

Single-Responsibility-Prinzip

„Es sollte nie mehr als einen Grund dafür geben, eine Klasse zu ändern.“ Robert C. Martin

- Dieses Prinzip besagt, dass jede **Komponente** **nur für eine Aufgabe verantwortlich** sein und auch die gesamte Funktionalität einer Aufgabe abdecken sollte.
- Werden **Änderungen** an der Software vorgenommen, sollte es einen **Grund** dafür geben.
- In der Praxis hat eine Komponente dann die richtige Größe, wenn sich die **Änderungen** aufgrund einer User Story **nur auf eine Komponente auswirken**.
- Die **Kernaussage** des Prinzips ist, dass **jede Klasse nur genau eine fest definierte Aufgabe zu erfüllen** hat. Wenn eine Klasse mehrere Verantwortungen zu tragen hat, führt das zu Schwierigkeiten bei zukünftigen Änderungen und das Fehlerrisiko steigt.
- Eine **hohe Kohäsion** – alle Methoden innerhalb einer Klassen haben einen starken gemeinsamen Bezug – **sollte angestrebt werden**.

