

Verstehen Sie Ihre eigenen Programme besser

Dokumentation von Code durch Designdokumente und Kommentare ist notwendig für das bessere Verständnis eigener oder fremder Programme. Praxiserfahrene Entwickler und Projektmanager wissen aber aus eigener Erfahrung, dass die Dokumentation nach einigen Änderungszyklen oft nicht mehr mit dem Quellcode übereinstimmt. In diesen Fällen sind Softwaretools hilfreich, die die Abhängigkeiten und die Architektur im Code visualisieren und damit zum besseren Verständnis beitragen.

Jeder Programmierer kennt dieses Szenario: Anfänglich hat ein Modul (oder eine Klasse bzw. ein anderer gekapselter Teil des Codes) eine saubere und klare Struktur und ist somit recht einfach zu verstehen. Einige Zeit später muss das Modul auf die eine oder andere Art geändert werden. Mit einigen Verbesserungen erfüllt dieses Modul nun auch diese Anforderungen. Was für ein tolles Design! Zwar erforderten die Optimierungen einige Kompromisse, aber wir sind sicher, dass wir uns beim nächsten Mal - wenn wir wieder Änderungen machen müssen - an diese wenigen Fallstricke erinnern werden. Und dann, wenn wir später schon in einem anderen Projekt arbeiten, wird eine dringende Änderung im alten Projekt erforderlich. Natürlich – so denken wir - wissen wir noch alles über den früher von uns entwickelten Code und können die Änderung somit problemlos vornehmen. Leider gibt es dann aber oft böse Überraschungen.

Die Schwierigkeit beim Fixen oder Verbessern unseres eigenen Codes besteht darin, dass wir in eine Falle tappen, weil wir glauben, alles über „unseren“ Code zu wissen. Sicherlich erinnern wir uns gut an das „Große und Ganze“, aber das eine oder andere Detail haben wir dann doch aus den Augen verloren. Vielleicht haben Kollegen in der Zwischenzeit bereits Änderungen gemacht, über die wir nicht informiert wurden, was unsere Aufgabe nicht einfacher macht.

In dieser oft auftretenden Situation sind Softwarewerkzeuge wie Imagix 4D, welche den Source-Code analysieren, sehr hilfreich. *** Es gibt die grundlegende Suche nach allen Verwendungen einer Funktion, Klasse oder Variablen. Während diese textbasiert ausgeführt werden können, sind semantisch gesteuerte Suchvorgänge vorzuziehen, die die Bedeutung von Identifiern kennen und vermeiden, uns mit irrelevanten Codezeilen zu überfluten.

Ebenfalls sind High-Level-Abstraktionen hilfreich. Selbst wenn unser Interesse eine einzelne Funktion ist, kann uns die Verwendung dieser Funktion in architektonischen Ebenen daran erinnern, dass es einige Verwendungen gibt, die über das ursprüngliche Design hinausgehen. Es wird uns auch zeigen, dass wir die Funktion einmal über mehrere Ebenen hinweg verwendet haben, da es sich um

eine Notfallkorrektur handelte und dies der schnellste Weg war, sie zu lösen. Typische Änderungen gehen normalerweise über eine einzelne Funktion, Variable oder Klasse hinaus. Was wir in diesem Fall sehen möchten, ist die Menge aller primär interessierenden Einheiten und aller Bereiche, die möglicherweise von ihnen betroffen sind. Anschließend können wir die Einsatzbereiche erkunden und einen Drilldown durchführen, um sicherzustellen, dass wir die Auswirkungen der Änderung verstehen.

Eine weitere wichtige Information ist das Verständnis der jüngsten Änderungen im Code. Dies geschieht auf einfachste Weise durch Textvergleiche von Quelldateien. In den meisten Fällen wird die Masse von Informationen aber sehr hoch sein, so dass das Durcharbeiten aller Änderungen sehr umständlich wird. Durch einen Vergleich der verschiedenen Code-Versionen auf einer höheren Ebene und einem Drilldown mit geeigneten Werkzeugen, können Änderungen relativ einfach überprüft werden.

Erfahrene Programmierer wissen, dass mehr als 50% des Entwicklungsaufwands in die Nacharbeit der entwickelten Software fließen. Während für die initiale Entwicklung von Software in der Regel Budget für Tools zur Verfügung steht, gibt es in der späteren Wartungsphase oft Widerstand, auch in nützliche Tools zu investieren, um die Software im Laufe der Zeit zu warten. Investitionen in Software-Analysertools helfen allerdings auch nach Abschluss der initialen Entwicklung dabei, Zeit und Kosten zu sparen und unbeabsichtigte Nebenwirkungen von Änderungen des ursprünglichen Codes zu vermeiden. Selbst für die besten Programmierer ist es schließlich unmöglich, sich an alle Details in der Software zu erinnern.

Ein gutes Software-Werkzeug, welches Sie bei dem Verständnis eigener und fremder Programme unterstützt ist Imagix 4D der kalifornischen Imagix Corp.

Weitere Informationen zum Tool sind verfügbar unter

https://www.verifysoft.com/de_imagix4d.html

Übersetzung eines Blogbeitrags, Quelle: <https://www.imagix.com/blog/re-familiarizing-yourself-with-your-own-programs/>

Verifysoft Technology, www.verifysoft.com - 2020