

## **Jubiläumsfondsprojekt Virtuelle Teams**

### **4. Fallstudie - Open Source Software Entwicklung**

anhand des Beispiels von

**PURE DATA**, Programm zur audiovisuellen Generierung und Analyse in Echtzeit

Mag. Andrea Mayr  
Februar 2003

## Inhalt

4.1 Allgemeines zu Open Source Organisation	3
4.1.1. Open Source Software	
4.1.2. Open Source Lizenzierung	
4.1.3. Grundsätze der offen Kollaboration	
4.2. Jedes Projekt ist ein Einzelfall	6
4.2.1. Entwicklung von PD seit 1986	
4.2.2. Motivation PD zu starten	
4.2.3. Der Name PD	
4.3. Auswahl, Datensammlung und Methode	10
4.3.1. Projektauswahl	
4.3.2. Erhebungsmethoden	
4.4. Kollaboration Konkret	13
4.4.1. Voraussetzungen (Inputfaktoren)	
Aufgabentyp - Organisationsstruktur	
PD'lers als Team	
Die geographische Streuung	
PD'lers individuell	
4.4.2. Prozessfaktoren	
Projektarchitektur	
Teamführung	
Aufgabenkoordination - Zielorientierung	
Motivation - Verantwortung	
Kommunikation	
4.5 Herausforderungen	23

## 4.1.) Allgemeines zu Open Source Organisation

### 4.1.1.) Open Source Software<sup>1</sup>

Als 'Open Source' wird alle Software bezeichnet, deren Quellcode - die Programmiersprache, in der die Software geschrieben ist - frei zugänglich, kopier- und modifizierbar ist. Der Zugang zum Quellcode ist Voraussetzung, um an der Software Veränderungen vornehmen zu können, sei es um Fehler zu beheben oder die Funktionalität zu verändern. Bei sogenannter proprietärer Software wird der Quellcode als 'Firmengeheimnis' streng unter Verschluss gehalten und ist nur einer kleinen Gruppe von ausgewählten Programmieren zugänglich. Das Kopieren und/oder Modifizieren des Programms ist Dritten nicht erlaubt.

Open Source Software (OSS) gibt es seit Mitte der 1980er Jahre und seit Mitte der 1990er Jahre stellt sie in stetig wachsenden Bereichen eine ernst zunehmende Alternative zu proprietärer Software dar. Besonders das Internet würde ohne OSS in seiner heutigen Form nicht bestehen. Die Mehrheit der Webserver, beispielsweise, beruht auf OSS (Apache) und das Protokoll, das Emails verschickt und empfängt (Sendmail), ist ebenfalls OSS. In den letzten Jahren hat das OSS Betriebssystem Linux viel Aufmerksamkeit erregt als eine technisch mindestens gleichwertige Alternative für Server und seit einiger Zeit auch als Betriebssystem für normale PCs.

Der Erfolg der OSS beruht auf zwei Organisationsprinzipien, die sie radikal von konventioneller, firmenintern entwickelter Software unterscheiden: die Lizenzierung und die offene Kollaboration.

### 4.1.2.) Open Source Lizenzierung:

OSS untersteht dem Urheberrecht. Die Lizenz ist aber so formuliert, dass sie die Rechte der BenutzerInnen erweitert und nicht wie üblich einschränkt. Die wohl am weitesten verbreitete Open Source Lizenz ist die General Public License (GPL), die Mitte der 1980er Jahre von Richard Stallman verfasst wurde.<sup>2</sup> Zentral ist die Festschreibung der folgenden 3 Freiheiten:

1. Die Freiheit, das Programm für jeden Zweck auszuführen.
2. Die Freiheit, den Quellcode des Programms wörtlich zu kopieren und zu verbreiten, sofern der Copyright-Vermerk und der Quellcode mitkopiert und mitverbreitet werden.
3. Die Freiheit, das Programm zu verändern und diese veränderte Version zu kopieren und zu verbreiten, sofern das abgeleitete Werk Angaben über die Änderung enthält und gebührenfrei und unter den selben Lizenzbedingungen veröffentlicht wird.

---

<sup>1</sup> Die Begriffe 'Freie Software' und 'Open Source Software' werden hier synonym verwendet. Der erste besteht seit Mitte der 1980s Jahre, der letztere wurde 1998 geprägt. Die Unterschiede zwischen den beiden Konzepten sind vor allem weltanschaulicher Art und können in diesem Zusammenhang vernachlässigt werden.

<sup>2</sup> <http://www.gnu.org/copyleft/gpl.html>

Die GPL ist die älteste OSS Lizenz, aber nicht die einzige. Allen Lizenzen gemein sind die Punkte 1, 2 und der erste Teil von Punkt 3, der Modifikationen und deren Verbreitung erlaubt. Der zweite Teil von Punkt 3, der verlangt, dass auch abgeleitete Werke unter die gleiche Lizenz fallen, ist nicht immer vorhanden.<sup>3</sup>

Die offene Lizenzierung ist wichtig, denn sie gewährt Rechtssicherheit, nicht nur denen, die OSS benutzen, sondern vor allem denen, die OSS schreiben.

Die Lizenz schreibt rechtsverbindlich fest, dass der Code eines OSS Programms nicht plötzlich wieder dem freien Zugriff entzogen werden kann. Damit wird es möglich, auf bereits von Anderen entwickeltem Code aufzubauen, was natürlich viel effizienter ist, als jeweils von Grund auf neu anzufangen. Dies erklärt zum Teil warum sich OSS in den letzten Jahren so rasend schnell entwickelt hat und zu einer echten Konkurrenz kommerzieller Software geworden ist.

Zum anderen Teil lässt sich die erstaunliche Entwicklungsdynamik aus den Grundsätzen der offenen Kollaboration erklären.

Während die GPL mit dem Namen Richard Stallman verbunden ist, sind die Prinzipien der offenen Kollaboration mit dem Namen Linus Torvalds (Linux) verbunden.

#### **4.1.3.) Grundsätze der offenen Kollaboration:**

1. Selbstmotivation: OSS Projekte nehmen in der Regel ihren Anfang wenn ein/e ProgrammiererIn auf ein Problem stösst, zu dem es noch keine passende Softwarelösung gibt. Nun gibt es zwei Möglichkeiten. Falls sich das Problem lösen lässt, in dem man ein bestehendes OSS Programm ändert, dann ist die effizienteste Strategie, die neue Funktionalität so zu schreiben, dass sie dem bestehenden Programm beigelegt werden kann. Da oftmals dasselbe Problem von mehreren Personen gleichzeitig angegangen wird, macht es Sinn, mit diesen zusammenzuarbeiten und nicht alles selbst zu entwickeln.

Falls das Problem nicht innerhalb eines bestehenden Programms angegangen werden kann, muss ein ganz neues Projekt gestartet werden. Oftmals können aber auch dazu bereits beträchtliche Codesegmente aus anderen Programmen benutzt werden. Falls sich andere EntwicklerInnen finden lassen, die das gleiche Problem als dringlich empfinden und den vorgeschlagenen Lösungsansatz als sinnvoll erachten, dann kann die Zusammenarbeit beginnen.

Wesentlich für die Selbstmotivation ist, dass das Projekt lösungs- und nicht verkaufsorientiert ist. Dies heisst nicht, dass es keinerlei kommerzielle Aspekte innerhalb der OSS gibt, aber das Programm selbst ist nicht kommerziell, denn der Quellcode ist ja frei vorhanden. Dies wiederum ist Voraussetzung, dass eine möglicherweise über alle Erdteile verstreute Entwicklergemeinschaft zusammenfinden kann. Die Vorteile einer solch grossen Entwicklergemeinschaft sind so bedeutend,

---

<sup>3</sup> Für einen Überblick über die verschiedenen OSS Lizenzen, siehe <http://www.opensource.org>

dass es sich auch für Unternehmen lohnen kann, freie Software zu schreiben (und dann Services und individuelle Modifikationen auf kommerzieller Basis anzubieten).

