

Agiler Informatikunterricht: Soziale Aspekte der professionellen Softwareentwicklung im Schulunterricht erfolgreich erfahrbar machen

Timo Göttel
Angewandte und Sozialorientierte Informatik
Universität Hamburg
tgoettel@acm.org

Abstract: In diesem Artikel wird diskutiert, warum das Image der Informatik nach wie vor negativ behaftet ist. Es wird die These aufgestellt, dass dies zu Teilen darauf zurückzuführen ist, dass Informatikunterricht nicht in ausreichendem Maße die sozialen Aspekte der heutigen professionellen Softwareentwicklung (SE) berücksichtigt. Das Papier vermittelt, dass bereits beim ersten Kontakt mit Informatik darauf Wert gelegt werden sollte, dass es sich bei der Softwareentwicklung um eine gemeinschaftliche Arbeit handelt, bei der Lösungen immer im Dialog mit anderen Menschen entwickelt werden. Es wird dargelegt, dass in aktuellen Bildungsempfehlungen soziale Aspekte der Informatik zwar genannt werden, jedoch davon auszugehen ist, dass diese mangels Methoden kaum in Schulen vermittelt werden. Diese Ansicht wird gestützt durch Ergebnisse einer Lehrerbefragung. Dieses Papier stellt darüber hinaus dem Schulkontext angepasste Methoden der SE, bzw. der agilen Methoden vor. In Projekten mit Schülerinnen und Schülern wurden diese bereits mehrfach erprobt, so dass in diesem Artikel Hinweise zur erfolgreichen Anwendung gegeben werden können.

1 Einleitung

Die Informatik hat einen schlechten Ruf bei Studienanfängern. Die Anfängerzahlen bleiben deutschlandweit unter der möglichen Aufnahmekapazität der Universitäten. Dies ist besonders erstaunlich, handelt es sich doch bei den heutigen Studienanfängern um *digital natives* [Pr01], die mit der Informationstechnologie (IT) aufgewachsen sind und diese ausgiebig nutzen. Trotzdem sehen Jugendliche allem Anschein nach in der Informatik noch immer das befremdliche Berufsfeld, das nur für Solisten attraktiv ist [Ga09]. Demgegenüber sind gerade soziale Netzwerke beliebte und selbstverständlich genutzte Anwendungen von Jugendlichen. Offenbar als natürlich erscheint es ihnen, dass IT die soziale Interaktion, wie sie sie kennen und lieben, ermöglicht und gestaltet. Für sie ergibt sich daher auch nicht, dass das IT-Berufsbild einen ähnlichen interaktiven dynamischen und zwischenmenschlichen Charakter hat, der sich z.B. auch in der Art und Weise von social web Anwendungen widerspiegelt.

Die Informatik besitzt viele soziale und kreative Aspekte [Ro08b], die meist dem gemeinschaftlichen Lösen von Problemen dienen. Davon findet sich noch recht wenig im Schulunterricht wieder. Während es z.B. nach Romeike vielversprechende Ideen, Methoden und

Werkzeuge gibt, um Kreativität passend zu vermitteln [Ro08a], findet man in der Literatur kaum Hinweise auf Methoden oder Werkzeuge, um die sozialen Aspekte abzudecken. Es ist davon auszugehen, dass dieses Versäumnis auch einen Anteil an dem eingangs beschriebenen schlechten Ansehen der Informatik hat: Überspitzt formuliert, sitzen im Unterricht alle Schülerinnen und Schüler vor ihren Computern, und es gibt wahrscheinlich zwei bis drei Personen, die schnell Lösungen auf irgendwie magische Weise und aus dem Nichts erarbeiten. Das Bild der Einzeldisziplin, bei der kryptische Anweisungen in den Computer gehackt werden, ist geboren. Dieses Bild steht im Gegensatz zu Berufsbildern, die man in der professionellen Softwareentwicklung (SE) findet, wo Lösungen immer häufiger in Teamarbeit und durch eine Vielfalt von sozialer Interaktion entstehen. Naheliegender erscheint es daher zu untersuchen, wie die sozialen Aspekte der heutigen SE im Unterricht erfahrbar gemacht werden können. Besonders eignen sich hier als Bezugspunkt die agilen Methoden, die besonderen Wert auf soziale Interaktion und dynamische Prozesse legen. Diese Grundprinzipien sind in dem agilen Manifesto prägnant formuliert [Be01].

Im Weiteren wird zur Veranschaulichung der Problematik zunächst die Diskrepanz aufgezeigt, die zwischen der schulischen Informatik und den aktuellen Strömungen im IT-Berufsfeld herrscht. Es werden mögliche Gründe anhand von Bildungsstandards und einer Lehrerbefragung ermittelt. Lösungsansätze zur Überwindung der Diskrepanz werden daraufhin erarbeitet, indem Anleihen bei der SE gemacht und deren Methoden in einen schulischen Kontext gebracht werden. Die aus Projekterfahrungen entstandenen Tipps zur erfolgreichen Anwendung der Methoden werden zusätzlich vorgestellt.

