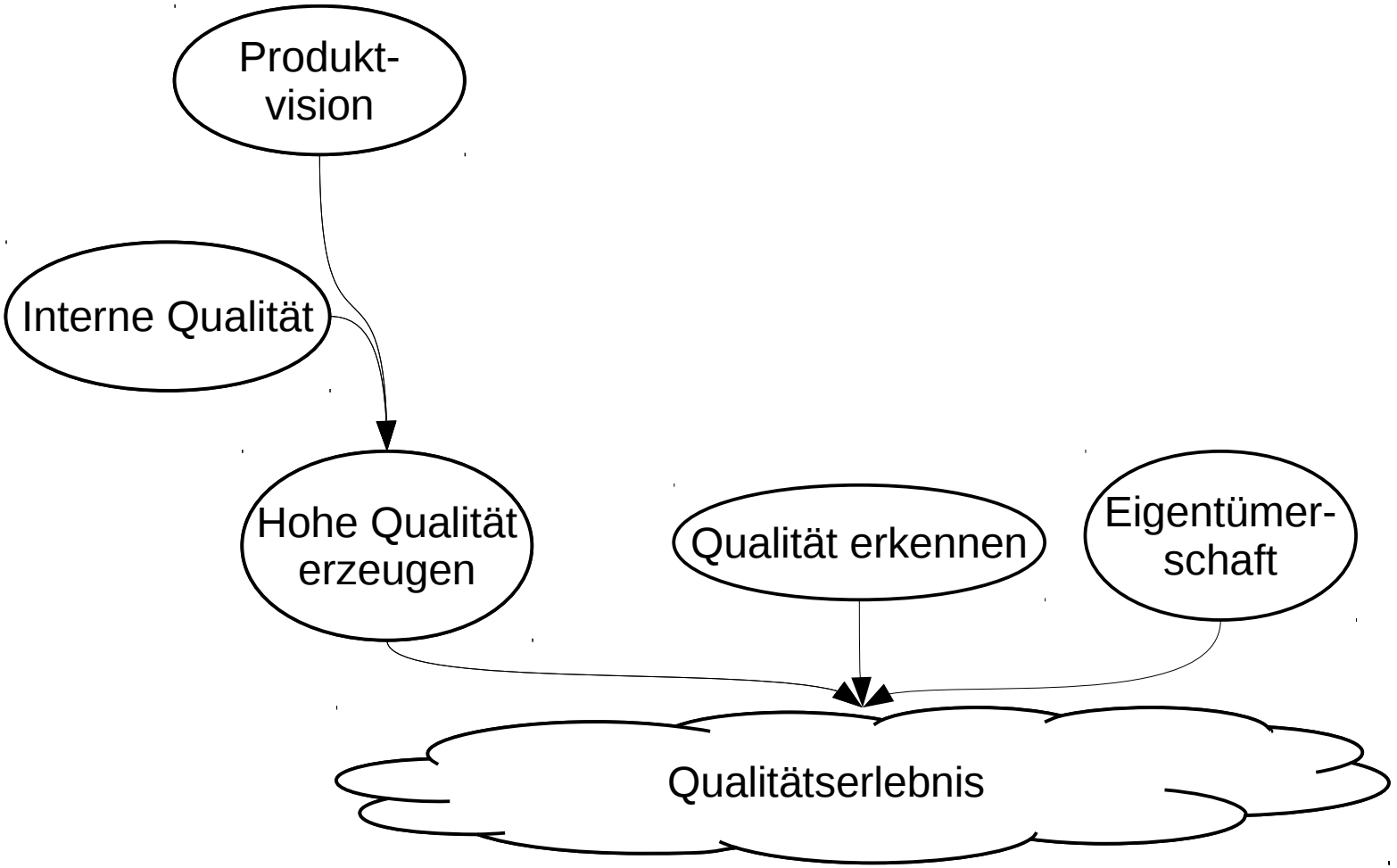
Freie Universität Berlin

Erfolgsfaktoren



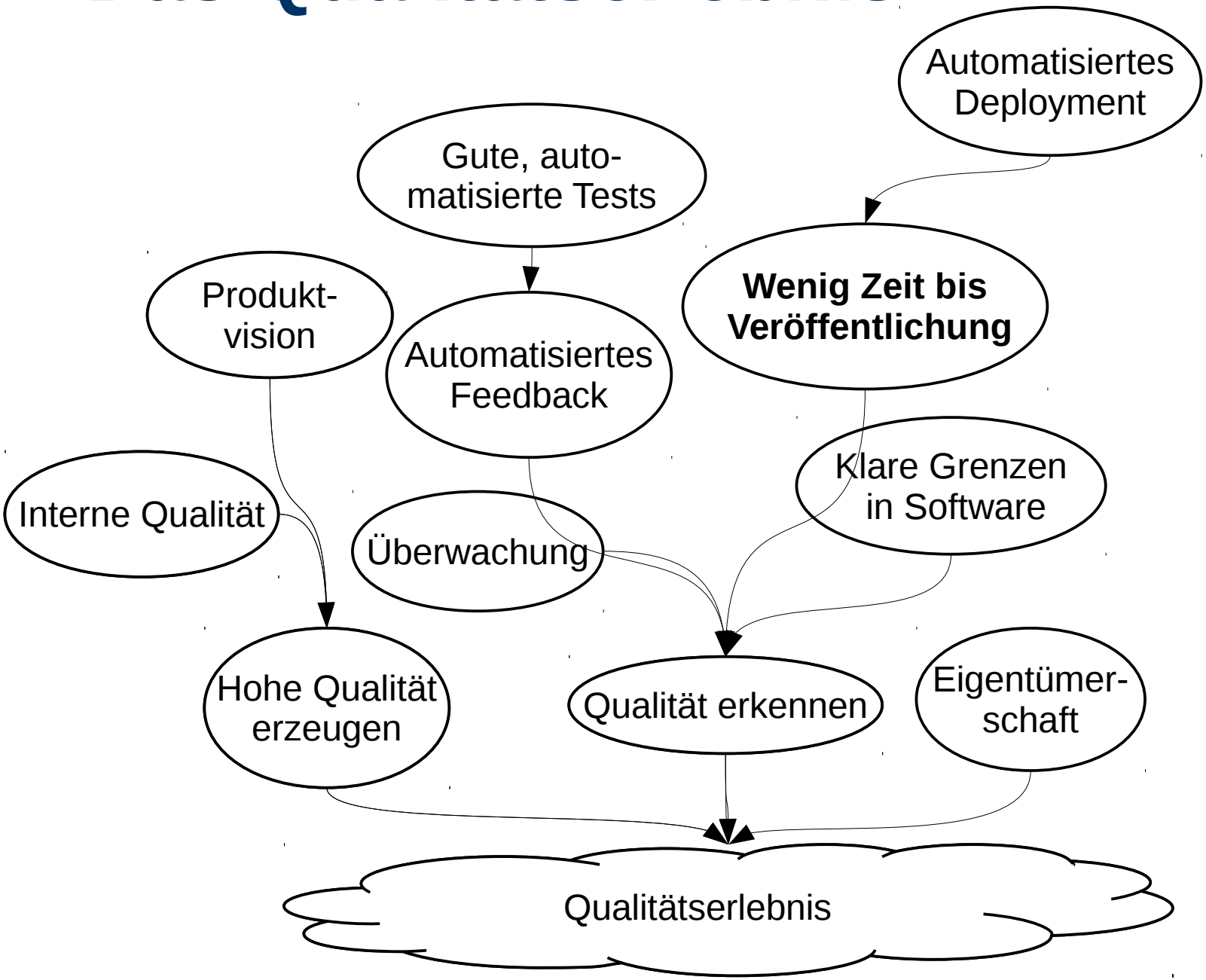