2. Offenheit des Inputs. Alle EntwicklerInnen/BenutzerInnen haben die Möglichkeit, die weitere Programmentwicklung zu beeinflussen. Dies kann von der Einreichung einer einfachen Fehlermeldung bis hin zur Übernahme der Verantwortung für den Unterhalt eines Programmsegments gehen. Der offene Input hat den Vorteil, dass viel schneller Fehler gefunden und behoben werden, denn alle BenutzerInnen sind (potentielle) TesterInnen und MitentwicklerInnen. Denjenigen, die diese Rolle auch aktiv ausüben wollen, wird ein Forum geboten, innerhalb dessen ihre Expertise maximal ausgewertet werden kann. Je breiter ein Programm getestet werden kann, desto stabiler wird es. Da die Wirklichkeit immer komplexer ist als ihre Simulation, ist die Chance, einen Fehler im Code zu finden, viel grösser, wenn er in realen Anwendersituationen getestet werden kann, als wenn die Testphase auf ein firmeninternes Labor beschränkt wird.
3. Versionierung. Software ist nie 'fertig' sondern wird immer weiterentwickelt. Software, die nicht mehr weiterentwickelt wird, 'zerfällt', denn ihre Fehleranfälligkeit steigt, weil sie nicht mehr auf das sich verändernde Umfeld, innerhalb dessen sie angewandt wird, angepasst wird. Softwareentwicklung ist ein kontinuierlicher Prozess, der eigentlich nur aus anwendungsorientierten Gründen in diskrete Schritte (Versionen) unterteilt wird. Jede auch noch so gute Programmversion kann noch verbessert werden, denn a) sie ist nie fehlerfrei und b) enthält nie alle gewünschte Funktionalität. Proprietäre Software wird aus kommerziellen Gründen nur in grösseren Abständen veröffentlicht. Das heisst, dass alle Fehler bis zur nächsten Version unbehoben bleiben. OSS dagegen wird viel öfter veröffentlicht, das heisst, dass Fehler schneller behoben werden können und neue Funktionalität schneller verfügbar wird.

Um diesen Prozess für verschiedene EntwicklerInnen/BenutzerInnen praktikabel zu machen, gibt es OSS oftmals in verschiedenen 'Qualitätsstufen'. Die Linux-Distribution Debian, beispielsweise, gibt es als 'stable', 'unstable' und 'testing' Version. Die 'stable' Version ist ausführlich getestet von höchster Qualität. An ihr werden keine Veränderungen mehr vorgenommen. Die 'unstable' Version enthält den neuesten Code, ist aber noch nicht ausführlich getestet worden und zielt deshalb auf SoftwareentwicklerInnen ab, die, damit sie neueste Funktionalität benutzen können, bereit sind, allfällige Fehler zu beheben. Die mittlere, die 'testing' Version, ist schon so weit stabilisiert, dass ihre Funktionalität nicht mehr verändert wird, aber noch Korrekturen vorgenommen werden.

Diese Strategie der Veröffentlichung von Code erlaubt es, ihn gleichzeitig kontinuierlich weiter zu entwickeln und eine stabile Version für die reinen AnwenderInnen zur Verfügung zu stellen.

4. Flache Hierarchien. Der OSS Entwicklungsprozess steht allen offen, aber er ist gleichzeitig so organisiert, dass die guten Beiträge herausgefiltert werden können. Dazu Bedarf es sogenannter 'maintainer' deren Rolle es ist, zu entscheiden, welcher

neue Code, der von der verteilten Entwicklergemeinde zugesandt wurde, in die nächste Version des Programms aufgenommen wird. Meistens sind dies ehrenamtliche Tätigkeiten, die von denjenigen ausgeübt werden, die innerhalb der 'community' das grösste Ansehen und Vertrauen geniessen. Die freie Verfügbarkeit des Quellcodes garantiert nun, dass die Maintainer ihre Macht nicht missbrauchen können. Denn findet eine genügend grosse Zahl relevanter EntwicklerInnen, dass sich ein Projekt nicht in die richtige Richtung entwickelt, könne sie einfach einen neuen Maintainer bestimmen und das Projekt spaltet sich. Diese Organisationsform wurde auch schon 'sanfte Diktatur' genannt, denn der Maintainer besitzt im Prinzip absolute Macht zu entscheiden, welcher Code aufgenommen wird, diese Macht ist aber instabil, insofern er/sie sie nur ausüben kann, so lange andere dieser Person freiwillig zuliefern.

Die Lizenzierung schafft den rechtlichen Rahmen, innerhalb dessen OSS entwickelt werden kann, während die offene Kollaboration die Praxis organisiert, um das Potential, das in der Lizenz geschaffen wird, auch ausreizen zu können.

Die Bandbreite der bestehenden OSS Projekte und Entwicklungskontexte ist riesig. Sie reicht von kommerziellen Firmen, die OSS entwickeln, um einen Service Markt zu schaffen (IBM, Red Hat, Suse) zu Universitäten, die OSS als Teil ihrer akademischen Forschung schreiben und wie alle akademische Forschung frei publizieren; von professionellen ProgrammiererInnen, die an OSS Projekten während ihrer Arbeitszeit teilnehmen (die überwältigende Mehrheit der InformatikerInnen arbeitet heute in Betrieben die Software benutzen, aber keine verkaufen, auch sie sind lösungs- und nicht verkaufsorientiert) zu EntwicklerInnen, die in ihrer Freizeit, aus einer Vielzahl von Gründen, an solchen Projekten mitarbeiten. Es kann vorkommen, dass in einem einzigen Projekt alle Kontexte gleichzeitig vertreten sind. Diese ungewöhnliche Mischung aus Produktionskontexten hat sich in den letzten 10 Jahren als ausserordentlich produktiv und auch stabil erwiesen.

## **4.2.) Jedes Projekt ist ein Einzelfall**

Wie bereits erwähnt, ist die OSS ein sehr breites Feld innerhalb dessen es erhebliche Unterschiede gibt, je nach Entwicklungs- und Anwendungskontext. Noch weniger als bei proprietärer Software gibt es das 'typische' Projekt, sondern um die Entwicklungsdynamik eines spezifischen Projektes zu verstehen, muss der Kontext mitberücksichtigt werden.

Bei der nachfolgenden Fallstudie handelt es sich um eine Untersuchung der gegenwärtigen Entwicklung des Programmes Pure Data (PD). Dieses Programm wird in erster Linie von MusikerInnen für die eigene Arbeit entwickelt und benutzt. Die Lösungsorientierung ist hier besonders stark, während das kommerzielle Interesse bei den ProgrammiererInnen im Hintergrund steht. PD ist eine graphische Programmierumgebung zur audiovisuellen Generierung und Analyse in Echtzeit ('real-time graphical programming environment for audio and graphics analysis and rendering'). Es läuft plattformübergreifend unter den Betriebssystemen: Windows,

Linux, Irix oder Macintosh OSX.

#### **4.2.1.) Entwicklung von PD (Ursprung, Verzweigungen, Start)**

Der Kern des Programmes PD wurde 1996 von Miller Puckette mehr oder weniger im Alleingang geschrieben.

Zehn Jahre zuvor, 1986, hatte Miller Puckette im Auftrage des IRCAM<sup>4</sup> ein Programm, den sog. Patcher geschrieben. Dieser bildete nicht nur den Ursprung für PD, sondern für eine ganze Programmfamilie, die seither entstanden ist und in verschiedenen Konfigurationen noch immer aktiv weiterentwickelt wird.

Die Rahmenbedingungen in Bezug auf Software-Entwicklung und -Anwendung haben sich in der Zeitspanne zwischen 86 und 96 völlig verändert. Hardware wurde nicht nur leistungsfähiger, sondern auch für den Privat-Gebrauch erschwinglich. Grafische Benutzeroberflächen, wie wir sie heute kennen, wurden entwickelt. Microsoft Windows und Apple Macintosh, als auch das auf offenen Sourcen basierende Betriebssystem Linux, fanden Verbreitung.

Vor diesem Hintergrund veränderten sich natürlich die Strategien von Software Firmen. Hatten sie ursprünglich mit Hardware Herstellern gemeinsam an hoch spezifischen Lösungen gearbeitet, versuchten sie nunmehr ihre Produkte so zu gestalten, dass sie keine speziellen Hardware Architekturen mehr verlangten, sondern auch auf billigen Allzweck-Prozessoren funktionierten. In vielen Fällen war das nicht einfach in einer Änderung eines Teiles des Programmes zu erreichen, sondern erforderte eine grundlegende Änderung in der Programm Architektur.

Das Patcher Programm von Miller Puckette wurde Ende der 80er Jahre vom IRCAM an die Firma Opcode lizenziert, wo 1990 eine erste umfassende Überarbeitung des Programmes durch David Zicarelli durchgeführt wurde. Das Programm wurde bekannt unter dem Namen MAX/Opcode und läuft in aktueller Version als MAX/MSP auf Apple/Macintosh Computern.

