



Blankziehen

Warum fällt es uns manchmal so schwer,
Wissenslücken zuzugeben?

Ein Vortrag vor 50 Leuten. Der Vortragende wirft mit Fachausdrücken um sich, von denen ich die meisten kenne, aber bei „Idempotenz“ ist es aus mit meinem Wissen – davon habe ich keine Ahnung. Doch statt zu fragen, hole ich das Smartphone heraus und sehe nach. Mein danach dürres Verständnis wird bald wieder dahin sein. Und somit wird von diesem Vortrag nur die Erinnerung an eine Niederlage übrig bleiben. Ich wusste mal wieder nix.

Warum habe ich nicht den Referenten gefragt? Der hätte das sicher erklären können, und wenn nicht, hätte ich nachgehakt oder ein anderer der Anwesenden wäre eingesprungen und hätte es mir in seinen Worten nahegebracht. Aber ich war zu feige.

Wie sähe das aus, wenn der Chefredakteur einer Zeitschrift für Softwareentwicklung nicht weiß, was Idempotenz ist?

Dann wäre mein Renommee dahin und das der Zeitschrift auch. Später im Gespräch mit anderen Teilnehmern stellte sich heraus, dass auch sie keine Ahnung hatten und sich genauso wenig zu fragen trauten wie ich. Wie sähe das aus, wenn der Senior Software Engineer nicht weiß, was Idempotenz ist, während seine Kollegen mit im Vortrag sitzen. Wie doof ist das nur? Statt Wissenslücken zu schließen, will man nur ja nicht als der Unwissende dastehen.

Es fällt offenbar schwer, selbstbewusst mit Lücken umzugehen. Dabei gibt es doch keine dummen Fragen – wurde uns ja schon als Kindern eingetrichtert. Freilich ergibt es keinen Sinn, bei 50 Fachbegriffen 50 nachzufragen. Aber meist handelt es sich ja nur um wenige Sachverhalte, die einem neu sind.

Aus einer anderen Perspektive betrachtet adelt Unwissen den Fragenden sogar. Man stelle sich Software-Guru XYZ vor. Ihm unterstellt man geballtes Wissen. Fragt er nun nach, tut das seinem Renommee keinen Abbruch. Im Gegenteil: „Toll, der traut sich, seine Wissenslücke selbstbewusst zu zeigen“, denkt man dann, und vielleicht noch: „Auch so einer weiß eben nicht alles“ oder: „Ich hätte nicht gedacht, dass er das nicht weiß“. Ich habe mir auf jeden Fall vorgenommen, das nächste Mal zu fragen. Dann oute ich mich vielleicht als Unwissender, helfe aber mir und anderen, die sich nicht zu fragen trauen.

Übrigens ergibt eine idempotente Funktion bei mehrfachem Aufruf immer dasselbe Resultat. Eine lesende Funktion auf eine Datenbank sollte also immer idempotent sein und nicht plötzlich andere Werte liefern.

Viel Spaß mit der dotnetpro, die hoffentlich Ihre Wissenslücken schließt!

Tilman Börner
Chefredakteur dotnetpro



Christian Jacob

bringt Ihnen mithilfe von
Tic Tac Toe das Test Driven
Development näher (S. 42)



Ralf Westphal

hat ChatGPT auf den Zahn
gefühl, ob es sich als Entwick-
lungstool eignet (S. 82)



Christian Havel

weiß Cosmos DB als NoSQL-
Datenbank in der Cloud einzu-
setzen (S. 114)