

Team? Welches Team?

Scrum hat uns versprochen, dass wir ein festes Team haben und jeder in diesem Team irgendwann zwangsläufig alle Aufgaben im Team zu ungefähr 80% übernehmen kann: UX, Coding, Administration, sein tiefes Fachwissen jedoch weiter pflegt. Gibt es diese "T shaped persons" oder gibt es eventuell doch einen Grund, warum wir noch Spezialisten haben? Und wie soll oder kann ein agiles Team eigentlich aussehen?

Nachdem aus den klassischen Ops-Leuten "Infrastructure Coder" geworden sind, nachdem Entwicklerteams YBIYRI (You build it, you run it) machen, nachdem wir alle in der Cloud sind - nachdem wir also alle T-Shaped-Persons geworden sind, die tief in ihrer Fachlichkeit verwurzelt sind, aber dennoch ein breites Wissen der restlichen benötigten Skills haben, seitdem können wir als festes kleines Team eine Applikation von ihrer Entstehung bis zu ihrem Betrieb vollständig betreuen.

Ist das so?

Oder werden nicht immer noch Netzwerkspezialisten, Systemingenieure, Stagespezialisten, Ops-Leute und Java-Entwickler eingestellt? Benötigen wir eventuell andere Fachrichtungen in unseren Teams? Wie sieht es mit rechtlichen Fragen aus? Wer designed unsere Nutzerschnittstellen für uns?

In welchen Sprachen coden wir eigentlich unsere Infrastruktur - und wer beherrscht diese Sprachen? Wer arbeitet an der Problemlösung, wenn die Hütte brennt? Und wer schafft es, Infrastruktur so zu designen, dass die Hütte kontrolliert abbrennt und alle wissen, an welchen Stellen sie effektiv löschen können? Wer übernimmt die Verantwortung für die Applikationen oder den Infrastruktur-Stack der dabei entsteht?

In diesem Talk betrachte ich die verschiedenen Team-Setups, die in einem agilen Umfeld möglich sind: Feste Teams von Generalisten, feste Teams von Spezialisten, aber auch dynamische Teams, die sich je nach gefordertem Thema zusammensetzen. Ich werde die Vor- und Nachteile der einzelnen Setups vorstellen, und zeigen, wo sich welche Teamstruktur bewährt hat - und wo sie eher problematisch ist.