

Erfahrungsbericht Gruppenarbeit

Die Arbeit in einer Gruppe zeigt, wie gut oder schlecht sich Menschen aufeinander einstellen können.

Leider mussten wir die Erfahrung machen, dass sich trotz Absprache und Aufteilung der Aufgaben eine Gruppenarbeit nicht leicht gestalten lässt, wenn nicht alle Teammitglieder das gleiche Engagement für die Arbeit aufbringen. So besteht die Gefahr, dass die Arbeit nicht fertig wird.

Da wir bemüht waren unsere Gruppe aufrecht zu erhalten, haben wir den Fehler begangen Unstimmigkeiten herunterzuspielen und an das Team zu glauben. Jetzt können wir sagen, dass es von sehr großer Wichtigkeit ist, die kleinsten Unstimmigkeiten anzusprechen und zu versuchen diese sofort zu lösen. Jedoch sollte dabei beachtet werden, dass das Gleichgewicht der Zugeständnisse ausgewogen bleibt.

Rückblickend können wir sagen, dass wir die Entscheidung, uns von unseren Teammitgliedern zu trennen, sehr viel eher hätten treffen müssen, um uns unsere Arbeit am Projekt zu erleichtern. So haben wir viel Arbeit doppelt gemacht, weil wir auf die Mitarbeit unserer Teammitglieder hofften.

In unserer „kleinen“ Gruppe haben wir gesehen, dass trotz Stress es sehr wichtig ist, eine gute Kommunikationsgrundlage zu schaffen. Dadurch war es möglich Schwierigkeiten anzusprechen und gemeinsam eine Lösung zu finden. Hierbei bleibt Kritik am Gruppenmitglied nicht aus, jedoch sollte dies immer ehrlich und im besten Falle konstruktiv sein.

Ein wichtiger Punkt in unserer Gruppenarbeit ist die Zuverlässigkeit. Getroffene Vereinbarungen wurden eingehalten und evtl. Abweichungen rechtzeitig bekannt gemacht, so dass die Möglichkeit bestand rechtzeitig auf die Veränderungen zu reagieren. Ebenfalls wichtig ist die Kommunikation und Koordination. Dies gelang uns sehr gut bei der Zubereitung unserer Mahlzeiten. Hier hatten wir die Möglichkeit, eine kleine Auszeit zu nehmen, aber auch weitere Vorgehensweisen zu besprechen.

Durch die Schwierigkeiten bei unsere Gruppenarbeit haben wir viele zusätzliche Erfahrungen sammeln können, die uns in späteren Teams nützlich sein werden, z.B. was es bedeutet, termingerechte Abgaben zu leisten.

Erfahrungsbericht eingesetzte Technologien

PostgreSQL

PostgreSQL ist eine professionelle relationelle Open-Source-Datenbank, die in den meisten Linux-Distributionen bereits in der Installation ausgewählt werden kann. Auch die Installation von Hand ist denkbar einfach: entpacken, starten, Datenbank einrichten – los geht's. Auch die Konfiguration für den Zugriff über ein Netzwerk ist nicht schwierig. Im Gegensatz zu beispielweise MySQL fehlen zwar die graphischen Benutzeroberflächen, die Bedienung per Konsole ist jedoch komfortabel gestaltet und gut dokumentiert. Die Datenbank ist ein robustes und mächtiges Werkzeug für die Verwaltung großer Mengen an Daten. Die gebotenen Funktionen werden zwar durch ein Projekt wie unseres gerade mal angetestet – zumal Castor nur ein Bruchteil benutzen kann – aber die einfache und zuverlässige Handhabung und der stabile Betrieb überzeugen trotzdem.

Die graphischen Frontends, die es kostenfrei für PostgreSQL im Internet gibt, sind leider entweder nicht annähernd mächtig genug, um die Möglichkeiten der Datenbank dem Benutzer darzubieten oder erfordern häufig höheren Aufwand an Einarbeitung als es die Bedienung per Konsole erfordert.