Festzustellen bleibt jedoch, dass sich das vorherrschend negative Image der Informatik nicht nur durch die hier aufgezeigten Methoden allein bekämpfen lässt. Man muss sich darüber im Klaren sein, dass sehr viele Aspekte und gesellschaftliche Prozesse beteiligt sind. Die genannten Methoden sollen daher nur als ein Schritt in die richtige Richtung verstanden werden, in der Hoffnung, dass weitere Ansätze darauf aufbauend in der Praxis entwickelt werden.

2 Informatik an der Schule

Bei den folgenden Betrachtungen sei natürlich immer darauf verwiesen, dass sich diese an den verfügbaren Empfehlungen und Bildungsstandards orientieren. Engagierte Lehrer vermögen es bereits ohne die hier vorgeschlagenen Methoden, die formulierten Inhalte und sozialen Aspekte anschaulich zu vermitteln. Zahlreiche Beobachtungen und Gespräche sowohl mit Schülerinnen und Schülern als auch mit Lehrern im Rahmen von Projektveranstaltungen ergaben jedoch, dass es gerade in der Vermittlung von sozialen Aspekten und besonders der Gruppenerfahrungen der Informatik an Methoden und Ideen zu mangeln scheint. Dieser Eindruck wurde zudem durch eine Umfrage unter Informatiklehrern im Rahmen eines Workshops gefestigt (siehe dazu Kapitel 2.1).

Im Folgenden soll auf Publikationen zum Informatikunterricht an Schulen eingegangen werden. Besonderes Augenmerk wurde hierbei auf soziale Aspekte gelegt. In diesem Zusammenhang ist besonders darauf zu achten, ob und wie die geforderten sozialen Aspekte auch der SE entsprechend widerspiegelt werden können. Prinzipiell sei gesagt, dass

dieser Artikel in der Überzeugung verfasst wurde, dass bereits in Empfehlungen und Bildungsstandards Elemente enthalten sein sollten, die zur Steigerung der Attraktivität des Fachs beitragen. Ein Fokus auf die reine Zweckmäßigkeit und die großen Berufschancen dürfte nicht ausreichend sein, um IT-Berufe in ein besseres und zutreffenderes Licht zu stellen.

In den Grundsätzen und Bildungsstandards für die Informatik, die durch die Gesellschaft für Informatik e.V. (GI) herausgegeben wurden¹, finden sich einige Hinweise darauf, dass Informatik als ein Schulfach begriffen werden sollte, das besondere interdisziplinäre und soziale Aspekte aufweist [Pu08]. Hierbei fällt auf, dass besonders zu Anfang der Publikation darauf Wert gelegt wird, dass Lösungen für informatische Probleme sehr häufig im Team erarbeitet werden müssen. Daher wird verlangt, dass ein guter Informatikunterricht den Schülerinnen und Schülern das Selbstvertrauen dazu vermittelt. Ein weiterer Aspekt ist, dass es den Verfassern darauf ankommt, dass die informatische Schulausbildung vorbereitet auf *“das Leben, so wie es ist”*. Die Autoren siedeln ihre Empfehlungen bewusst zwischen Input- und Output-Orientierung an, trotzdem sind gerade Hinweise zu passenden Methoden zur Vermittlung von sozialen Aspekten schwer auffindbar. Es ist verwunderlich, dass immer wieder darauf hingewiesen wird, dass künftige Lebenssituationen (a.a.O., S.1, S.4, S.6) berücksichtigt werden sollen, aber nie darauf verwiesen wird, dass es durchaus Möglichkeiten gibt, sich der Elemente der SE zu bedienen, um dies realitätsnah zu gestalten. Auch wird darauf eingegangen, dass gemeinsames Erarbeiten von informatischem Wissen gemeinsam stattfinden soll. Begründet wird dies jedoch meist damit, dass so möglichst effektiv Informatikstoff vermittelt werden kann; auch hier finden sich keine Anhaltspunkte, dass der Bezug zur realen SE die Attraktivität des Fachs steigern könnte. Der Artikel stellt heraus, dass nur in der Informatik die Möglichkeit für Schülerinnen und Schüler besteht, zu erfahren, *“ob die konstruktive Arbeit mit technischen Werkzeugen für sie möglich und attraktiv, eben eine Lebensperspektive ist”* und somit auch die reale SE an Schulen erfahrbar sein muss. Die Interdisziplinarität der Informatik wird darüber hinaus erwähnt, jedoch nicht als Faktor der Informatik erkannt, der die Attraktivität steigern könnte; vielmehr wird nur darauf verwiesen, dass so fächerübergreifende Stoffe vermittelt werden können. Hier wäre eine offensivere Herangehensweise wünschenswert, die es vermag, gerade den nicht von Haus aus an der Informatik interessierten Schülerinnen und Schülern das Fach schmackhaft zu machen. Eine genauere Untersuchung nach sozialen Aspekten lohnt sich bei den folgenden identifizierten Prozessbereichen: *“Modellieren und Implementieren”*, *“Kommunizieren und Kooperieren”* und *“Darstellen und Interpretieren”*. Auffallend bei *“Modellieren und Implementieren”* ist, dass ein deutlicher Fokus auf Werkzeuge gelegt wird und dabei nicht erwähnt wird, dass gerade diese Arbeit nur noch sehr selten als Einzelleistung in der SE zu verstehen ist. Hier scheint die Möglichkeit vertan, den Schülerinnen und Schülern zu vermitteln, dass gerade das Erarbeiten dieser Lösungen einen hochgradig kreativen und zwischenmenschlichen Aspekt der IT-Berufsbilder darstellt. Gerade die Forderungen bei *“Kommunizieren und Kooperieren”* erscheinen als sehr treffend. Betrachtet man jedoch die Praxis, so scheint es einen Mangel an Methoden zu geben, die Lehrern an die Hand gegeben werden können (siehe hierzu nochmals 2.1). In den Betrachtungen zu *“Darstellen und Interpretieren”* werden leider nur selten Hinwei-

