Softwareentwicklung im Deutschen Wetterdienst

Seminar Softwareentwicklung in der Wissenschaft
Theresa Eimer

Betreuer: Dr. Hermann Lenhart

Inhalt

1 Vorwort

1.1 Zielsetzung

Der Deutsche Wetterdienst ist für das Thema "Softwareentwicklung in der Wissenschaft" insofern interessant als er nicht nur Software entwickelt, die von Wissenschaftlern genutzt wird und entsprechenden Standards genügen muss, sondern auch als Behörde eine klar definierte langfristige Aufgabenstellung hat, wie es in der Wissenschaft nicht zwingend üblich ist. Dadurch lohnt es sich deutlich mehr als etwa im universitären Bereich, Programme nicht nur auf korrekte und möglichst schnelle Ergebnisse, sondern auf Benutzbarkeit auszulegen und gegebenenfalls auch deutlich größere Projekte zu realisieren, die die alltäglichen Aufgaben der Meteorologen beim Deutschen Wetterdienst vereinfachen und automatisieren. Somit steht diese Art der Softwareentwicklung zwischen derjenigen für Unternehmenskunden und der in der Wissenschaft. Die Projektgröße und die zur Entwicklung benutzten Methoden variieren trotzdem noch. Zwei Eigenentwicklungen, die das sehr gut zeigen werden hier vorgestellt: Das spezifischere und kleinere MetMaster und die deutlich größere Ninjo Workstation. MetMaster wurde dabei von einem der Meteorologen, meinem Interviewpartner Herrn Dr. Thomas Bruns, entwickelt. Im Gegensatz dazu ist Ninjo das Projekt eines internationalen Zusammenschlusses mit Beteiligung des Deutschen Wetterdienstes. Beide sind wichtige Hilfsmittel im Arbeitsalltag in der Hamburger Niederlassung des Wetterdienstes.

1.2 Zur Methodik

Die Grundlage meines Vortrags und auch dieser Ausarbeitung ist ein Interview mit dem Referatsleiter des Seewetterdienstes in Hamburg, Herrn Dr. Thomas Bruns. Er ist promovierter Meteorologe, hatte also zwar während seiner Ausbildung durchaus Kontakt mit dem Thema der Softwareentwicklung, aber in anderen Kontexten und in einer anderen Tiefe als ein Informatiker es gehabt hätte. Dementsprechend ging es auch in dem Interview weniger um technische Details als um die Entstehung seines Programms MetMaster, das im Deutschen Wetterdienst Hamburg zur Vorhersage der Wetterlage für Schiffsrouten benutzt wird, sowie auch ein wenig um die Ninjo Workstation Software, bei der er allerdings nur Nutzer ist. Deren Entwicklungsleiterin Sibylle Hauke war jedoch so freundlich mir einige Schulungs- und Konzeptfolien zukommen zu lassen, denen ich vieles zu Ninjo selbst und vor allem zum Entwicklungsprozess entnehmen konnte.

2 Der Deutsche Wetterdienst

2.1 Der Deutsche Wetterdienst allgemein

Der Deutsche Wetterdienst (im Folgenden auch DWD) ist eine Anstalt des öffentlichen Rechts, deren Aufgabenspektrum auf dem "Gesetz über den deutschen Wetterdienst" basiert. Dazu gehören unter anderem die Herausgabe von Unwetterwarnungen, die Bereitstellung von aktuellen Wetterdaten und Vorhersagen für Luft- und Seefahrt sowie Dienstleistungen für Privatpersonen oder Firmen, die Auskünfte über Wetter oder Klima benötigen[1]. Mit einem Messnetz und einem eigenen Klimamodell werden dafür die entsprechenden Daten auf einem Großrechner in Offenbach berechnet. Mitarbeiter in den sechs großen Niederlassungen, unter anderem Hamburg, und 20 kleineren Außenstellen und Flugwetterwarten können diese dann von einem FTP-Server anfordern, um so auf Anfragen in ihrem Zuständigkeitsbereich zu reagieren oder die aktuelle Wetterlage zu überwachen. Darüber hinaus gibt es auch internationale Kooperationen, etwa in der Forschung oder in der Entwicklung der Ninjo Workstation Software.

