

Changing an existing eVC for new project requirements is a grueling task. It's really painful for any eVC developer to witness his creation being torn apart by an end user, but a little prudence from the developer can result into longevity of eVC. Want to learn how to do it? We'll get there in a minute.

But first, I'd like you guys to meet M.P. Rashid - *Think Verification*'s first guest author. He is a verification engineer (how else) at Tata Elxsi in Bangalore. He's really enthusiastic about verification with Specman and a regular reader of Think Verification (way to go!). Today he's going to talk about using hook methods (hooks) efficiently, which is a concept we should all get familiar with.

Before we get started - just a quick reminder that if you would like to share **your** thoughts about functional verification, *Think Verification* warmly welcomes guest authors and bloggers.

Ok, fasten your seatbelts folks - you're about to Get Hooked:

[Download the article \[pdf\]](#)

[Download code example \[e file\]](#)