¹ Die GI spricht hierbei von einem Mindestmaß an informatischem Wissen, das spätestens mit der mittleren Reife erlangt werden soll und somit breiten Anklang im zukünftigen Gesellschaftsbild finden soll.

se gegeben, dass gerade diese Fähigkeiten eigentlich nur in Kleingruppen erlernt werden können, um so z.B. zu erfahren, wie Darstellungsformen von anderen Teilnehmern interpretiert werden.

In den Empfehlungen für ein Gesamtkonzept zur informatischen Bildung an allgemein bildenden Schulen, herausgegeben durch die GI, wird darüber hinaus auch die Vermittlung und Hervorhebung von Sozialkompetenz in den Aufgabenbereich der Informatikbildung gerückt. Es wird hierbei gezielt gefordert, dass Schülerinnen und Schüler befähigt werden sollen, *“Gruppenprozesse zu planen und mitzugestalten, Kritik entgegenzunehmen bzw. konstruktiv formulieren zu können, einen Arbeitsrollenwechsel zu erleben und akzeptieren zu können”* [Br00]. In diesem Text findet sich sogar der konkrete Hinweis, dass diese Fähigkeiten auch in der realen (IT-) Arbeitswelt wiederzufinden sind. Es wird jedoch nicht darauf verwiesen, dass diese Gegebenheit selbst auch ein zu vermittelndes Wissen ist.

In der Empfehlung der Kultusministerkonferenz (KMK) zur Stärkung der mathematisch-naturwissenschaftlich-technischen Bildung finden sich manche Forderungen, die in ihrer Interpretation und Durchführung zum Ansehen der Informatik und auch der Qualitätssicherung im Informatikunterricht beitragen können [Ku09]. Darin wird darauf verwiesen, dass möglichst *“konkrete Erfahrungen mit naturwissenschaftlichen, ingenieurwissenschaftlichen und technischen Berufen zu ermöglichen”* sind. Hierbei wird zwar in erster Linie darauf Wert gelegt, dass Netzwerke zu Firmen entstehen sollen, so dass sich Jugendliche vor Ort davon ein Bild machen können. Im Umkehrschluss ist jedoch auch denkbar, dass eben konkrete Methoden aus dem Berufsbild in den Schulkontext überführt werden sollten. Weiteren Anreiz bietet auch das in der Empfehlung der KMK beschriebene Handlungsfeld *“Gesellschaftliche Akzeptanz”*, in dem die Rede davon ist, dass das Interesse an technischen Fragestellungen bei Schülerinnen und Schülern zu wecken sei. Hier ist es natürlich verwunderlich, dass nicht auch darauf hingewiesen wird, welche kreativen, sozialen und interdisziplinären Berufsbilder in diesem Bereich vorhanden sind.

Betrachtet man die oben genannten Empfehlungen und Hinweise, so liegt es nahe zu behaupten, dass es mangels konkreter Methoden wohl sehr schwierig für Lehrer ist, diese Inhalte angemessen zu vermitteln und sie somit nicht ihren Weg in die Schulpraxis finden. Um diesen Eindruck zu überprüfen, wird hier eine Lehrerbefragung herangezogen.