Castor

Castor ist ein Open-Source-Framework, das Java-Objekte auf relationelle Strukturen und XML-Dokumente mappen kann. Die Idee ist ein Traum für jeden Programmierer, doch obwohl grundlegende Aufgaben mit wenig Konfigurationsaufwand zuverlässig und mit hoher Performanz durchgeführt werden, kann der Versuch, komplexere objektorientierte Strukturen auf eine Datenbank zu übertragen, schnell zum Alptraum werden. Durch mangelhafte Dokumentation und überraschende Seiteneffekte ist die Verwendung von Castor für die Entwicklung von Software nicht geeignet. Da jedoch nicht einmal die Version 1.0 erreicht wurde, sollte vielleicht nicht zu früh geurteilt werden: Das Entwicklerteam und die Community sind lebhaft – eine Kombination, die nicht unterschätzt werden sollte.

VMware Workstation

VMware Workstation ist ein Programm, das virtuelle Maschinen für Gastbetriebssysteme bereitstellt. Dadurch findet ein weiteres Betriebssystem einen Rechner mit Standardhardware vor, auf dem es als einziges läuft. Die Ressourcen des tatsächlich vorhandenen PCs müssen also von beiden Betriebssystemen geteilt werden. Dadurch ist die Performance auf beiden eingeschränkt, was besonders bei der Entwicklung einer Software eine Geduldsanstrengung bedeuten kann. Trotzdem ist unsere Erfahrung mit diesem überaus souveränen Tool in der Version 3.1 durchgehend gut, die Version 4, die wir außerdem ausprobiert haben, war leider noch nicht zu empfehlen, da trotz neu hinzugekommener Funktionen das Vertrauen in die Souveränität des Programms durch merkwürdige Zwischenfälle im Wirtbetriebssystem gestört wurde.

VMware Workstation ist ein tolles Programm für jeden, der mehrere Betriebssysteme zur Verfügung haben möchte, ohne Risiken einzugehen. Um ein Betriebssystem auszuprobieren, bevor man sich dafür entscheidet, es tatsächlich auf der Festplatte zu installieren oder um einen Server im Hintergrund auf einem anderen Betriebssystem laufen zu lassen und damit die Netzwerkkommunikation zu testen, ist es ideal. Bedient man sich dem zweiten Betriebssystem jedoch regelmäßig, um damit zu arbeiten, sollte man in Erwägung ziehen, eine eigene Partition dafür einzurichten.

Eclipse

Wir haben uns für die Entwicklungsumgebung Eclipse entschieden, da wir von unseren Kommilitonen nur gute Dinge erfahren haben.

Wir haben die neuste Version von Eclipse verwendet und sind auf kleinere Bugs gestoßen, die die Arbeit manchmal ein wenig behinderten. Bspw. unterscheidet sich der Menüaufbau bei den Versionen für Windows von der unter Linux. So hat es einige Zeit in Anspruch genommen, bevor wir unter Windows externe JAR`s einbinden konnten, da wir den Menüpunkt, mit dem dies tadellos unter Linux funktionierte, nicht finden konnten. Bis wir durch Zufall das gesamte Projekt ausgewählt hatten und unter File → Preferences die gewünschte Einbindung vornehmen konnten. (Wir waren bereits soweit, die einzelnen .class – Dateien der JAR`s zu decompilieren.) Dieser Menüpunkt ist aber nur anwählbar, wenn im Ansichtsbaum das gesamte Projekt-Paket ausgewählt ist.

Eclipse bietet viele Möglichkeiten, das Programmieren zu erleichtern, bspw. Codeergänzung und Generierung von Gettern und Settern o.ä.

Diese Eigenschaften haben uns bei unserer Arbeit sehr unterstützt.

Die Entwicklungsumgebung ist sehr flexibel gestaltbar, dadurch ist es möglich die Entwicklungsumgebung individuell einzustellen.