Am IRCAM wurde parallel dazu auch weiterentwickelt. Seit 1989 arbeitete Miller Puckette im "Ircam Musical Workstation Team" und entwickelte eine erweiterte Version von MAX, namens MAX/FTS<sup>5</sup>, die im Rahmen der IRCAM Signal Processing Workstation (ISPW) wiederum kommerziell vermarktet wurde. Nach seinem Ausscheiden führte das "IRCAM real time systems" Team, von Francois Dechelle 1995 gegründet, dieses Projekt weiter. Aus diesem Strang geht das Programm jMax hervor, das mittlerweile auch mit offenem Sourcecode unter der GPL verfügbar ist.

Miller Puckette ging 1994 an die University of San Diego, um dort zu unterrichten. Er beschloss ein Programm zu schreiben, das sich generell an MAX orientierte, jedoch

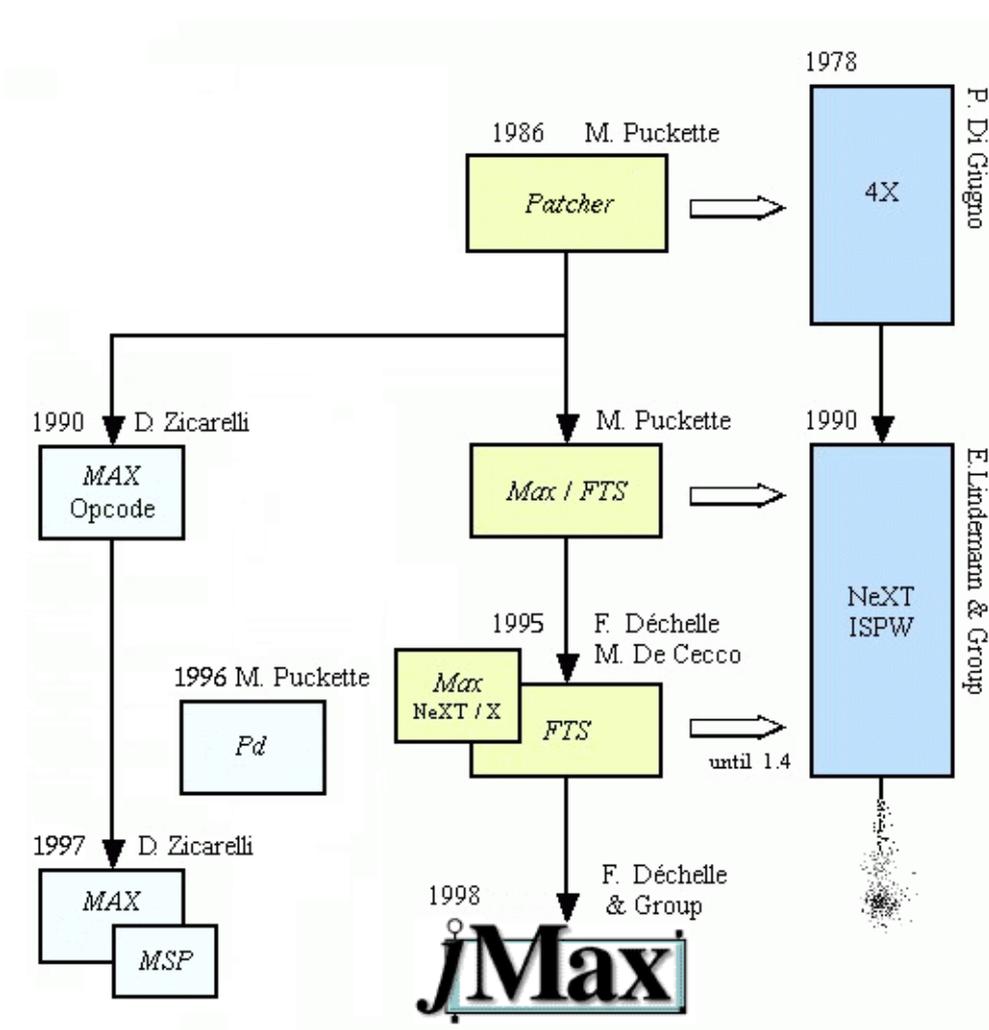
---

<sup>4</sup> Institut de Recherche et Coordination Acoustique/Musique <http://www.ircam.fr>

<sup>5</sup> <http://crca.ucsd.edu/~msp/bio.htm>

Schwächen, die er mittlerweile in MAX sah, ausräumen und Stärken beibehalten sollte. Als er 1996 in einem seiner ersten Vorträge über PD die Programmarchitektur erläuterte merkte er an, dass seiner Einschätzung nach, der Erfolg und das Scheitern des Programmes genauso vom Finden einer guten Community abhängig sei, wie von den Design Spezifikationen des Programmes selbst. Diese Erkenntnis ist ganz typisch für OS Projekte.

Die nachfolgende Graphik zeigt die Verzweigungen der Programmfamilie die aus dem ursprünglichen Patcher hervorgegangen ist.



Weiterentwicklung des ursprünglichen Patcher Programmes in verschiedene Versionen, vom IRCAM aus gesehen.<sup>6</sup>

<sup>6</sup> Quelle: <http://www.ircam.fr/equipes/temps-reel/jmax/en/history.html>

#### 4.2.2.) Die Motivation PD zu starten

Zehn Jahre lang hat Miller Puckette sowohl alleine als auch im Team, an Max - der Software die er ursprünglich kreierte - gearbeitet, bevor er sich dazu entschloss nochmal von Vorne anzufangen.

Die Begründung zum Redesign des Programmes ist in technischen Unterlagen jeweils mit der Erwähnung der Verbesserung in Bezug auf Schwächen im MAX Programm schnell abgetan. Das ist ein klares Argument, aber sicher nicht der einzige Grund, der dazu geführt hat, PD zu starten. Betrachtet man das, was in einzelnen seiner Arbeitsschwerpunkte als essentiell hervorgehoben wird, mit seiner beruflichen Situation in dieser Zeit - dem Wechsel vom Ircam zur University of California, San Diego - zeichnet sich ein sehr viel breiteres Spektrum an Argumenten ab, die diesen Schritt bewegten.

- Einfaches Reproduzieren von elektronischen Musikstücken unabhängig von proprietärer Technologien<sup>7</sup>:  
Miller Puckette ist neben seiner Profession als Programmierer auch Sound Künstler. Er setzt seine Software aktiv in Performances ein, überschreitet somit die Grenze zwischen Konstruktion und Anwendung. Neben seiner eigenen Aktivität als Musiker hat er auch über Jahre hindurch andere Künstler begleitet, die aufwendige Kreationen bzw. Kompositionen in Software eingeschrieben haben, die sehr kurze Zeit später nicht mehr oder nur mit überdimensionalem Aufwand abrufbar waren.
- Dezentralisieren von Computer Musik<sup>8</sup>:  
Programme wie MAX (oder nunmehr PD) stellen - vereinfacht gesprochen - elektronische Musikinstrumente dar. Gekoppelt an teure Lizenzkosten, den ursprünglich noch extrem kostenintensiven Hardwarevoraussetzungen und einer limitierten Gruppe von Personen, die den Sourcecode einsehen und somit programmieren konnten, schränken den Anwenderkreis auf einen sehr kleinen, institutionell oder kommerziell unterstützten, ein.
- Emanzipation als Programmierer  
Restriktionen, die proprietäre Software Entwicklung im Allgemeinen kennzeichnen, kamen auch zum Tragen, als Miller Puckette seinen Job wechselte und seinen Lebensmittelpunkt von Europa nach USA verlegte. Er durfte sein Arbeitswerkzeug als Programmierer nicht mehr weiterentwickeln.<sup>9</sup>

Gründe wie diese sind sicher einflussreich gewesen in der Entscheidung PD als Projekt neu zu starten. Sowohl die offenen Sourcen, als auch die freie Lizenz sollten einen möglichst breiten Zugang für BenutzerInnen und EntwicklerInnen sichern.

---

7 New Public-Domain Realizations of Standard Pieces for Instruments and Live Electronics, MP, 2001

8 Ibid.

9 Winfried Ritsch, Interview

### 4.2.3.) Der Name PD

Der Name des Programmes steht als Akronym für 'Pure Data', das - für ProgrammiererInnen offensichtlich - bezeichnend ist für den objektorientierten Ansatz des Programmes ("..which means that a 'pure datum' is handled as a data object within pd"<sup>10</sup>), wird aber auch -gerade durch den Entstehungszusammenhang des Programmes - als eine Referenz hin zum Begriff 'public domain' verstanden<sup>11</sup>. Klar zu unterscheiden ist in diesem Zusammenhang die Verwendung des Begriffes 'public domain' in assoziativer Art, bezogen also auf die grösstmöglichen Freiheiten bei der Anwendung des Programmes, im Gegensatz zum lizenzrechtlichen Status 'public domain', der durch den Wegfall von urheberrechtlichen Ansprüchen kommt (beispielsweise durch Verjährung oder durch fehlende Kennzeichnung), was bei PD jedoch nicht der Fall ist. Heute wird PD von einer weltweit verteilten Gruppe von ProgrammiererInnen/MusikerInnen entwickelt. Die nachfolgende empirische Untersuchung befasst sich mit ihrer Zusammensetzung, Struktur und Dynamik.