Freie Universität Berlin

Das Qualitätserlebnis



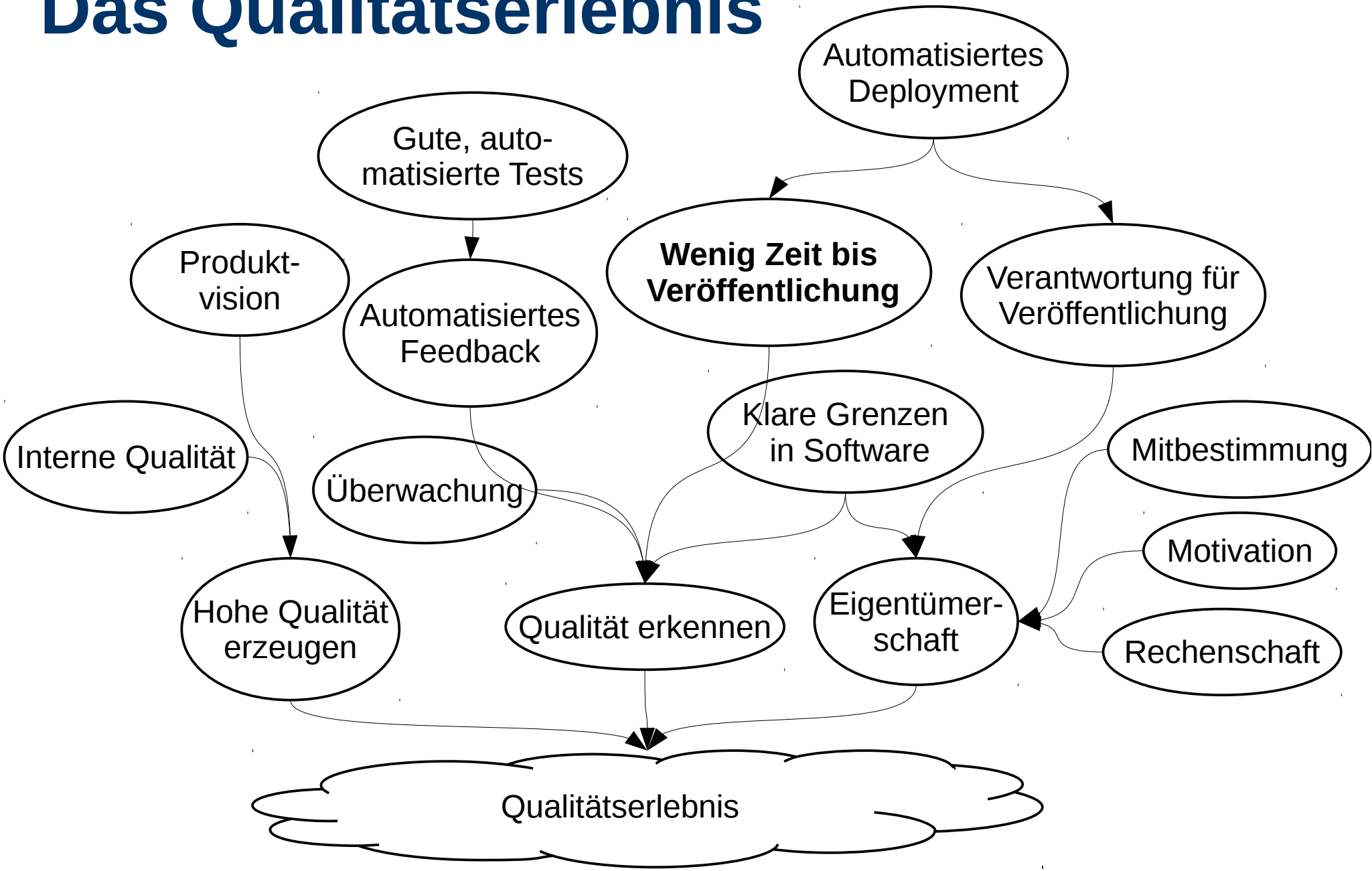
Freie Universität Berlin

Das Qualitätserlebnis



Freie Universität Berlin

Das Qualitätserlebnis



Freie Universität Berlin

Das Qualitätserlebnis