

Den eigenen Hafen kennen, bevor man zu neuen Ufern aufbricht

Jon Erik de Linde
PITSS GmbH
Stuttgart

Schlüsselworte

Projektabschätzungen, ,Analyse, Projektplanung, Ist-Analyse, Metriken, Qualität, Komplexität

Einleitung

Egal ob Ihr eine neue Standardsoftware implementiert, eine neue Entwicklung in einer anderen Technologie vorhabt oder die vorhandene Softwarelösung verbessern und sichern wollt – es ist immer wichtig die derzeitige Situation genau zu kennen. Dies gilt sowohl für die eingesetzte Softwarelösung als auch für das Projektumfeld. Die eigene Situation neutral beurteilen zu können hilft bei jeder Planung und Abschätzung von Projekten.

Wieviel Analyse ist für eine gute Projektplanung notwendig? Ab wann wird die Analyse zum Selbstzweck? Wie sieht meine Anwendung aktuell aus?

Anerkannte Software Metriken wie Halstead und McCabe usw. können helfen die Qualität der Anwendung neutral zu bewerten und hilft dabei Schwachstellen zu erkennen.

Anhand von zahlreichen Projekten rund um Oracle Forms, ADF und APEX während den letzten Jahren hat sich immer wieder feststellen lassen wie unerlässlich eine Ist-Situationsanalyse ist.

Analysemöglichkeiten und Mertriken für Softwarequalität

Abschätzung in Projekten ist immer eine Herausforderung.

„Estimating the cost of software is, at best, an educated guess.” Kane Mar

(<http://scrumology.com/estimate-the-total-cost-of-agile-projects/>, 30.09.2014)

In “Schätzen in Agilen Projekten” schlägt Boris Gloger einige gute Methoden um Projekte genauer abschätzen zu können.

Unsere Kunden wollen in vielen Fällen eine Planbare Größe wie viel Investment eine Änderung kosten wird. Mit einem fixen Preis hofft der Kunde das Risiko auf die Seite des Dienstleisters zu schieben. Jeder Dienstleister wird die Unschärfe in der Abschätzung auf den Preis aufschlagen und so wird der Kunden in den meisten Fällen mehr für eine Dienstleistung zahlen als benötigt wurde.

Ist in Softwareprojekten eine Investitionsplanung möglich? Aus unserer Erfahrung ja – aber genau darauf beruht es. Wenn für eine Entwicklung, ein Prozess oder Projekt vergleichbare Daten vorliegen kann und sollte Erfahrungen verwendet werden um Projekte besser und genauer abschätzen zu können. Eine Innovation, eine Pionierarbeit, eine Revolution lässt sich in vielen Fällen nicht abschätzen.

In vielen unseren Projekten sind wir als Modernisierungspartner gefragt und überführen eine Anwendung (in unserem Fall in den meisten Fällen von Oracle Forms) von einer Technologie zu einer anderen.

Die vorhandene Anwendung dient in diesen Fällen oft als Spezifikation und Blueprint für die neue Entwicklung. Die Kunden haben über Jahre eine Anwendung entwickelt der maßgeschneidert ist für Ihre Bedürfnisse (einer der Vorteile der Individualentwicklung).

Wir haben daher für unsere Abschätzungen die Möglichkeit geschaffen die vorhandene Anwendung zu Analysieren – einen Status Quo der Komplexität, der Architektur und der Abhängigkeiten der Anwendung zu identifizieren.

Die Ermittlung anerkannte Metriken wie McCabe's Cyclometric Complexity, Halsted's Volume und die Maintainability Index hilft es die Anwendung anhand neutralen Zahlen zu bewerten.

Diese Komplexitätsmessungen fließen unter anderen in die Aufwandabschätzungen für eine Migration von Forms nach ADF.

Analysemöglichkeiten erlauben es uns aber auch nicht nur die Abschätzung genauer durchzuführen.

Für die Durchführung einer Migration ist die Erstellung von UserStories eine wichtige und oft Zeitaufwändige Aufgabe. Durch das Aufzeichnen von User initiieren Prozessen, haben wir die Möglichkeit geschaffen UserStories einfach und genau festzuhalten.

Mit der Zeit sind viele Kennzahlen zusammengekommen welche Ermittelt werden – eine Erweiterung ist wahrscheinlich. Die Kennzahlen sind u.a.:

Kennzahlen aller Objekte einer Oracle Forms Anwendung.

Dies beinhaltet Anzahl der Datenbankobjekte, Forms, Libraries und Reports sowie deren einschließenden Objekte wie z.B. Windows, Canvases, Blöcke, Items, Programmeinheiten, etc.

Lines of Code (LOC) / Lines of Documentation (LOD)

Gesamtübersicht über die Verteilung des Codes und der Kommentare über die gesamte Anwendung unterteilt nach DB Ojekten und Anwen-
dungskomponenten wie Forms und Reports.

Durchschnittswerte wie Zeilenlänge, Zeilenanzahl, Variablen, Argumente je Programmeinheit.

Allgemeiner Aufbau der Programmeinheiten mit Betrachtung der Programmblöcke, anonymen Programmblöcke und Exception Blöcke.

Übersicht, die zeigt, wo sich die Programmlogik (z.B. Validierung, Navigation, etc.) bzw. die Business Logic (DML) befindet.

Betrachtung der Objekte in Relation der oben aufgeführten Kennzahlen, d.h. wo werden welche Objekte wie verwendet.

Allgemeine Übersicht über die Anzahl der komplexen PUs bewertet nach Anzahl der Statements in %.

Detaillierte Bewertung der Kommentare in den Programmeinheiten

Betrachtung möglicher Redundanzen und Referenzen

Verwendung von Libraries, Globalen Variablen, Package Variablen

Betrachtung des Source Codes nach den allgemein anerkannten Metriken:

Halstead

McCabe

Maintainability Index

Überprüfung bestehender Namenskonventionen

Einführung neuer Namenskonventionen

Obsolete Code

Obsolete Objects

Ihre

Kontaktadresse:

Erik de Linde

PITSS GmbH

Industriestr. 3

D-70565 Stuttgart

Telefon: +49 (0) 711-91401215

Fax: +49 (0) 711-91401211

E-Mail: edelinde@pitss.de

Internet: www.pitss.de