2.1 Lehrerbefragung

In einem Fragebogen wurden, abzielend auf die oben beschriebenen sozialen Aspekte im Informatikunterricht, 17 Informatiklehrer aus Hamburg und Schleswig-Holstein befragt. 13 Teilnehmer waren Gymnasial-, zwei Real- und einer Gesamtschullehrer, vier davon waren weiblich. Die Studie wurde von uns im Anschluss an einen Workshop durchgeführt, die Teilnahme war freiwillig.

Auf die Frage, in welchen Veranstaltungsformen Informatikinhalte durch die befragte Person vermittelt wurden, ergab sich folgende Verteilung (Mehrfachnennungen waren erlaubt): drei als Pflichtfach, zwölf als Wahlpflicht, zwei Wahlfach, sechs AG und nur eine Projektwoche. Diese Antworten sind bemerkenswert, da die Mehrzahl in Wahlpflicht und Wahlfach liegt. Naheliegender ist, dass zu diesem Zeitpunkt bereits eine Vorauswahl der Schülerinnen und Schülern stattgefunden hat. So ist davon auszugehen, dass nur noch eine gewisse, ohnehin schon informatikaffine Klientel teilnimmt. Wünschenswert wäre eine

Vielzahl von Projektangeboten, da dort meistens ein breiteres Spektrum an Schülerinnen und Schüler beteiligt ist und somit auch die Chance besteht, andere Zielgruppen für die Informatik zu begeistern. Darüber hinaus dürften gerade Projektwochen und AGs besonders gut den Projektcharakter der SE vermitteln dürfen.

Bei der Frage, welche Inhalte vorwiegend in den Lerneinheiten programmiert wurden (Mehrfachnennungen waren auch hier erlaubt), gaben sieben Teilnehmer an, dass Anwendungen entwickelt wurden, sechs bezeichneten Algorithmen als Hauptfokus und niemand identifizierte Games als Lerninhalt. Darüber hinaus gab es noch andere Anwendungen die Verwendung fanden und extra genannt wurden: PovRay, Robotik / Lego Mindstorms, Robot Karol und Microsoft Office. Positiv bleibt festzuhalten, dass es einen Fokus auf Anwendungen gab (die ja auch Algorithmen beinhalten können). So ist wenigstens davon auszugehen, dass prinzipiell auf Fragen der SE Bezug genommen werden kann. Unter dem Aspekt der Kreativität in der Informatik ist es jedoch enttäuschend, dass niemand die gute Gelegenheit nutzt und mit Games das aufgreift, was die meisten Jugendliche begeistern dürfte und darüber hinaus einen guten Anknüpfungspunkt an aktuelle Entwicklungen in der SE darstellt.

In der Befragung zu verwendeten Web-Plattformen zur Unterstützung der Arbeit der Schülerinnen und Schüler, ergab sich folgendes Bild: sechs boten das Schul-CommSy an, sechs lo-net², fünf gaben andere, nicht weiter spezifizierte an. Erfreulich ist hier ganz klar, dass dementsprechend alle Teilnehmer solche webbasierte Unterstützungen anbieten, die natürlich (wenn auch wohl in leicht anderer Form) Einsatz in der SE finden.

Aus der offenen Frage, ob Gruppenarbeit stattfindet und wenn ja, wie diese gestaltet sei, ergibt sich, dass Gruppenarbeit von nahezu allen Lehrern umgesetzt wurde (im Schnitt mit einer Gruppenstärke von zwei bis drei Teilnehmern), es jedoch häufig an Methoden mangelt, um diese zur Zufriedenheit der Lehrer und der Schülerinnen und Schüler durchzuführen. Es wurde geäußert, dass es gerade an der Aufgabenverteilung unter den Teilnehmern hapere, da es schwerfällt, dort das richtige Herangehen zu vermitteln. Darüber hinaus entsteht aus den Antworten der Eindruck, dass manche Gruppenarbeit trotzdem bedeutet, dass jeder Teilnehmer Aufgaben an einem eigenen PC bearbeitet. Dies zeigt insgesamt ganz klar, dass hier ein Verbesserungsbedarf besteht, der die sinnvolle Gruppenarbeit in geordnete Bahnen bringt. Auch hier liegt es nahe, sich der Projektmanagementmethoden der SE zu bedienen und dies auch unter diesem Aspekt den Schülerinnen und Schülern zu vermitteln.

Bei den Antworten zu der Frage, ob Dokumentationen angefertigt werden müssen und zu welchem Zeitpunkt dies geschehe, ist herauszulesen, dass diese eher zum Ende hin angefertigt werden müssen und diese Aufgabe von sehr wenigen Schülerinnen und Schülern geschätzt wird. Somit wird deutlich, dass attraktivere Herangehensweisen vorgestellt werden müssen, um Programmiererergebnisse im schulischen Kontext bewerten zu können. Hier gibt die SE keine offensichtliche Antwort, da es dort nicht um Einzelleistungen, sondern um ein fertiges und funktionierendes Produkt geht. Trotzdem soll versucht werden, Artefakte aus der SE, die parallel zur Programmierung entstehen, zumindest in Teilen zur Dokumentation und Präsentation zu verwenden (siehe Kapitel 4.3).