## 4.3.) Auswahl, Datensammlung und Methode

### 4.3.1.) Projektauswahl

PD wurde als Fallstudie ausgewählt, weil es sich dabei um ein sehr aktives, internationales, internet-koordiniertes Projekt handelt, welches aber auch einen starke lokale Komponente besitzt, denn einige zentrale Mitglieder sind in Österreich ansässig.

Dies erlaubte eine umfassende Beobachtung des Projektes durch die Verwendung unterschiedlicher Erhebungsmethoden. Von besonderer Bedeutung hierbei waren:

- Zugang der Autorin zu online Kanälen, die zum Informationsaustausch zwischen den EntwicklerInnen der internationalen Community genutzt werden.
- die Erreichbarkeit von aktiven Community Members in Face-to-Face Situationen, zur Durchführung von ExpertInnen-Interviews
- der Kontakt zu einer oder mehreren Schlüsselpersonen, deren Stellenwert in der Community als Reputationsfaktor zugunsten der Umfrage geltend gemacht werden kann (persönlicher Aspekt)

Die Einschätzung des Projektes als sehr aktiv beruht auf folgenden Kriterien:

- Rege Kommunikation zwischen den Community Members.
- Wachstum dieser Community
- Etabliertheit des Produktes in seinen spezifischen Benutzerkreisen.

---

<sup>10</sup> Vgl. Günther Geiger, <http://www.iem.at/maillinglists/pd-list/2000-07/0036.html>

<sup>11</sup> Vgl.: FOLDOC online Dictionary of Computing:  
<http://foldoc.doc.ic.ac.uk/foldoc/foldoc.cgi?query=PD>

Während PD sicherlich nicht das bekannteste, oder grösste OSS Projekt ist, war es die obengenannte Kombination von Gründen und auch die relativ überschaubare Grösse, die den entscheidenden Ausschlag zur Auswahl gaben.

Zugunsten der Verfügbarkeit von lokalen ProtagonistInnen wurde der Auswahlprozess des Untersuchungsobjektes ursprünglich nicht über eine ausschliessliche Online Recherche gestartet. Zielführend schien vielmehr, lokal die entsprechenden Zielgruppen anzusteuern, die potentielle EntwicklerInnen von Software beherbergen, wie z.B lokale Linux Benutzer Gruppen (LUG's), oder akademische Kreise deren Schwerpunkte informatisches Arbeiten miteinschliessen.

#### **4.3.2.) Erhebungsmethoden**

Auf die Möglichkeit die zu untersuchende Gruppe, sowohl in ihren Kommunikationskanälen online agieren zu sehen, als auch individuell vertreten in Face-to-Face (F2F) Situationen erreichen zu können, wurde, wie eben erwähnt, schon bei der Auswahl des Projekts Wert gelegt. Eine Strukturierung der Erhebungsmethoden liesse sich demzufolge nach diesen Kriterien anlegen, wobei dies jedoch ein Zugang wäre, der in erster Linie eine ForscherInnen Perspektive in Bezug auf die Anwendung technischer Hilfsmittel beschreibt, anstatt -was in diesem Fall von Vorteil wäre- Erhebungskonstellationen als Verhaltensstrategien zu beschreiben. Hierbei kann eine Ausdifferenzierung unabhängig von den verwendeten Hilfsmitteln geschehen und bezieht sich auf die Sichtbarkeit oder Verdecktheit der Forscherin. Die Strukturierung wurde in die Gegenüberstellung von Aktivitäten, die ankündigend/teilnehmend versus zurückhaltend/beobachtend ausgeführt wurden, aufgeteilt.

- **ankündigend/teilnehmend**

- qualitativer Zugang in teilstrukturierten Gesprächen mit Experten<sup>12</sup> (F2F)  
Einzelne Programmierer aus der Arbeitsgruppe, die in Österreich lokalisiert sind, wurden dazu eingeladen.
- quantitativer Zugang durch eine webbasierende Umfrage in Form eines Fragebogens (online)  
In Kommunikationskanälen die vorwiegend zur Weiterentwicklung des Programmes genutzt werden, wurde auf die quantitative Umfrage hingewiesen. Eine Mailingliste, die das primäre Medium zu diesem Zweck darstellt, ist auch in einer Website archiviert abzurufen. Um einer Teilnahme von Personen entgegenzuwirken, die nicht Teil der EntwicklerInnengruppe ist, wurde dieser Fragebogen mit Usernamen und Passwort geschützt.

---

<sup>12</sup> Die Experten waren ausschliesslich maennlich, deshalb wurde hier geschlechtsspezifisch formuliert.

- **zurückhaltend/beobachtend**

- Abgrenzen des Aktivitäts-Feldes der PD-Community (online)  
Hier wurde der Frage nachgegangen welches Kommunikationsmittel sich die Gruppe wie intensiv bedient.
- qualitative Auswertung von Gesprächssequenzen (online)  
Die öffentliche Archivierung der gesamten Korrespondenz in Bezug auf die Weiter-Entwicklung des Programmes, ermöglicht es punktgenau anzusetzen und Gesprächssequenzen die zu einer Entscheidungsfindung führen, nachzulesen. Die qualitative Analyse dieses Materials gestaltete sich in der Praxis jedoch als äusserst schwierig, da der Grossteil der Kommunikations-Abfolgen von technischen Beschreibungen geprägt ist.
- quantitative Auswertung von Gesprächssequenzen (online)  
Hierzu wurde die gesamte, öffentlich archivierte Email Korrespondenz des EntwicklerInnen Teams seit Anfang 1998 (vier einhalb Jahre) in eine Datenbank eingespeist und untersucht.
- teilnehmende Beobachtung als Mailinglisten Subskribentin (online)  
Mit der Entscheidung zur Untersuchung dieser Gruppe erfolgte - soweit als möglich - die Subskription an allen relevanten Kommunikationskanälen der Gemeinschaft. Das Folgen auf inhaltlicher Ebene gestaltete sich rückwirkend - wie oben beschrieben- als schwierig, auf regulärer täglicher Basis ist es gut möglich.
- teilnehmende Beobachtung bei Gruppentreffen (F2F)  
Die EntwicklerInnengruppe in Wien trifft sich regelmässig in Face-to-Face Situationen. Da diese Treffen über die Mailingliste angekündigt werden und üblicherweise in einem Lokal stattfinden ist es möglich unangekündigt als auch unerkannt teilzunehmen.

Die sehr unterschiedlichen Erhebungsmethoden hatten genauso vielfältige Ergebnisse zur Folge. Das Layout der Fragestellungen in qualitativer, als auch quantitativer Art hat sich jedoch von Anfang an an der Vorgabe einer Matrix aus Input-, Prozess- und Output Variablen orientiert. Fragestellungen zu diesen drei Teilbereichen wurden versucht aus individueller als auch gruppenspezifischer Perspektive zu erörtern. Der Möglichkeit der Querreferenzierung in einem Endbericht, die auf dieser Struktur aufbaut, wurde somit versucht entgegenzukommen.

## **4.4.) Kollaboration Konkret**

Dieser Abschnitt basiert auf den Ergebnissen der einzelnen Erhebungsmethoden und versucht durch deren Referenzierung untereinander eine möglichst dichte Beschreibung der Projektgruppe zu erhalten.

### **4.4.1.) Voraussetzungen (Inputfaktoren)**

#### **Aufgabentyp - Organisationsstruktur**

Der Aufgabenstellungen, die die heutige Weiterentwicklung des Projektes charakterisieren, werden meistens bestimmt durch den Bedarf an Lösungen zu Problemen, die sich aus der eigenen Arbeit der Community Mitglieder ergeben. Die Motivationsfaktoren, die zum Start des Projektes führten, wurden bereits in Kapitel 4.2.2. umrissen.

Als Organisationsstruktur für dieses Vorhaben dient ein OSS Entwicklungs-Szenario. Die Teamzusammensetzung ist - wie in den meisten OSS Projekten - abhängig von freiwillig kooperierenden ProgrammiererInnen. Die Interessen zur Partizipation sind dabei individuell geprägt.