2.2 Die Niederlassung Hamburg und Herr Dr. Bruns

Die DWD Niederlassung Hamburg hat eine spezielle Rolle im Wetterdienst; sie beschäftigt sich als einzige mit dem Seewetter. Dadurch ist sie Ansprechpartner für Wetterfragen auf See sowie mit zwei Schiffen auch Ausgangspunkt von Forschungsreisen und zuständig für alle Fragen, die mit dem Meer zusammenhängen. Passenderweise wird sie entsprechend als Seewetteramt bezeichnet. Ein Referat im Seewetteramt ist die Seeschifffahrtsberatung, wo Schiffe beraten werden, die herausfinden wollen, ob sie ihre Fahrt durchführen können und was die wettertechnisch beste Route wäre. Dr. Thomas Bruns leitet dieses Referat. Vor seiner Zeit beim DWD Hamburg promovierte und arbeitete er an der Universität Hamburg und am Max-Plank-Institut Hamburg als Meteorologe.

[1] §4.1 BGBl S. 2871

3 MetMaster

3.1 Über das Programm

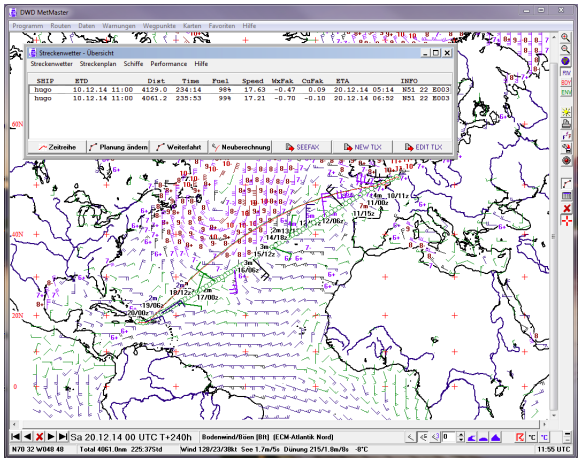

Abb. 1: Eine Route

MetMaster ist entstanden, um die manuelle Arbeit des Abgleichens einer Schiffsroute mit dem Wetter entlang der Route zu automatisieren. Die Funktionalität des Programms war also klar umrissen: Die aktuellen Wetterdaten und Prognosen müssen eingelesen werden und mit deren Hilfe dann eine schnelle und ungefährliche Route gefunden werden (Abb.1).

Es kann dabei ausgewählt werden aus welchem Klimamodell die Daten importiert werden (Abb. 2). Außerdem können auch Analysen der Wetterdaten für eine bestimmte Region auf diese Weise angefordert und für die Berechnung genutzt werden.