Abschließend kann man zusammenfassen, dass es gerade für die oben beschriebenen sozialen Aspekte und den Realitätsbezug an einfachen Methoden zu mangeln scheint, die Lehrern mit auf den Weg gegeben werden können, um Gruppenarbeit zu vereinfachen und strukturiert verlaufen zu lassen. Darüber hinaus scheint es jedoch wichtig, dass der spannende und kreative Entwicklungsprozess im Vordergrund steht und störende Nebenaufgaben, wie z.B. Abschlussdokumentationen interessanter gestaltet werden bzw. entsprechend der SE während des eigentlichen Projekts nebenläufig entstehen. Wie bereits erwähnt, bieten agile Methoden Techniken an, um diese Probleme zu bewältigen.

3 Agile Methoden in der professionellen Softwareentwicklung

Agile Methoden folgen einem Manifest, das 2001 von Kent Beck et al. verfasst wurde und bereits vorhandene populäre Herangehensweisen und Denkweisen der SE zusammenfassen sollte, indem es ein besonderes Augenmerk auf Werte wie z.B. Kommunikation und Feedback innerhalb der Gruppe legt [Be01]. Es entstand aus dem Wissen, dass große Projekte vormals immer darunter litten, dass Auftraggeber und Entwickler schnell unvereinbare Sichten auf das Produkt entwickeln, wenn die SE nach klassischen Modellen, wie z.B. dem Wasserfallmodell aufgebaut ist. Ein Projektmanagement, das dem Manifest folgt, war demzufolge gefragt. Den Autoren zufolge sind Softwareentwicklungsprozesse durch agile Methoden mehr auf sozialen Austausch fokussiert, und die entstehenden Produkte sind hierdurch besser an die Bedürfnisse der Nutzer angepasst. Wobei zu sagen ist, dass es den Autoren neben sozialeren Arbeitsstrukturen auch bzw. wohl in erster Linie um eine effektivere Beziehung zwischen Auftraggeber und Entwickler geht.

Bis zum heutigen Tage haben sich zwei Varianten der Agilen Methoden besonders etabliert: Extreme Programming (XP) nach Beck [Be00] und Scrum, das aus mehreren Arbeiten heraus entstanden ist, wobei wohl drei Publikationen den größten Anteil daran zu haben scheinen [TN86, Be99, SB01]. Beide Varianten versuchen möglichst allumfassend, den gesamten Prozess der SE abzudecken und passende Methoden anzubieten. Die folgenden Beschreibungen von XP und Scrum sind lediglich auf soziale Strukturen bzw. Methoden fokussiert, um den ursprünglichen Punkt dieses Papiers nicht aus den Augen zu verlieren.

4 Agile Methoden an Schulen

Programmiertechnische Vorgehensweisen der agilen Methoden wurden in Schulkontexten bereits erfolgreich eingesetzt [We05]. Im Gegensatz dazu betrachtet der vorliegende Artikel die sozialen Aspekte der Arbeitsprozesse der agilen Methoden. Im Folgenden werden entsprechende Methoden aus XP und Scrum vorgestellt, die bereits in mehreren Projekten an Schulen und ähnlichen Einrichtungen erfolgreich verwendet wurden. Es handelt sich um Projekte, bei denen die grafische Programmierumgebung Scratch² oder Greenfoot³ verwendet wurde, um interkulturelle Geschichten am Computer zu erzählen [Gö09a]. For-

² <http://scratch.mit.edu/>, zuletzt besucht am 14.04.2011

³ <http://www.greenfoot.org/>, zuletzt besucht am 14.04.2011

schungsziel der Projekte war es, aus dem Verhalten der Schülerinnen und Schüler Werkzeuge zu entwickeln, um die dynamischen und zwischenmenschlichen Aspekte der Programmierung erfahrbar zu machen und zu unterstützen. Erste daraus entstandene prototypische Werkzeuge liegen als Ergänzung von Scratch um kollaborative Funktionalitäten vor und sind Gegenstand eines anderen Artikels [Gö09b].

Die Methoden werden nachfolgend aus der Sicht der SE beschrieben und dann gegebenenfalls mit Hinweisen versehen, wie die jeweilige Methode in einem schulischen Kontext anzuwenden ist, und welche Probleme auftreten können und wie diesem entgegengewirkt werden kann.