Die Frage, wer nun diese Personen sind, die an der Entwicklung von PD teilnehmen, welche Faktoren als motivationsfördernd einzuschätzen sind und welche weiteren soziodemographischen Erkenntnisse aus den einzelnen Untersuchungen zu ziehen sind, versucht der folgende Abschnitt zu vermitteln.

#### **PD'lers als Team**

Die PD Community besteht aus einer grossen Gruppe an technikversierten Personen, die vorwiegend über internet-basierende Hilfsmittel miteinander kommunizieren. Sie tun das, um in Bezug auf die Benutzung und Weiterentwicklung des Programmes PD und dessen Erweiterungen Informationen auszutauschen.

Die Grenze zwischen EntwicklerInnen und AnwenderInnen in dieser Gruppe ist fließend. Zum Einen ist dies zurückzuführen auf die Konstellation eines OSS Entwicklungsszenarios, das seiner Art entsprechend nach Aussen hin zur Teilnahme lädt, zum Anderern durch die Möglichkeiten als AnwenderIn bereits programmiertechnisch komplexe Funktionen in das Programm miteinbinden zu können.

Es gibt eine kleine Gruppe an Programmierern, die den Kern der Community bilden, im fließenden Übergang zu einem - im Vergleich dazu - grossen Kreis aus BenutzerInnen mit durchwegs sehr hohen technischen Qualifikationen im Bereich der

digitalen und/oder analogen Signalverarbeitung. Die Meisten verwenden das Programm bereits über einen langen Zeitraum, setzen es experimentell ein und realisieren oft sehr spezifische Anwendungen. Diese BenutzerInnen sind grossteils auch diejenigen, die auf spezielle Problemstellungen bei ihren Anwendungen hinweisen, somit einen wichtigen Teil im Prozess zur Stabilisierung des Programmes beitragen.

Nach Aussen hin wird die Gruppe, nichtzuletzt durch ihr starkes Wachstum, sehr lose. Die Vielfalt an Sammelpunkten im Netz, die von einzelnen BenutzerInnen initiiert und von anderen genutzt und mitgestaltet wird, lässt die Bestimmung der Zahl von PD'lers nicht zu.

Was sich allerdings gut in Zahlen ausdrücken lässt, sind die Ergebnisse der Untersuchung einer Mailingliste, die ein zentrales Kommunikationsmedium in der Weiterentwicklung des Programmes ist: pd-list, die offizielle Mailinglist für die PD-Community. Pd-list war ursprünglich die einzige Liste, die die Community unterstützte, mittlerweile gibt es neben dieser Liste noch weitere Verteiler namens pd-dev, pd-announce und pd-off-topic.

Gestartet wurde die erste Liste im Jänner 1998 am Institut für Elektronische Musik und Akustik (IEM) der Universität für Musik und darstellende Kunst in Graz. In der Unterstützung von Miller Puckette und derer, die ihn zu diesem Zeitpunkt in der Entwicklung des Programmes bereits halfen, sollte gleichzeitig den Studierenden des Instituts ein Anschluss an die Produktions- als auch Anwendungsaktivitäten von PD geboten werden.

Anzahl der SubskribentInnen in den einzelnen Listen im Februar 2003:

<u>Listenname</u>	<u>SubskribentInnen</u>	<u>Inaktiv Status</u>
pd-list: (main)	584	(90)
pd-dev: (developers)	410	(70)
pd-ot: (off-topic)	185	(17)
<u>pd-announce</u>	<u>440</u>	<u>(54)</u>
Summe abzüglich Überschneidungen durch Subskription an mehreren Listen	716	~ 16%
Aktiv subskribierte Email Adressen (ca):	600	

Eine realistische Anzahl der Email Accounts, die PD bezogene Informationen erhalten kann aus der Summe aller Listen abzüglich der Überschneidungen durch mehrfach-Subskriptionen hergeleitet werden: 716. Diese Anzahl verschiedener Adressen ist allerdings noch um den durchschnittlichen Wert von 16%, der den Inaktiv-Status<sup>13</sup> einzelner Email-Adressen kennzeichnet, zu vermindern und beläuft

<sup>13</sup> vorübergehende Deaktivierung der Zustellung von Nachrichten bei Mailinglisten

sich demzufolge auf ca. 600. Die Zahl 600 kann somit als ein Richtwert<sup>14</sup> zur Orientierung in der Dimension des Teils der Community gesehen werden, der auf die Mailingliste als Informations- und Kommunikationsmedium zurückgreift. Recherchen im Rahmen dieser Studie zufolge, trifft das auf die Mehrheit der Community Members zu.

### **Die geographische Streuung**

Als zentrales Kommunikationswerkzeug, dessen Archiv über das World-Wide-Web einsehbar ist, bot die Mailinglist gute Voraussetzungen um einer Einschätzung der Community in unterschiedlichen Kategorien näherzukommen. Die Zahl von mehreren hundert Community Members verringert sich stark, schwenkt man den Blick von der langen Liste der eingeschriebenen grossteils passiven Subskribentinnen auf die Liste derer, die sich innerhalb der Gruppe aktiv mit Beiträgen beteiligen. Die Auseinandersetzung mit dem Mailinglisten Archiv ermöglicht die Annäherung an diesen aktiven Kern. Dort scheinen natürlich nur die Adressen derer auf, die sich aktiv beteiligen.

Das Archiv geht zurück bis Anfang 1998 und eröffnet beim ersten sporadischen Blättern bereits einen Eindruck der Dynamiken innerhalb der Arbeitsgemeinschaft PD. Generell sind einzelne Beiträge in diesem Archiv nach dem Datum ihres Eintreffens sortiert, innerhalb der zwölf Monate pro Jahr wird diese chronologische Anordnung nur durch die Zusammenfassung von Themenschwerpunkten, sog. Threads, unterbrochen.

Was in den einzelnen Nachrichten des Archives neben zweckbezogenen Inhalten an Parametern bezeichnend für die Absender der Mitteilungen ist, sind Sende-Adressen und Datumsangaben. Neben der Uhrzeit zu der eine Mitteilung an die Verteilerliste geschickt wurde, verrät uns die Datumsangabe auch die Zeitzone aus der eine Nachricht abgeschickt wurde. Diese Information hilft ein geographisches Netz zu skizzieren, dass in Abstimmung mit Häufigkeiten einzelner Länder Codes Rückschlüsse auf Dichten in einzelnen Gebieten zulässt.

Um diese Auswertung zu erhalten wurden die Daten des gesamten Archivs über mehrere Schritte hindurch prozessiert und in eine eigene Datenbank eingespielt. Die Abfrage dieser Daten zeigt uns die geographische Streuung. Die Einträge beziehen sich auf die Anzahl verschiedener Absender aus der entsprechende Zeitzone pro Halbjahr. Mehrfache Beiträge einzelner Absender wurden ignoriert.

---

<sup>14</sup> Anzumerken sei hier, dass es Faktoren gibt, die diesen Wert sowohl nach Oben als auch nach Unten hin beeinflussen. Veraltete Email Adressen, doppelte Subskription einzelner Personen unter verschiedenen Email Accounts als auch das Einschreiben weiterer Verteilerlisten innerhalb der PD Listen bringen diese Zahl ins Schwanken. Die Annahme, hinter einer Email Adresse jeweils eine Person zu finden, ist natürlich eine Vermutung und muss dementsprechend vorbehaltlich verstanden werden.

(Längengrade: 120-30 Grad West) ~ Amerika

	<b>-8h</b>	<b>-7h</b>	<b>-6h</b>	<b>-5h</b>	<b>-4h</b>	<b>-3h</b>	<b>-2h</b>
1998/1-6	3	3	1	4	2		2
1998/7-12	4	5	2	1	1		
1999/1-6	8	4	4	6	4		
1999/7-12	7	7	3	3	3		
2000/1-6	9	3	2	9	5		
2000/7-12	3	2	1	9	12		
2001/1-6	3	7	2	15	19		
2001/7-12	9	7	5	20	22	3	
2002/1-6	9	16	6	35	33		

Vertikal: die Zeitspanne in halbjährlicher Abstufung.

Horizontal: die Zeitzonen in Relation zur Greenwich Mean Time

(Längengrade: 15 Grad West- 75 Grad Ost) ~ Europa

	<b>-1h</b>	<b>+/- 0h</b>	<b>+1h</b>	<b>+2h</b>	<b>+3h</b>	<b>+4h</b>	<b>+5h</b>
1998/1-6	2	3	2	5			
1998/7-12		1	6	4			
1999/1-6		2	7	6			
1999/7-12		6	11	9			
2000/1-6		3	17	19	1		
2000/7-12		3	16	27	2		
2001/1-6		14	38	31	3		
2001/7-12		13	44	41	2		1
2002/1-6		33	73	65	1	2	

Vertikal: die Zeitspanne in halbjährlicher Abstufung.