Mit Eclipse ist es ebenfalls möglich, ein CVS-System einzubinden, so dass die Arbeit im Team erleichtert wird.

Durch die bereitgestellte Debug-Funktionalität, wurde unsere Arbeit an unserem Projekt erheblich erleichtert, dadurch fiel es leichter Fehler herauszufinden. Obwohl gesagt werden muss, dass die Debug-Funktionalität durchaus verbesserungswürdig ist, v.a. wenn man diese mit Debuggern aus C++ vergleicht, die im Umgang komfortabler und manchmal auch aufschlussreicher sind.

Abschließend bleibt zu sagen, dass die Arbeit mit Eclipse gut verlief und nach einer gewissen Einarbeitungszeit auch erleichtert wurde.

Suse Linux 8.0

Unser Server läuft auf Suse Linux 8.0. Wir haben uns für dieses Betriebssystem entschieden, weil PostgreSQL am zuverlässigsten auf einem Linux-System läuft.

Unsere Teammitglieder, die ursprünglich die Kommunikation zwischen Server und Client realisieren wollten, haben die Entscheidung für Suse 8.0 gefällt, da einer von beiden bereits einen Server mit diesem Betriebssystem zu Hause hat.

Zwischen Suse 7.x und 8.0 wurde eine umfassende Umstrukturierung vorgenommen, um die längst geplante Umsetzung einiger Standards erreichen. Zusätzlich sind viele Neuerungen vorgenommen worden, die das Betriebssystem auch für Linux-Neulinge zu einer interessanten Alternative zu Windows für ihren PC machen sollte. Leider ist diese Version auf den Markt gekommen, bevor sie umfassend getestet und bereinigt worden ist. Mögen die Bugs dem Desktop-Anwender nicht so auffallen, für den Entwickler bringen sie allzu oft unliebsame Überraschungen: Dienste starten nicht so, wie sie sollten oder werden beim Übergang in ein anderes Runlevel nicht sauber beendet. Auch die Ressourcen werden leider nicht so sehr geschont, wie man es von anderen Linux-Distributionen kennt

Trotzdem ist das neue Suse überzeugend komfortabel und einfach zu bedienen.

Zudem wurden die meisten Stolpersteine durch die inzwischen nachgefolgten Versionen beseitigt.

Swing

Swing ist eine mächtige Bibliothek, mit der eine Oberfläche gut gestaltet und verwaltet werden kann.

Allerdings ist es immer wieder verwunderlich, wenn kleinere Dinge aus keinem erkennbaren Grund nicht funktionieren. Jedoch kann meistens ein Ausweg gefunden werden.

Mit Swing lassen sich schnell Oberflächen zusammenstellen. Wenn die Oberfläche jedoch Dinge beinhaltet, die etwas komplizierter sind, gibt es im Internet viele Seiten, auf denen man sich austauschen kann und mit ziemlicher Sicherheit ein ähnliches Problem mit der entsprechenden Lösung finden kann.

Durch die vielen Möglichkeiten, die Swing bietet und der guten Dokumentation ist die Arbeit mit Swing zu empfehlen.

Helliker

Helliker ist eine open source Java ID3 tag – Bibliothek, mit deren Hilfe ID3-Tags ausgelesen und geschrieben werden können. Die Bibliothek ist in Java geschrieben und ist im Netz frei verfügbar (<http://www.id3.org>).

Die Anwendung ist unkompliziert und erste Erfolge sind schnell sichtbar.

Helliker führt eine Bugliste. Der Vorteil dieser Liste, es treten keine Bugs auf, die dort nicht aufgelistet sind. Die Schwächen sind also bekannt, aber man kann sich auf diese einstellen.

In News Groups gibt es die Möglichkeit, sich über Probleme auszutauschen und evtl. auch eine Lösung zu finden.

Alles in Allem hat Helliker unseren Ansprüchen vollkommen genüge getan und unser Projekt um nette Funktionalitäten erweitert, die die Bedienerfreundlichkeit verbessern.