Es sei empfohlen, diese Methoden in einer Auftaktveranstaltung vorzustellen und den klaren Bezug zu den realen agilen Methoden herzustellen. Besonders geeignet zur Steigerung der Attraktivität der Methoden scheint der Verweis auf die Grundwerte der agilen Methoden, die auch in einem schulischen Kontext von großer Bedeutung sein dürften: Kommunikation, Respekt, Mut, Einfachheit und Feedback [Be00].

4.1 Pair programming

Die bekannteste Methode von XP dürfte das *pair programming* sein. Die Idee dabei ist, dass jeweils vor einem Rechner zwei Entwickler gemeinsam programmieren. In der Praxis bedeutet dies, dass eine Person die Tastatur verwendet und somit den Quellcode schreibt, wobei die zweite Person die Arbeit hinterfragt, auf Fehler prüft und alternative Lösungsvorschläge macht. Es ist wichtig, dass die Person mit Tastatur die eigenen Gedanken bei der Arbeit mündlich mitteilt, um der zweiten Person das Erkennen von möglichen Denk- und Strukturfehlern zu erleichtern. Idealerweise beginnt die Person mit Tastatur erst zu schreiben, wenn sich das Paar einig über die Lösung ist. In der Regel werden die Tastatur und damit die Rollen mehrmals untereinander im Laufe des Arbeitstages getauscht.

Im schulischen Kontext ist darauf zu achten, dass die Rolle der Person ohne Tastatur ernsthaft eingenommen wird und dass es zu dem Wechsel der Tastatur wirklich kommt. Hier ist eine zeitliche Taktung sehr hilfreich. Darüber hinaus ist bei Paaren mit stark unterschiedlichem Wissenstand darauf zu achten, dass die Person mit einem geringeren Kenntnisstand die Tastatur häufiger verwendet und der Partner sich darin übt, sein Wissen zu vermitteln. Hierbei ist besonders acht zu geben, dass nicht einfach die Anweisungen diktiert werden, sondern dass Konzepte erklärt werden. Notfalls hilft es hier Paare zu tauschen.

In mehreren Projekten konnte beobachtet werden, dass gerade Schülerinnen das Programmieren im Paar sehr schnell annehmen und häufig davon begeistert sind, dass gemeinsam mit einem Partner Lösungen erarbeitet werden. Dies ist wohl einer von vielen Schritten, um weibliche Jugendliche für IT-Berufe zu begeistern. Studien mit weiblichen Studierenden lassen zumindest die Hoffnung zu, dass solche sozialen Aspekte der SE das Selbstbewusstsein bezüglich der eigenen Informatikfähigkeiten stärken, das benötigt wird, um sich einen IT-Beruf zuzutrauen [Be04].

4.2 User Stories

Es handelt sich bei *user stories* um die kleinteilige und prägnante Beschreibung von elementaren Funktionalitäten, die einem Endnutzer mit dem fertigen Softwareprodukt zur Verfügung stehen sollen. Zu finden sind *user stories* sowohl bei XP als auch bei Scrum.

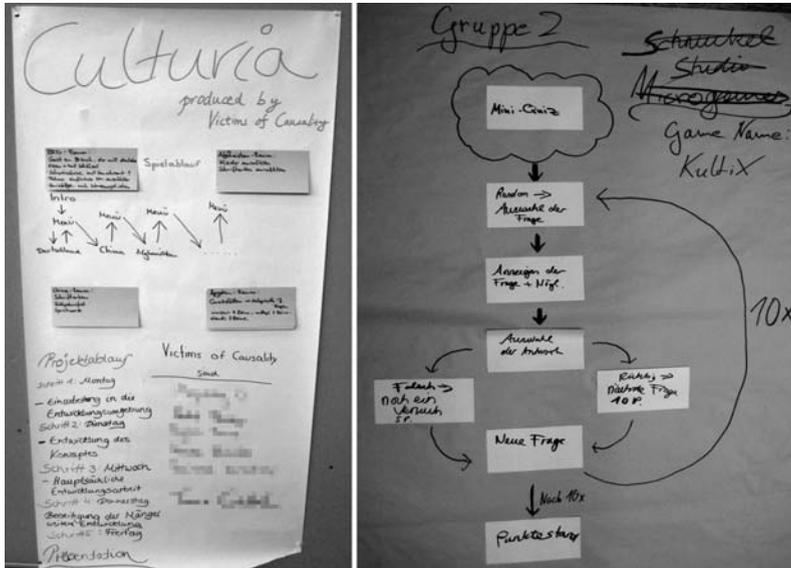


Abbildung 1: Die Poster des *informative workspace* können als Diskussionsgrundlage, zur Präsentation und zur Dokumentation verwendet werden (Namen nachträglich anonymisiert)

Ziel ist es, diese elementaren Anforderungen nicht aus dem Auge zu verlieren, wobei die *user stories* gleichzeitig zur Organisation der Aufgaben innerhalb des Entwicklungsteams dienen. Häufig werden einzelne *user stories* dann von Programmierpaaren übernommen und fertiggestellt. Erst nach Fertigstellung widmet sich ein Paar neuen *user stories*.