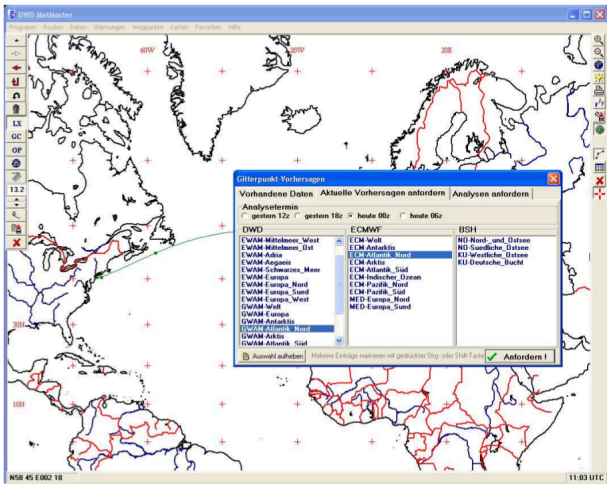

Abb. 2: Klimamodelle

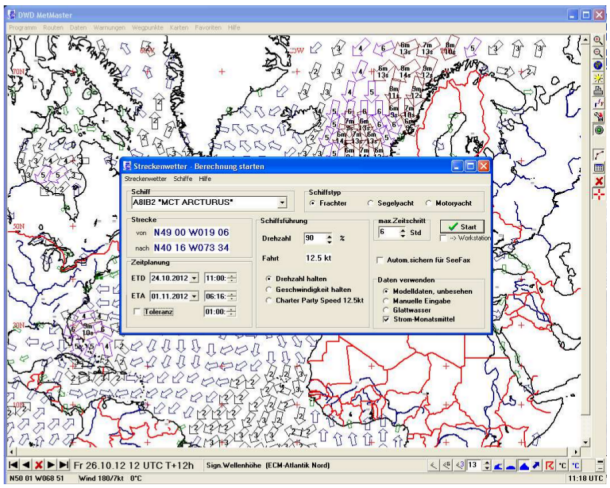

Abb. 3: Ein Datenbankeintrag

Um diese Berechnungen für jedes Schiff anzustellen, werden einige Daten wie zum Beispiel der Tiefgang oder die Länge des Schiffes benötigt. Diese werden in einer Schiffsdatenbank gespeichert (Abb. 3). Es können einige optionale Punkte, wie etwa der Kraftstoffverbrauch, gespeichert werden, die dann detailliertere Ausgaben erlauben - in diesem Fall eine Einschätzung wie hoch der Verbrauch auf der gewählten Route wäre. Zusätzlich kann das Programm die Route und die Wetterdaten visualisieren und die Ergebnisse der Berechnungen direkt über den offenbacher FTP-Server des DWD als E-Mail oder Fax an den Kunden weiterleiten. Bei der Visualisierung gibt es einige Auswahlmöglichkeiten, etwa zwischen Wind- und Wellenstärke, außerdem können Ländergrenzen und Flüsse eingefärbt werden. Insgesamt ist das jedoch nicht der Fokus von MetMaster und deshalb ist die graphische Oberfläche auch auf Funktion und nicht auf Design ausgelegt.

3.2 Technische Daten

MetMaster ist in Pascal geschrieben, benutzt wird dafür die Entwicklungsumgebung Delphi, die Windows-spezifisch ist und dementsprechend MetMaster nur auf Windows lauffähig. Für einige Funktionen, zum Beispiel den FTP Zugriff, wurden außerdem Plugins benutzt. Das Programm ist objektorientiert aufgebaut, der Programmcode verteilt sich auf 129 Dateien, die insgesamt etwa vier Megabyte groß sind. Es ist also relativ viel Quellcode für nur einen Entwickler, aber MetMaster ist ein verhältnismäßig kleines Programm. Ein Benutzerhandbuch oder eine umfassende Dokumentation existiert nicht.

3.3 Entwicklung

Da Herr Dr. Bruns Meteorologe ist, konnte er durchaus programmieren, allerdings war das natürlich nie seine Hauptaufgabe. Im Interview berichtete er, wie er und einige Kollegen in den Neunzigern versucht hatten Arbeitsabläufe