Horizontal: die Zeitzonen in Relation zur Greenwich Mean Time

(Längengrade: 90 Grad Ost- 180 Grad Ost) ~ Japan/Australien

	<b>+6h</b>	<b>+7h</b>	<b>+8h</b>	<b>+9h</b>	<b>+10h</b>	<b>+11h</b>	<b>+12h</b>
1998/1-6				1			
1998/7-12							
1999/1-6				1		1	
1999/7-12				2	1	1	
2000/1-6				2		1	
2000/7-12			1	1		1	1
2001/1-6				4	3	2	
2001/7-12				2	5	2	
2002/1-6			2	4	6	2	

Vertikal: die Zeitspanne in halbjährlicher Abstufung.

Horizontal: die Zeitzonen in Relation zur Greenwich Mean Time

Zieht man die Endungen der Domainnamen, sog. Top Level Domains (TLD's) der Emailadressen auch heran, um Aufschluss über die Lokalisierung von Pd-Members zu bekommen, bestätigt sich die Anhäufung in den Gebieten, wie sie die Auswertung der Zeitzonen in obiger Tabelle ergaben. TLD's die keine Ländercodes darstellen können in dieser Art nicht interpretiert werden. Es handelt sich dabei um .com, edu, .Info, .net und .org Adressen.

Häufigkeiten einzelner TLD's aus der Subskribenten Liste:

<i><b>TLD</b></i>	<i><b>%</b></i>	<i><b>TLD</b></i>	<i><b>%</b></i>	<i><b>TLD</b></i>	<i><b>%</b></i>	<i><b>TLD</b></i>	<i><b>%</b></i>
.com	27.37	.ca	3.72	.fi	0.74	.ms	0.19
.edu	12.66	.ch	0.74	.fm	0.19	.nl	1.12
.Info	0.19	.cl	0.37	.fr	2.23	.pl	0.19
.net	8.57	.co	0.19	.hk	0.19	.pt	1.12
.org	7.08	.cx	0.19	.ie	0.56	.se	1.12
.at	10.06	.de	5.96	.it	2.05	.si	0.19
.au	1.86	.dk	1.12	.jp	2.05	.uk	5.03
.be	1.68	.es	0.74	.lu	0.19	.ws	0.37

Diese Auswertung der Daten zeigen klar, dass sowohl passive, als auch aktive Community Members über den gesamten Globus verteilt lokalisiert sind. Europa, Usa, Japan und Australien sind dabei die geographischen Abschnitte, die als dominierend hervortreten<sup>15</sup>.

### **PD'lers individuell**

PD ist kein Tool, das intuitiv einsetzbar ist. Es bedarf sowohl überdurchschnittlicher informationstechnologischer Kenntnisse, als auch einer Beschäftigung mit der Funktions- und Einsatzweise des Werkzeugs selbst. Diese Zugangs-Voraussetzungen wirken sich mit den Umständen des jeweiligen individuellen Entdecken des Programmes - das der Umfrage zufolge meist durch gezielte Hinweise erfolgt - stark auf die Zusammensetzung der Gruppe aus. In dieser Hinsicht wirkt die Gruppe sehr homogen.

Der Versuch PD'lers individuell zu erfassen, kann natürlich nur darauf abzielen ein Bild typischer Merkmale von Personen zu skizzieren, die dieser Gruppe angehören. Um dieser Aufgabe entgegenzukommen wurde ein Fragebogen gestaltet, der Passwort-geschützt über eine eigens angelegte Website abrufbar war. Skepsis in Bezug auf den Erfolg dieser Erhebungsmethode, die auf Kategorisierung abzielt, hat sich im nachhinein als unnötig erwiesen, da das quantitative Ergebnis zufriedenstellend ausgefallen ist. Die Rückmeldungen erfolgten ausserdem prompt.

<sup>15</sup>Im Hinblick auf die Tatsache, das Netzanbindung in anderen Teilen der Welt rar ist, ist auch anzunehmen, dass es sich auf diesen Gebiete beschränkt.

Von insgesamt 40 Antworten erfolgten 23 bereits innerhalb der ersten 24 Stunden nach Aussendung des Aufrufs. Zwei Rückmeldungen waren leider durch einen Fehler in der Übermittlung unvollständig und wurden nicht berücksichtigt. Demzufolge beziehen sich alle folgenden Fragebogen Ergebnisse auf Antworten von 38 Personen.

Mehr als die Hälfte derer, die den Fragebogen ausgefüllt haben, haben sich mit Namen und Email Adresse zu erkennen gegeben, was jedoch nur 'optional' verlangt war. Bei erfahrenen NetzbenutzerInnen lässt das klar auf eine grosse Unterstützungsbereitschaft schliessen, was den Repräsentationswert der Umfrage hebt. In vier Abschnitten wurden Fragen in Bezug auf Soziales, Aufgabenkoordination, Arbeitszufriedenheit, usw. gestellt. Der erste Abschnitt enthielt dabei Kernfragen zum persönlichen Profil der Teilnehmer<sup>16</sup>.

Diejenigen die diese Umfrage unterstützt haben, sind zu mehr als 60% bereits länger als 2 Jahre in der Community. 13% sind es länger als die pd-list Mailingliste besteht, also länger als 4 Jahre. 23 von 38 (60,5%) waren innerhalb des letzten Jahres auch Teil von anderen Software Entwicklungsgruppen.

Viele Künstler, vor allem Musiker sind Teil dieser Gruppe. PD bildet eine Basis, die die beiden Felder Kunst und Technik vereint thematisiert. Es lässt sich hier nur im Einzelfall eine Differenzierung in der Interpretation dieser beiden Felder in Bezug auf die Arbeit mit dem Programm finden. Der Übergang zwischen jenen, die sich als Künstler mit guten Programmierkenntnissen bezeichnen und jenen, die sich als Programmierer sehen, die ihren Code kreativ einsetzen, ist fließend.

Im Fragebogen wurde um eine Eigeneinschaetzung in einer fünfstufigen Skala zwischen "user" und "developer" gebeten. Im Kontext von PD, dass als Programm auf das Zielpublikum MusikerInnen/KünstlerInnen ausgelegt ist, könnte das auch als eine Skala interpretiert werden, die zwischen dem Einsatz von Pd als musikalisches Instrument und der Programmierung von PD verstanden werden kann. Die Auswertung ergab eine grosse Ausgewogenheit in der Bandbreite zwischen user und developer mit einem geringen Mehrgewicht auf Seite der users:

	<i>user</i>	< >	<i>middle</i>	< >	<i>developer</i>
%	18,4	21,1	26,3	21,1	13,2

Die Art der primären beruflichen Verpflichtungen in der befragten Gruppe scheinen nicht stark von PD-relevanten Themen abzuweichen. Diese Tätigkeiten, die frei eingetragen werden konnten, lassen in 29:9 Fällen vermuten, dass sie zumindest in verwandten Arbeitsbereichen ansetzen. Auf die Frage, ob ein direkter oder indirekter monetärer Verdienst aus der Arbeit mit PD entstünde, stimmten 36,8% der Befragten zu, 50% verneinten, 13,2% fanden die Frage unangebracht.

Als durchschnittliches Alter wurde zu mehr als 65% 20-29 Jahre angegeben. Der Rest ist älter. Unter Zwanzigjährige gab es keine.

<sup>16</sup> In Bezug auf den Fragebogen wird auf geschlechtsneutrale Formulierung verzichtet, da die befragte Gruppe ausschliesslich männlich war.

PD'lers sind vorwiegend männlich. Die befragte Gruppe war es gar zu 100%.

Was sich nicht nur in der Umfrage stark hervorhob, sondern auch in den teilnehmenden Beobachtungen, war, dass es ein sehr ausgeprägtes Gruppenbewusstsein innerhalb der Gemeinschaft gibt. An den Face-to-Face Treffen lokaler Linux User Groups (meist LUG-Stammtische genannt) orientieren sich Treffen der PD Community. Derartige Gelegenheiten gibt es in Wien, Berlin, Ghent, New York, Koeln und Barcelona. Anstatt, wie es die LUG's betreiben, nämlich in Gaststätten gemütlich zu plaudern, verlegen lokale PD-Gruppen ihre Treffen in Clubs und nutzen das um ihre Arbeit vor Publikum zu präsentieren.

