

Schlußbewertung

Zusammenfassung / Durchführung

Aufgrund der Vergrößerung des Teams von zwei auf vier Mitglieder (zu Sebastian Sauer und Tobias Krebs sind Sascha Kupper und Bernd Steindorff gestoßen), ergab sich eine längere Einarbeitungsphase, welche zum Kennenlernen der Techniken und des Codes genutzt wurde.

XP

Pair-Programming / Einarbeitung

Die ersten vier Wochen wurden für das Kennenlernen des Quellcodes von KoMacro und deren Zusammenhang genutzt. Bereits hier ist intensiv über das Design nachgedacht und diskutiert worden. Die ersten Refactorings wurden vorgenommen, sowie eine Aufteilung der Zuständigkeiten innerhalb des Projekts.

So hat man am Anfang eine größere Phase der Einarbeitung gehabt, welche zeitintensiv war. Relativiert hat sich das jedoch, als die Gebiete abgegrenzt wurden und das Wissen zum eigenständigen Arbeiten in Teams vorhanden war.

Als geeignetes Mittel hat sich hier das Pair-Programming herauskristallisiert. So konnte das Know-how gut übertragen werden und in den Fachgesprächen schnell bessere Konzepte für vorhandene Lösungen eruiert werden ($1+1 > 2$).

Refactoring

Besonders in SWPII stand das Refactoring des in SWPI entstandenen Quellcodes im Vordergrund. Neben dem anfänglichen Code-Review zur besseren Einarbeitung wurden insbesondere die Konzepte zur Persistierung, Wartbarkeit, als auch der kompletten Abbildung der einzelnen Aktionen vollständig überarbeitet (hierzu gibt es ein spezielles Sequenzdiagramm Anhang 3).

In Bezug auf die Einbindung in Kexi stand genauso eine Überarbeitung an, die zusätzlich durch den Gedanken der weiteren Portierung in die KOffice-Libs geleitet wurde.

Tests

Die großen Refactorings haben nach sich gezogen, dass die Tests von SWPI zum Teil unwirksam wurden, nach Abschluss des jeweiligen Abschnitts aber sofort wieder aktualisiert wurden. Insbesondere die Tests des Umgangs mit den Variablen sowie der Persistierung haben Schwächen bei der Implementierung aufgezeigt, welche direkt behoben wurden und so zu einem sauberen Framework geführt haben.

Hierzu wurde das KUnitTest-Framework verwendet und mit eigener Funktionalität über C-Makros erweitert. Zur Visualisierung der Testabdeckung gibt es eine Kommandozeilenausgabe sowie eine Test-GUI.

C++, Qt, KDE und Linux

Die verwendete Sprache im KDE-Projekt ist C++, was zu einigem Einarbeitungsaufwand innerhalb des Teams geführt hat, da an der TFH-Berlin im Bereich Medieninformatik Java vorherrschend ist. Die Frameworks Qt und KDE sind den neuen Teammitgliedern vollständig unbekannt gewesen, weshalb hier nochmal Zeit investiert werden musste.

Die Implementierung geschah unter Linux, was insbesondere im Netzwerk der TFH-Berlin (OpenNet) zu großen Problemen geführt hat, da die Unterstützung für Linux-Clients nicht dokumentiert ist und

auf persönliche Anfrage eine qualifizierte Auskunft ausblieb

Seiteneffekte in KOffice und Kexi, SVN, Kommunikation

Durch die Benutzung des KDE-SVN's sind zahlreiche Seiteneffekte aufgetreten, da jeder SVN-Benutzer Änderungen einbringen konnte. Insbesondere in Kexi, von welchem wir wegen der Einbindung als Plugin abhängig waren, gab es zahlreiche Updates, die einen Kontakt zu anderen Programmieren bzw. dem Maintainer bedurften. Die Kommunikation war über viele Kanäle wie IRC, Mail (und -listen) und ICQ erforderlich. Sämtlicher Austausch auch in Bezug auf die Kommentare im Quellcode mussten in englischer Sprache erfolgen. Bei über 1.500 Entwicklern im KDE-Projekt ist die Benutzung dieser weit verbreiteten Sprache Pflicht gewesen.

Manche Probleme ließen sich auch selbst beheben, haben aber unabhängig von KoMacro Zeit gekostet.

Ausblick

In KOffice 1.6, welches im September 2006 als finale Version erscheinen soll, wird KoMacro im Rahmen von Kexi ausgeliefert.

Die weitere Planung sieht eine Portierung von QT3 nach QT4 und Integration in die Koffice-Libs zur Wiederverwendung in weiteren Anwendungen wie Krita vor.