Im schulischen Kontext handelt es sich wohl um eine Methode, die gerade zu Anfang eine Hilfestellung benötigt, da Schülerinnen und Schüler häufig dazu tendieren, möglichst alle Funktionalitäten in einer Beschreibung abdecken zu wollen. Elementare Aufgaben erscheinen ihnen schnell als zu banal. Daher empfiehlt es sich für den Anfang, Beispiele vorzugeben und gemeinsam mit den Teilnehmern zu erarbeiten, dass diese kleinteilige Herangehensweise die Gruppenarbeit erleichtert, da nur so Aufgaben verteilt werden können und niemand befürchten muss, dass immer nur ein anderer Teilnehmer die spannenden und fordernden Aufgaben vermittelt bekommt, wohingegen man selbst entweder zu banale oder aber unlösbare Aufgaben zugeteilt bekommt.

4.3 Informative workspace

Bei XP wird empfohlen, dass sämtliche Prozesse und Lösungswege auf Postern skizziert werden. Diese Poster sollen in einem allgemein zugänglichen Raum zur Verfügung stehen, so dass sich Teilnehmer jederzeit über den aktuellen Stand der Entwicklung informieren können und darüber hinaus eine gemeinsame Diskussionsgrundlagen haben. Den meisten Schülerinnen und Schülern ist auf Anhieb nicht ersichtlich, wozu sie solche Poster anfertigen sollen. Es empfiehlt sich daher, ihnen klar zu machen, dass solche Poster die Arbeitswege darstellen und somit sehr gut verwendet werden können für mögliche Präsentationen und Abschlussdokumentationen (siehe beispielsweise Abb. 1). Es ist empfehlenswert, für

die Veranstaltungsleiter von Zeit zu Zeit auf die Poster einzugehen, gezielt Fragen zu stellen, ob diese noch aktuell sind, und Diskussionen anhand der Poster zu führen.

4.4 Meetings

Eine Methode, die man bei XP und in Scrum findet, ist ein tägliches Treffen zu Beginn des Arbeitstages (XP: *stand up meeting*, Scrum: *daily scrum meeting*), bei dem alle Projektmitglieder kurz der Gruppe mitteilen sollen, was am vergangenen Tag geschafft wurde, wo Probleme lagen und welche Entwicklungen für den aktuellen Tag vorgesehen sind.

Hier muss gerade zu Beginn stark darauf geachtet werden, dass jede Person individuelle Erfahrungen mitteilt. In der Klassensituation entsteht häufig eine Dynamik, in der auf den Vorredner verwiesen wird. Dies ist möglichst zu unterbinden, in dem man die Wichtigkeit deutlich macht, dass die Gruppe einen Eindruck erlangen soll, das jeder für das gemeinsame Ziel arbeitet.

5 Zusammenfassung und Ausblick

Die hier genannten Techniken der agilen Methoden sollen es durch ihren klaren Berufs- und Realitätsbezug erlauben, Schülerinnen und Schülern zu verdeutlichen, welche Chancen IT-Berufsbilder darstellen. Darüber hinaus sollen diese Methoden aber auch Gruppenarbeit im Informatikunterricht unterstützen. Gerade im Bereich der Gruppenunterstützung mangelte es bislang an leicht anzuwendenden Methoden, die sich für Lehrer, Schülerinnen und Schüler als attraktiv darstellen und auch einfach anzuwenden sind. Sicherlich sind diese Methoden nicht in allen Unterrichtssituationen einsetzbar, da dort häufig auch andere Kriterien von Belang sind. Wünschenswert wäre es, dass während der Unterrichtsvorbereitung immer wieder hinterfragt wird, ob der Einsatz der genannten Methoden möglich ist. In der Regel ergibt sich daraus ein Mehrwert durch die dynamisch gestaltete Gruppenarbeit bei nur sehr geringen Einsatzhürden.

Anzuraten ist, dass diese Methoden mit dem klaren Bezug zur SE vorgestellt werden und zu Anfang aktiv unterstützt werden. Es bleibt zu sagen, dass die Einführung dieser Methoden immer auch mit Widerwillen (gerade unter den männlichen fortgeschritteneren Teilnehmern) verbunden sein können; hier ist es natürlich angeraten, die Vorteile der Methoden weiterhin glaubhaft und überzeugend zu vermitteln. Ganz im Sinne des agilen Manifestos (“interactions over processes and tools”) ist es jedoch wichtig, dass diese Methoden bei Bedarf gemeinsam mit den Teilnehmern angepasst werden können. Mischformen, die sich an speziellen Bedürfnissen orientieren erscheinen sinnvoll. Sie vermögen es, dynamische Prozesse, wie sie in der SE stattfinden, abzubilden und sind so ein weiterer Faktor, um dem Berufsbild Informatik zu einem passenderen Image zu verhelfen.