Spätestens hier scheint sich herauszukristallisieren, dass die Dynamik um dieses Programm starken subkulturellen Charakter hat. Der Reputationsfaktor, dem in Bezug auf freie Kollaborationsformen im Netz grosse Wichtigkeit zugeschrieben wird, kommt als Qualitätsmerkmal auch im Bereich der Kunst zum Tragen. Im lokalen Umfeld kleiner Szenen ist das ähnlich. Die Steigerung des individuellen Reputationswertes innerhalb eines kurzen angesichtlichen Treffens folgt natürlich anderen Regeln, als bei der Entwicklung eines Produktes online. Trotzdem fördern sie sich gegenseitig. Eine breitere Öffentlichkeitswirksamkeit durch solche regulären Treffen könnte übrigens die Möglichkeit bieten, die Homogenität der Gruppe z.B. zugunsten einer Auflockerung des Geschlechterverhältnisses zu verändern.

#### **4.4.2.) Prozessfaktoren**

##### **Projektarchitektur**

Pd erlaubt ein modulares Einbinden von externen Programmteilen, sog. External. Das bedeutet, dass es möglich ist, den Programmkern durch einzelne externe Programmteile erweitert, zu benutzen. Der Änderungs- und Distributionsprozess des Kernes obliegt klar hierarchisch Miller Puckette. Das Schreiben und Verteilen von externen Programmfunktionen - die beim Ausführen des Programmes übrigens nicht mehr von den Programmkern immanenten Teilen unterschieden werden können - steht Allen frei. Diese Projektarchitektur ermöglicht und erlaubt ein wildes Spriessen von externen Programm-Fragmenten während Pd als Kern unangetastet bleibt.

Die Möglichkeit diese External in den Betrieb des Programmes modular einbinden und somit programmiertechnisch experimentell ansetzen zu können, ohne PD an sich verändern zu müssen, klammert viele Konfrontationsgründe von vornherein aus, die eine andere Projektarchitektur mit sich bringen würde. Will jemand über eine bestimmte Funktion unter PD verfügen, die in einem External vorhanden ist, kann diese Person das Programm in der Art starten, dass sie verfügbar ist. Andere müssen das nicht. Eine Modularität in dieser Art ermöglicht es, eine nahezu unbegrenzte Vielfalt an External anzubieten. Eine Vielfalt, die mittlerweile bereits existiert und die ursprüngliche Einschätzungen längst übertroffen hat. Schwerwiegend in dieser Hinsicht ist allerdings, die Verwaltung und Beschreibung dieser External, da ein

Konzept zur Namensgebung oder Funktionsbezeichnung nicht besteht. Die entsprechend dokumentierte Aufbereitung der Externalis stellt derzeit eine wichtige Aufgabe innerhalb der Projektgruppe dar.

### **Teamführung**

Miller Puckette besitzt unumstritten die höchste Autorität innerhalb der Community. Er hat das Programm PD (oben als Programmkern beschrieben) geschrieben und war massgeblich am Aufbau der Community beteiligt. Mittlerweile haben auch andere Programmierer an PD mitgestaltet. Die Portierung des Programmes auf andere als dem ursprünglichen Betriebssystem<sup>17</sup> ist beispielsweise eine der Leistungen, die im Programm PD selbst festgeschrieben werden mussten. Derartige Erweiterungen hat Miller Puckette gerne aufgenommen. Trotz der autoritären Haltung Miller Puckettes in Bezug auf die Erweiterungen des Programmes PD, genießt er höchstes Ansehen innerhalb der Community. Entscheidungen, in denen er Vorschläge abgelehnt hat, werden im Nachhinein als Positiv gewertet.

Miller Puckette ist - in kleinen Abständen aber doch permanent - innerhalb der Community präsent. Auf den zur Entwicklung relevanten Mailinglisten kommentiert er regelmässig Vorschläge und kollaboriert mit anderen Community Members in der Verbesserung des Programmes.

Einzelne Teammembers positionieren sich selbst innerhalb der Community abhängig von ihren spezifischen Fähigkeiten, ihren Leistungen und ihrem Kommunikations-Geschick.

### **Aufgabenkoordination - Zielorientierung**

Grundsätzlich kann man sagen, dass es keine zentral gelenkte Aufgabenkoordination gibt. Es gibt maximal informelle Schwerpunktsetzungen innerhalb der Gemeinschaft, wobei sich einzelne Personen demzufolge bestimmter Aufgabenbereiche annehmen. Die Anzahl derer, die diese Software im Studienkontext oder zur Realisierung selbst- oder fremdbestimmter Projektaufgaben einsetzen, ist nicht zu unterschätzen. Dies hat zur Folge, dass Aufgaben meist funktionsorientiert sind und nicht unbedingt innerhalb der Community gestellt werden. Die Zielorientierung ist somit meist an die Lösung einer konkreten Anwendung gebunden. Der effizienteste Weg noch nicht vorhandene Funktionalitäten zu realisieren, ist mit Anderen an ihrer Entwicklung zu kollaborieren.

### **Motivation - Verantwortung**

Die Motivation von PD'lers zur Programmierung entsteht -wie eben erwähnt - durch das Interesse an bestimmten Funktionalitäten. Bei selbstgestellten Aufgaben ist der künstlerische Hintergrund einzelner Personen ein massgebender Ausschlag dafür. Ihr Engagement rührt aber zum Teil auch von gruppendynamischen Prozessen. Pd wird

---

<sup>17</sup> z.B. die Portierung auf Linux am IEM durch Guenther Geiger

in der gruppeninternen Reflexion sehr positiv wahrgenommen. Über das Internet verbunden finden sich in der Gruppe sehr spezifisch ausgebildete Personen, die sich gegenseitig optimal unterstützen. Das Entstehen von lokalen Kleingruppen, die sich in F2F Situationen treffen, wird jeweils auch in der internationalen Mailingliste angekündigt. Über das rein technische Lösungs-Engagement hinweg entsteht auf diese Weise ein grosses Identifikationspotential mit der Gruppe, das letztendlich auch als Motivationsfaktor in Bezug auf die (künstlerische) Produktion mit PD auswirkt.

PD'lers arbeiten eigenverantwortlich.

## Kommunikation

Kommuniziert wird online und offline. 22 der 38 Befragten haben angegeben auch ihre Freizeit mit anderen Gruppenmitgliedern zu verbringen.

In Bezug auf PD-spezifische Themenstellungen wird aber vorwiegend die Mailingliste pd-list als verbindendes Element herangezogen. Der Email Durchlauf auf dieser Liste hat sich vom ersten Halbjahr 1998 mit 167 Nachrichten auf ueber 3200 Nachrichten im ersten Halbjahr 2002 gesteigert. Die durchschnittlich 20 Emails pro Tag haben auch dazu geführt, die Mailingliste zu kategorisieren. Das ermöglicht entweder einen speziellen Informationsbereich auszuwählen und nur diesen zu beziehen, oder denjenigen die auf mehreren Listen subskribiert sind, einen gut ersichtlichen Hinweis auf die Informations-Kategorie zu erhalten.

Neben dieser Liste werden auch andere Werkzeuge benutzt. In der Umfrage wurden folgende Häufigkeiten in Bezug auf anderen Kontaktformen angegeben (Angaben in %).

	<i>Never / Not applicable</i>	<i>Less than once a month</i>	<i>About once a month</i>	<i>About once a week</i>	<i>A few times a week</i>	<i>Daily</i>
Mailing List	5,3	5,3	15,8	18,4	31,6	23,7
Personal Email	13,2	23,7	15,8	21,1	18,4	7,9
Personal Telephone Call	63,2	15,8	7,9	7,9	0,0	5,3
Face-to-face interaction	31,6	26,3	15,8	15,8	7,9	2,6
irc (Chat)	92,1	5,3	2,6	0,0	0,0	0,0

Angaben in %.

Zum Informations- und Datenaustausch wird ein enorm breites Spektrum an informationstechnologischen Hilfsmitteln eingesetzt. Im Folgenden eine Auflistung

und Kurzbeschreibung der Hilfsmittel, die wir im Fragebogen zur Bewertung angeboten haben.

Community Plattform <http://www.pure-data.org>

Es handelt sich hierbei um eine Website mit Community Funktionen. Sie verfügt über eine BenutzerInnen-Verwaltung (Februar 2003, 621 registrierte BenutzerInnen) und zeigt stetig die Anzahl derer an, die gerade Online sind. Ein Teilbereich bietet Software-Downloads an, ein anderer listet aktuelle Themenstellungen. Die Site basiert auf einem Content Management System namens 'Post Nuke', welches den funktionellen Grund für die dynamische Verwaltung von Community Inhalten bietet.