zu automatisieren, die bisher noch per Hand hatten durchgeführt werden müssen. Die Prognosen für die Seeschifffahrt gehörten ebenfalls dazu und so entwickelte er neben seiner Tätigkeit beim DWD MetMaster um seinen Arbeitsalltag zu erleichtern. Das Programm ist dementsprechend eher langsam entstanden und immer wieder um einige Funktionen erweitert worden. Besondere Techniken oder Hilfsmittel wurden bis auf einige Plugins nicht verwendet. Auch eine Versionsverwaltung gab und gibt es nicht. Im Moment ist MetMaster fertiggestellt, es gibt keine Pläne zur Erweiterung der Funktionalität, aber damit es wartbar bleibt, wird es gerade gut dokumentiert, was bisher noch kaum der Fall ist. Insgesamt wurde die Entwicklung ohne größeren Einsatz von Softwareentwicklungstechniken durchgeführt, da Herr Dr. Bruns dieses Handwerkszeug nicht zu seiner Verfügung hatte. Allerdings hat er sich durchaus mit einigen Themen auseinandergesetzt; so war MetMaster anfangs etwa rein prozedural aufgebaut und wurde dann umstrukturiert. Dadurch dass es aber keine Zusammenarbeit mit anderen Entwicklern gab und das Projekt noch von überschaubarer Größe ist, gab es bis auf die fehlende Dokumentation keine Schwierigkeiten aufgrund der fehlenden Kenntnisse aus der Softwaretechnik.

4 Ninjo

4.1 Über das Programm

Ninjo wurde im Jahr 2000 als eine Kooperation des Deutschen Wetterdienstes mit dem Geoinformationsservice der Bundeswehr ins Leben gerufen. Heute sind außerdem die Wetterdienste von Dänemark, Kanada und der Schweiz sowie drei private Firmen an dem Projekt beteiligt.

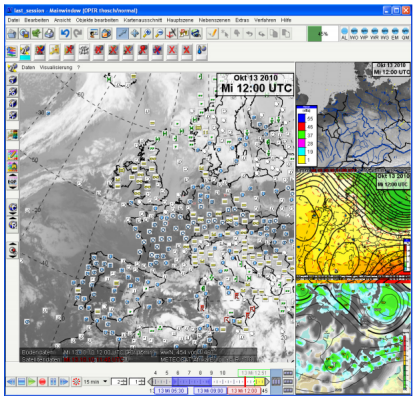

Abb. 4: Eine Beispielsession

Koordiniert wird die Entwicklung durch das "Ninjo Konsortium", das aus über 60 Personen besteht, unter anderem auch Entwicklern von Beratungsfirmen, die an der Konzeption und Entwicklung an über 10 Standorten beteiligt sind. Entsprechend ist auch das Programm sehr umfangreich: Ninjo ist ein Visualisierungs- und Datenverarbeitungsprogramm, das darauf abzielt, möglichst viele Funktionen, die in Wetterdiensten benötigt werden in sich zu vereinen. Dabei wird ein modularer Aufbau verwendet. Das Basissystem, das aus Servereinheit und einem Client besteht, wird ergänzt durch sogenannte Layer, die jeweils eine bestimmte Funktion erfüllen.

In der Anwendung können mehrere Ansichten in einem Fenster nebeneinander in veränderbaren Größen betrachtet werden. Ein solches Fenster wird dann als eine Session bezeichnet, deren Layout und Inhalt auch speicherbar sind. Eine Ansicht besteht wiederum aus einem oder mehreren Layern. Die Layer sind untereinander durchaus verschieden und können übereinander gelegt werden wenn sie kombiniert gebraucht werden.