Es kann nur angeraten werden, noch weitere Methoden zu adaptieren und wenn möglich immer ein Ohr an den aktuellen Entwicklungen der SE zu haben. Nur so erscheint es möglich, die starke Diskrepanz zwischen Realität und Ansehen der Informatik wenigstens in Teilen abzubauen. Darüber hinaus müssen weitere Schritte unternommen werden, will man langfristig das gesellschaftliche Ansehen der Informatik verbessern. Ein Ansatz ist, gezielt auf die vielen verschiedenen Aspekte der Informatik einzugehen (siehe

dazu [BCM06]) und diese früh und altersgerecht an Schulen zu vermitteln. Gerade die Chancen, die durch die enorme Interdisziplinarität der Informatik entstehen, scheinen nach wie vor nicht bei Schülerinnen und Schülern angekommen zu sein.

Die Methoden haben in den bisher durchgeführten Projekten mit Schülerinnen und Schülern dazu beigetragen, dass die Teilnehmer mit Spaß an klar geregelten Aufgaben in der Gruppe arbeiten konnten. Auf diesem Weg konnten sie für sich feststellen, dass es sich bei der Informatik um eine soziale interaktive Disziplin handelt.

Literaturverzeichnis

- [Be01] Beck, K. et al.: <http://agilemanifesto.org/> (zuletzt besucht am 14.04.2011), 2001.
- [BCM06] Biundo, S.; Claus, V.; Mayr, H. C., Hrsg: *Was ist Informatik? Unser Positionspapier*. Gesellschaft für Informatik e.V., 2006.
- [Be99] Beedle, M. et al.: *Scrum: A Pattern Language for Hyperproductive Software Development*, Seiten 637–651. Addison Wesley, 1999.
- [Be00] Beck, K.: *Extreme programming explained: embrace change*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2000.
- [Br00] Breier, N. et al.: *Empfehlungen für ein Gesamtkonzept zur informatischen Bildung an allgemein bildenden Schulen*. Gesellschaft für Informatik e.V. (GI), 2000.
- [Be04] Berenson, S. B. et al.: Voices of women in a software engineering course: reflections on collaboration. *Journal on Educational Resources in Computing*, 4(1):3, 2004.
- [Gö09a] Göttel, T.: *Culturia: Intercultural Learning by Designing Games*. The Inter-Disciplinary Press, 2009.
- [Gö09b] Göttel, T.: Virtual sandbox: adding groupware abilities to Scratch. In *Proceedings of the 8th International Conference on Interaction Design and Children*, Seiten 158–161, Como, Italy, 2009. ACM New York, NY, USA.
- [Ga09] García-Crespo, Á. et al.: IT Professionals' Competences: High School Students' Views. *Journal of Information Technology Education*, 8:45–57, 2009.
- [Ku09] Kultusministerkonferenz (KMK), Hrsg. *Empfehlung der Kultusministerkonferenz zur Stärkung der mathematisch-naturwissenschaftlich-technischen Bildung*. 2009.
- [Pu08] Puhlmann, H. et al.: *Grundsätze und Standards für die Informatik in der Schule*. Gesellschaft für Informatik e.V. (GI), 2008.
- [Pr01] Prensky, M.: Digital Natives, Digital Immigrants. *On the Horizon*, 9(5), 2001.
- [Ro08a] Romeike, R.: Towards students' motivation and interest: teaching tips for applying creativity. In *Proceedings of the 8th International Conference on Computing Education Research*, Seiten 113–114, Koli, Finland, 2008. ACM.
- [Ro08b] Romeike, R.: Sichtweisen einer Kreativen Informatik. In Torsten Brinda, Michael Fothe, Peter Hubwieser und Kirsten Schlüter, Hrsg., *Didaktik der Informatik - Aktuelle Forschungsergebnisse*, Seiten 129–138. Gesellschaft für Informatik e.V. (GI), 2008.
- [SB01] Schwaber, K.; Beedle, M.: *Agile Software Development with Scrum*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 1st. Auflage, 2001.
- [TN86] Takeuchi, H.; Nonaka, I.: New New Product Development Game. *Harvard Business Review*, 1986.
- [We05] Weigend, M.: Extreme Programming im Klassenraum, INFOS 2005 Dresden. Erweiterte Version. http://www.fernuni-hagen.de/schulinformatik/xp_weigend.pdf (zuletzt besucht am 14.04.2011), 2005