Websites: <http://crca.ucsd.edu/~msp/> oder <http://iem.kug.ac.at/pd>

Die persönliche Website von Miller Puckette enthält Informationen zu seiner Person und bietet das Programm PD zum Download an. Die PD Site des Instituts für Elektronische Musik bietet relevante Links zu PD.

Webbased Wiki: <http://pd.iem.at/pdwiki/>

Wiki ist ein Werkzeug zur gemeinschaftlichen Erstellung und Pflege von Hypertext-Inhalten. Es unterliegt keiner zentralen Kontrolle. Jede Person kann Inhalte einbringen und diese sofort über eine Wiki-Website verfügbar machen.

Innerhalb der PD-Gruppe ist die Intention dieser Plattform alle einzuladen und ihre Beiträge innerhalb einer sich sukzessive formenden Struktur einzubinden.

Datenbank mit External: <http://pd.iem.at/pdb>

Eine zentrale Datenbank, die ein Suchinterface anbietet um einzelne PD-Programmkomponenten zu finden. Norbert Math hat diese Datenbank programmiert und verwaltet sie auch langfristig. Es gibt mehrere hundert Einträge.

CVS (Concurrent Versions System): <http://sourceforge.net/projects/pure-data>

Ein CVS ist ein Versions Kontroll System. Wie tausende andere Open Source Software Projekte<sup>18</sup> ist PD auch innerhalb der Plattform Sourceforge vertreten. Die PD Sektion unter Sourceforge ist Mitte 2002 entstanden und zielt darauf ab, die sog. External (Erweiterungen) von PD in ihrem aktuellen Programmierstatus anzubieten. 16 PD EntwicklerInnen sind eingetragen.

---

18 50.965 'Hosted Projects' am 18.11.2002

Die Häufigkeiten des Einsatzes dieser Werkzeuge sind in Prozent folgendermassen ausgefallen:

	<i>Never / Not applicable</i>	<i>Less than once a month</i>	<i>About once a month</i>	<i>About once a week</i>	<i>A few times a week</i>	<i>Daily</i>
Community Plattform	21,1	21,1	28,9	23,7	5,3	0,0
Websites	5,3	31,6	39,5	15,8	7,9	0,0
Wiki	39,5	31,6	18,4	7,9	2,6	0,0
Datenbank	13,2	42,1	23,7	15,8	5,3	0,0
CVS	39,5	21,1	23,7	7,9	5,3	2,6

Angaben in %.

Zur Vernetzung unter den einzelnen Website-basierenden Informationsoberflächen gibt es mittlerweile einen Webring<sup>19</sup>. Dieser verbindet 12 verschiedene Websites untereinander.

#### 4.5.) Herausforderungen

Die Entwicklung des Programmes PD in seiner Produkt-Sparte und unter den gegebenen Voraussetzungen ist ein ehrgeiziges Unternehmen. Von einer einzelnen Person initiiert, ist nunmehr eine Struktur gewachsen, die auch für OS Entwicklung durchaus unüblich dezentral ist. Die extrem offene Architektur des Projektes, bestehend aus dem eigentlichen Programmkern und einer nicht exakt quantifizierbaren Grösse an Programmierleistungen in Form von Externalen und anderen Programmiererweiterungen<sup>20</sup>, bildet die Ursache dafür, dass es schwer möglich ist, die Grösse eines Kernentwicklerteams zu definieren<sup>21</sup>. Zudem ist die Differenzierung zwischen "einfachen" BenutzerInnen und ProgrammierInnen nur unscharf. In der Mailingliste pd-list finden sich sowohl AnwenderInnen als auch Kernentwickler<sup>22</sup>.

Trotzdem PD'lers über sehr spezifische Kenntnisse oder Interessen verfügen und ein klares Profil in Bezug auf Geschlecht, Alter und auch Bildung abgeben, wird die Gruppe von ihren Mitgliedern als nicht sonderlich homogen beschrieben.

Das grosse Wachstum der Gruppe innerhalb kurzer Zeit stellt neue Herausforderungen an die Community. Als besonders wichtig für das Projekt wird die Verbesserung der Kritikfähigkeit der einzelnen Mitglieder, aber auch der Community

<sup>19</sup> <http://pd.klingt.org/webring/info.html>

<sup>20</sup> In Form von: Externalen, Libraries, Abstractions

<sup>21</sup> In der Umfrage schwanken die Angaben dazu zwischen 1 und 20. Zusätzlich hätte ein Verweis darauf, ob sich die Angabe auf den Programmkern alleine oder auf die gesamten Programmierleistungen beziehen sollen, vermutlich genauere Angaben bewirkt.

<sup>22</sup> 'Core developers' sind im Fall von PD ausschliesslich männlich.

als Ganzes, in beide Richtungen - gebend und nehmend - erwähnt. Auch das Reduzieren egozentrischer Manier in Mitteilungen an die Liste wurde mehrmals unabhängig voneinander genannt, denn sie untergräbt den im wesentlichen technischen und funktionellen Charakter der Community.

Die Projektmailingliste, ist zwar durch das Wachstum stark frequentiert, wird aber immer noch allgemein als gut funktionierend beschrieben. Kommuniziert wird in Englisch, was denen, die Englisch als Muttersprache haben, entgegenkommt. Sprachprobleme verursachen natürlich gelegentlich Kommunikationsprobleme, wirken sich aber erfahrungsgemäss innerhalb dieser Gruppe nicht als Ursache ernsthafter sozialer Differenzen aus.

Das Wachsen der Community, das Entstehen verschiedener Web-Plattformen und das Einbinden von vielfältigen Werkzeugen zum Daten- und Informationsaustausch sind in Referenz zueinander verlaufen. Die Vielfalt der Kommunikationsmittel bewirkt eine verstärkte Dezentralisierung der Projektaktivitäten. Nicht von allen ProtagonistInnen der Community wird das als ausschliesslich Positiv bewertet.

Folgende Probleme wurden dabei genannt:

- Schwerfälligkeiten in Bezug auf die Durchführung von Änderungen seien durch den Versuch einer Konsensfindung unter zu vielen Beteiligten zurueckzuführen;
- Redundanzen in den Bemühungen Einzelner;
- Eine zentrale Adresse, die Dokumentation, Beispiele, Programmiererweiterungen, etc. verwaltet, fehlt.

Einige Weichen zur Koordination dieser neuen Anforderungen wurden bereits gestellt. So stellt das neue CVS eine Möglichkeit dar, Programmiererweiterungen zentral zu verwalten und weiterzuentwickeln. Wie alle neuen Lösungsansätze aber, stellen sie für die gesamte Gruppe grosse Herausforderungen dar.

Es erscheint allerdings wahrscheinlich, dass die PD'lers diese Probleme meistern werden, denn ihre Motivation ist hoch. Was sie nämlich besonders auszeichnet, ist die starke Zufriedenheit innerhalb der Gruppe. Was der abschliessende Auszug aus der Fragebogen Auswertung ergibt, hat sich in allen anderen Formen der Recherche bestätigt: Die Arbeit an PD wird als äusserst befriedigend und lohnenswert empfunden.

Our group morale is high.

<i>not applicable</i>	<i>strongly disagree</i>	<i>disagree</i>	<i>agree</i>	<i>strongly agree</i>
13,2 (5)	0,0 (0)	2,6 (1)	63,2 (24)	21,1 (8)

I enjoy being a member of this community.

<i>not applicable</i>	<i>strongly disagree</i>	<i>disagree</i>	<i>agree</i>	<i>strongly agree</i>
2,6 (1)	0,0 (0)	0,0 (0)	52,6 (20)	44,7 (17)

I goin intrinsic reward from it.

<i>not applicable</i>	<i>strongly disagree</i>	<i>disagree</i>	<i>agree</i>	<i>strongly agree</i>
7,9 (3)	2,6 (1)	10,5 (4)	50,0 (19)	28,9 (11)

It's fun.

<i>not applicable</i>	<i>strongly disagree</i>	<i>disagree</i>	<i>agree</i>	<i>strongly agree</i>
2,6 (1)	2,6 (1)	0,0 (0)	44,7 (17)	50,0 (19)

It' fame.

<i>not applicable</i>	<i>strongly disagree</i>	<i>disagree</i>	<i>agree</i>	<i>strongly agree</i>
15,8 (6)	21,1 (8)	34,2 (13)	23,7 (9)	5,3 (2)

Angaben in % (Personen)

Die Auswertung der gesamten webbasierenden Umfrage liegt als Anhang bei.