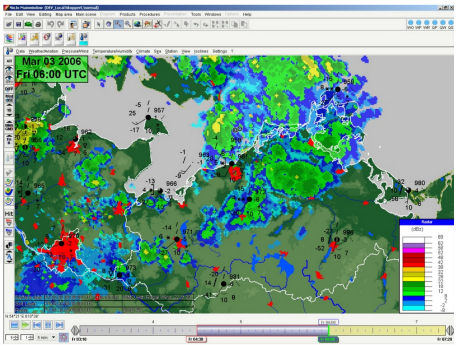

Abb. 5: Radar Layer

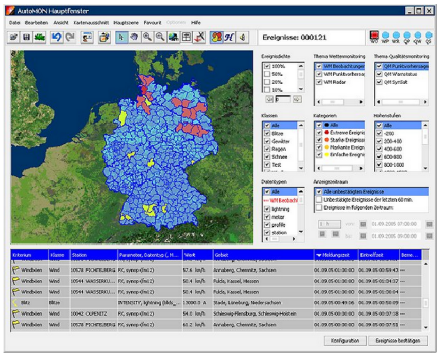

Abb. 6: Automatisches Monitoring

Die Bandbreite reicht dabei von Layern mit geographischen Daten über solche, die aktuelle Wetterdaten wie etwa Radardaten anzeigen (Abb. 5) bis hin zu wiederum anderen, die automatisch Meldungen ausgeben, wenn eine abnormale Wettersituation eintritt (Abb. 6). Durch die vielen verschiedenen Layer, das für ein solches System natürlich auch entsprechend komplexe Basissystem und die laufende Erweiterung, ist Ninjo sehr umfangreich und wird wohl auch weiterhin wachsen.

4.2 Technische Daten

Im Gegensatz zu MetMaster ist Ninjo plattformunabhängig; sowohl der Serverteil als auch der Client können auf Linux und auch auf Windows betrieben werden. Während der Entwicklung hat sich allerdings die Kombination aus einem Linux Server und dem Client auf Windows als am effizientesten herausgestellt[2]. Der Faktor Laufzeit ist durch die große Menge an abzurufenden und zu verarbeitenden Daten natürlich nicht zu vernachlässigen.
Ninjo ist komplett in Java programmiert. Zur Entwicklung werden außerdem die Entwicklungsumgebung Eclipse und Testlink, ein Testmanagementsystem, benutzt. Um die Zusammenarbeit an so vielen Standorten zu koordinieren, werden Tools zum gemeinsamen Arbeiten benötigt. Dazu zählen die Projektmanagementsoftware JIRA genauso wie die Autobuildplattform Jenkins, Perforce als Versionsverwaltung und die Wiki-Software Confluence.

4.3 Entwicklung

Der Entwicklungsprozess bei Ninjo ist stark strukturiert. Da viele verschiedene Gruppen an der Software beteiligt sind, gibt es auch eine klare Hierarchie und Aufgabenverteilung (Abb. 7).

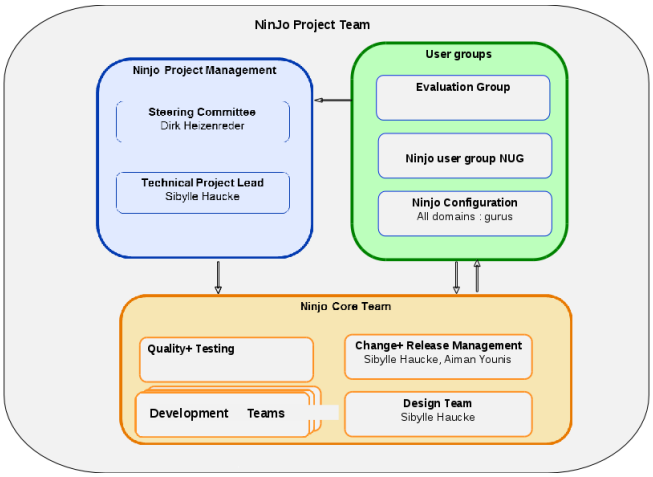

Abb. 7: Die Organisation

Die strategischen Ziele und die grobe Planung werden vom sogenannten "steering committee" festgelegt. Hier wird auch über die Finanzplanung entschieden. Mitglieder dieses Komitees sind Mitglieder des Ninjo Konsortiums sowie Lizenznehmer und die Treffen finden drei oder vier mal im Jahr statt. Eine Hierarchiebene darunter liegt die "CD group", das Design Team, das die technischeren Entscheidungen trifft. Es ist sowohl für die Architektur als auch für das Projekt- und Releasemanagement zuständig und plant die Aufgabenverteilung. Jeder Standort entsendet dafür den jeweiligen leitenden Entwickler. Zusätzlich werden auch andere erfahrene Softwareentwickler hinzugezogen. Die so festgelegte Architektur und Aufgaben werden durch die JIRA

[2]Koza, 2006

Software als Requirements oder Issues spezifiziert, sodass sie von den Entwicklern bearbeitet werden können.

