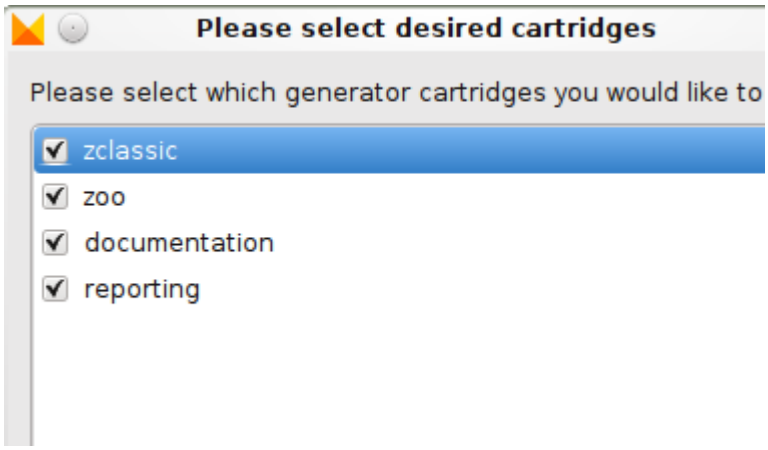

Von Scaffolding und UML zu MDSD und DSL

Jedes Framework soll dem Entwickler Arbeit abnehmen und helfen bestimmte Regeln einzuhalten. Auch wenn die intern verwendeten Paradigmen und Funktionen unterschiedlich ausgestaltet sind, verfolgen alle Frameworks das gemeinsame Ziel eine schnelle, einfache, flexible und effiziente Entwicklung individueller Anwendungen zu ermöglichen. Mit Schlagworten wie MVC-Muster, Komponenten oder APIs wird um die Gunst der Entwickler geworben. Oft wird mit einer so genannten Scaffolding-Funktion die Möglichkeit angeboten, mit minimalem Aufwand einen Prototyp für eine eigene Erweiterung zu generieren.
Häufig unbeachtet bleibt jedoch die Frage nach der langfristigen Wartbarkeit. Die einmal erstellten Anwendungen müssen bei Änderungen des verwendeten Frameworks aktuell und konform gehalten werden. Mit der Zeit entstehen mehr und mehr Anwendungen, die regelmäßig an neue Versionen und damit verbundene architektonische Änderungen angepasst werden müssen. Dies betrifft Entwickler von freien Modulen ebenso wie Agenturen, die Anwendungen für ihre Kunden warten.
Früher wurden modellbasierte Ansätze wie die UML zur Beschreibung und Dokumentation von Softwaresystemen verwendet. Jedoch scheiden diese Verfahren als intuitive Notationen aus, da sie sehr technisch und komplex aufgebaut waren.
Bei der modellgetriebenen Softwareentwicklung (MDSD – Model-Driven Software Development) wird eine Anwendung nicht einfach durch ein Modell beschrieben, sondern komplett durch ein Modell verkörpert. Die Modellierungssprache ist dabei auf einen bestimmten Problembereich zugeschnitten, man spricht hierbei von einer domänenspezifischen Sprache (DSL – Domain Specific Language). Dadurch bekommt die Sprache einen formalen Charakter und verleiht ihren Elementen eine entsprechende Semantik. Dies ist Voraussetzung für eine automatisierte Verarbeitung von Modellen, zum Beispiel für Modell-zu-Modell-Transformationen und dem Generieren von Quelltexten oder anderen Dateien, wie Dokumenten oder Konfigurationseinstellungen.
Mit der modellgetriebenen Entwicklung kann erreicht und sogar gewährleistet werden, dass alle Anwendungen den qualitativen Ansprüchen des

Frameworks genügen. Und zwar dauerhaft und ohne unnötige Fleißarbeit für die mehrmalige Lösung wiederkehrender Probleme. Neben der gestiegenen Geschwindigkeit bei der Entwicklung profitieren außerdem Wartbarkeit und Softwarequalität signifikant von diesem Ansatz. Änderungen im Generator können alle darauf basierenden Anwendungen automatisch nutzen. Das Projekt Zikula profitiert ebenfalls. Mit der Zeit können sich die Vorteile vervielfachen.

https://modulestudio.de/