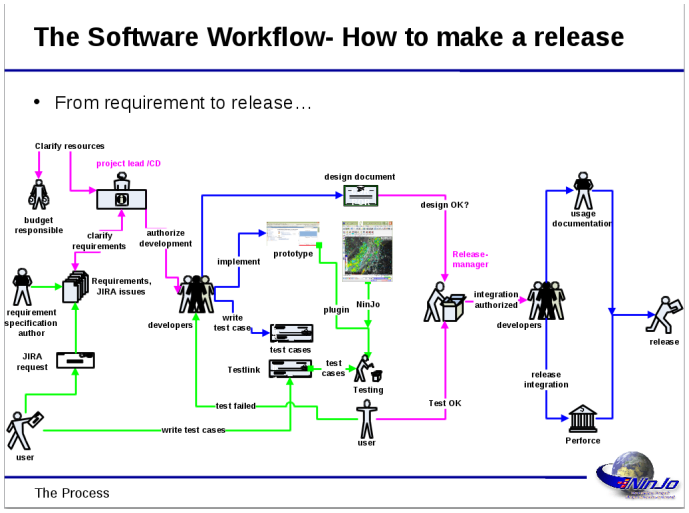

Abb. 8: Schematischer Prozessablauf

Dabei wird zunächst ein Entwurf erstellt, der in einem Prototyp implementiert und getestet wird bis ein "release manager" die bestandenen Tests akzeptiert und der Programmabschnitt integriert wird. Die Testfälle sollen in Anlehnung an die agile Programmierung auf den Anforderungen der CD group basieren und werden regelmäßig, in einigen Bereichen sogar täglich, ausgeführt. Außerdem gibt es Reviews für die Entwürfe und Teile des Codes. Zusammen mit verschiedenen Bestreben die Nutzer miteinzubeziehen, wie etwa im "steering committee", aber auch in einer jährlichen Nutzerkonferenz und in einigen Tests, die von Nutzern durchgeführt werden, sieht man hier eine Abwandlung der agilen Programmierung. Die in der agilen Programmierung sehr wichtige Komponente der Kommunikation ist bei Ninjo allerdings durch die geographische Verteilung und das große Team problematisch. Um sie trotzdem zu gewährleisten, wird nicht nur JIRA benutzt, sondern es kann durch ein System von verschiedenen Branches (Abb. 9) für verschiedene Entwicklungsstände, in der Versionierung und dem Confluence Wiki genau nachvollzogen werden in welchem Stadium sich eine Aufgabe gerade befindet. Das kann man auch auf Jenkins beobachten, wo eine Kennzeichnung existiert, je nachdem ob ein Teil des Projekts schon abgeschlossen ist, bearbeitet wird oder die Tests noch nicht besteht (Abb. 10).

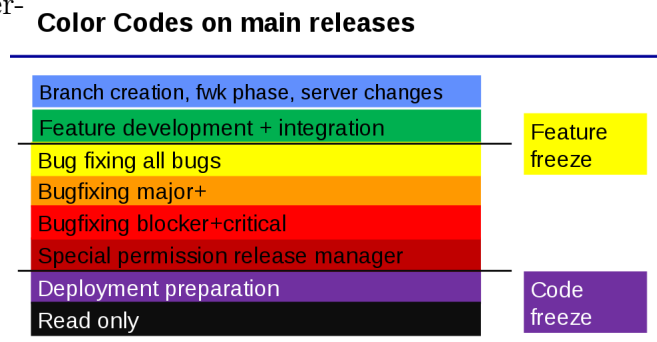

Abb. 9: Branches in der Entwicklung

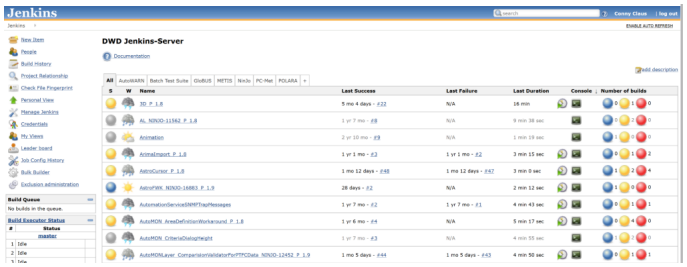

Abb. 10: Jenkins Interface

Jeder Entwickler wird angehalten, Tests für seinen Bereich selbstständig durchzuführen und nach Abschluss einer Aufgabe die entsprechenden Tutorials und Wikiartikel zu aktualisieren. So ist es möglich, benötigte Informationen relativ schnell zu finden und sich innerhalb des Projektes auf dem Laufenden zu halten. Der gesamte Ablauf ist sehr durchdacht, wie es für diese eher außergewöhnlichen Bedingungen auch notwendig ist. Durch Abwandeln agiler Methoden und dem Einsatz einiger Tools wurde ein System entwickelt, das eine ortsunabhängige Entwicklung in einem großen Team erlaubt, die, wenn man sie am Erfolg des Programmes misst, gut funktioniert.

5 MetMaster und Ninjo im Vergleich

5.1 Vergleich der Entwicklungsmethoden

Der direkte Vergleich der beiden Programme fällt natürlich schwer, da MetMaster und Ninjo in fast allen Punkten grundverschieden sind. Viele Unterschiede in den Entwicklungsmethoden sind dem Umfang der jeweiligen Programme geschuldet. Ein so großes Projekt wie Ninjo erfordert, besonders wenn es nicht an einem Ort entwickelt wird, eine gute Koordination der Arbeitsabläufe, anders als MetMaster, bei dem Herr Dr. Bruns freie Hand hatte. An dem genau festgelegten Arbeitsablauf im Ninjo Projekt sieht man das natürlich sehr gut. Je mehr Entwickler beteiligt sind, desto genauer müssen Absprachen und Richtlinien ausfallen. Auch Beschreibungen und die Dokumentation sind in diesem Zusammenhang von Anfang an deutlich wichtiger, da der eigene Code sehr wahrscheinlich relativ schnell von anderen nachvollzogen werden muss. Bei nur einem Beteiligten muss man für längere Zeit eher nicht davon ausgehen. Die Nutzung verschiedener Hilfsmittel zur Unterstützung der Entwicklung mag aber zum Teil auch an der größeren Verfügbarkeit der Tools und am größeren technischen Fachwissen der Ninjo Programmierer liegen. Eine Versionsverwaltung kann natürlich auch bei einem etwas kleineren Projekt ohne direkte Zusammenarbeit sinnvoll sein, allerdings muss man die entsprechende Software kennen und damit umgehen können. Auch die Definition von eigenen Best Practices deutet darauf hin, dass bei Ninjo viel Vorarbeit von erfahrenen Softwareentwicklern geleistet wurde. Das Projekt ist ein gutes Beispiel wie Software dank dem Internet auch stark verteilt entwickelt werden kann. MetMaster dagegen wurde unter sehr anderen Umständen entwickelt, nicht nur gab es nur einen Entwickler, MetMaster war auch nicht seine Hauptaufgabe. Außerdem waren die Ressourcen, die heute frei verfügbar sind in den neunziger Jahren noch keine Selbstverständlichkeit. Wenn Ninjo zeigt, wie sich heute neue Möglichkeiten zur Entwicklung ergeben, dann ist MetMaster eher ein Beispiel aus den Anfängen der Softwareentwicklung.

5.2 Zukunft der Programme

MetMaster ist ein fertiges Programm, bei dem allerdings noch einige Lücken in der Dokumentation geschlossen werden sollen, auch im Hinblick auf den nicht mehr allzu fernen Ruhestand von Herrn Dr. Bruns. Ein Mitarbeiter des DWD in Hamburg beschäftigt sich deshalb mit dem Code und versucht ihn mit Hilfe von Lazarus, einer Entwicklungsumgebung, die Delphi ähnelt, für Linux zu portieren. Außerdem gibt es, wenngleich noch recht vage,

Pläne, MetMaster in Ninjo zu integrieren. Dessen Entwicklung ist noch nicht abgeschlossen und es werden laufend bestehende Funktionen ausgebaut sowie neue Layer hinzugefügt.

6 Persönliches Fazit

Das Gespräch mit Herrn Dr. Bruns fand ich persönlich sehr interessant, genauso wie meine Recherche zum Thema Ninjo. Beide Programme und ihre Entwicklungsgeschichte sind auf ihre eigene Weise spannend, Ninjo aus informatischer Sicht natürlich noch etwas mehr. Der Entwicklungsprozess von Ninjo zeigt, welche Vorgehensweisen und Tools genutzt werden können, um einen sehr agilen Ansatz auch in großen Teams umzusetzen und gleichzeitig eine gute Kommunikation zu bewahren, selbst wenn die Entwickler sich nicht persönlich absprechen können und geographisch getrennt arbeiten.
Trotzdem war das Interview sehr informativ, nicht unbedingt was aktuelle Ansätze zur Softwareentwicklung betrifft, sondern im Hinblick auf die Anfänge der Softwareentwicklung im Seewetteramt Hamburg und auch auf die Erwartungshaltung an ein wissenschaftliches Programm. Nicht zuletzt ist dabei auch Dr. Bruns an sich zu erwähnen, der sich neben seiner Arbeit an MetMaster immer wieder mit Softwareentwicklung beschäftigte und aktiv versuchte sein Fachwissen auf dem Gebiet zu erweitern.
Softwareentwicklung in der Wissenschaft kann also beides sein, sehr groß, aber auch sehr klein. Die große Gemeinsamkeit beider Ansätze ist der Fokus auf den Nutzer. Herr Dr. Bruns hat MetMaster unter anderem für sich als Nutzer entwickelt, bei Ninjo gibt es mehrere Stellen im Prozess, an denen Nutzer beteiligt werden. Mehr noch als in der Wirtschaft ist für die Wissenschaft nämlich die Korrektheit und die Einhaltung von Konventionen wichtig. Es werden sehr viele und komplexe Daten auf einmal dargestellt und nur durch eine enge Zusammenarbeit mit der Zielgruppe kann garantiert werden, dass die Ausgaben auch verständlich und wissenschaftlich brauchbar sind. Eine, wie ich finde, gelungene Plattform für einen solchen Austausch bietet die jährliche Ninjo Nutzerkonferenz, bei der Wissenschaftler ausführlich mit den Ninjo Entwicklern über Probleme, Wünsche und Anregungen diskutieren können.

7 Quellen

- Interview mit Dr. Thomas Bruns
- Schulungsfolien des Ninjo Projekts
- Die Website des DWD:
  `dwd.de`
- Die Website von Ninjo:
  `www.ninjo-workstation.com`
- Der Vortrag von B. Koza beim 17. Treffen der EGOWS 2006 in Ungarn
  Zu finden unter:
  `http://www.ninjo-workstation.com/fileadmin/files/downloads/publications/Pub_EGOWS06_Koza.pdf`
- Handbuch zur Ninjo Version 1.3.5, Thomas Schubert, FU Berlin, 2010
- Bilder zu MetMaster: DWD Hamburg

Abbildungsverzeichnis