

Entwicklungsprojekt

Unterricht im Tutoren-Konzept "Chief-Programmer"

Kompetenznachweis Modul 9 'Berufsfelddidaktik', 2019/2020

Modul 9 & 10, Semester 2 & 4

Eingereicht am: 11.02.2019

Eingereicht von: Harald Müller

Eingereicht bei: Martin Vonlanthen

Stichworte

Selbstorganisiertes Lernen, Schülerzentrierter Unterricht, Handlungsorientierter Unterricht, Projektunterricht, Lerninseln mit Tutoren, Schüler helfen Schülern, Schüler unterrichten Schüler, Schüler als Tutoren, Schüler als Lehrer, Tutoren als Multiplikatoren.

Abstrakt, Zusammenfassung

Dieses hier vorgestellte Tutoren- oder Chief-Programmer-Konzept ist keine Alltags-Methode für Unterricht. Es ist geknüpft an ganz spezielle Konditionen. Aber wenn diese erfüllt sind, ist das eine sehr interessante Unterrichtsmethode, die für die ganze Klasse sehr hohe Motivation und eine sehr hohe Schülereinbindung bringt und dabei eine grosse netto Lernzeit resultiert.

Inhaltsverzeichnis

1.	Einleitung.....	3
1.1	Bedingungen als Kompetenznachweis.....	3
1.2	Übersicht über den Kompetenznachweis	3
2.	Erfahrungsbericht.....	3
2.1	Vorgeschichte.....	4
2.2	Ausgangslage.....	4
2.3	Entwicklung der Idee.....	4
2.4	Blick in die Literatur und theoretischer Hintergrund.....	5
2.5	Organisation.....	5
2.6	Umsetzung.....	6
2.7	Zusammensetzung der Modulnote	6
2.8	Modul-Fazit.....	7
3.	Konzeptvorlage für eine weitere Durchführung	7
3.1	Vorbedingungen.....	7
3.2	Teameinteilung.....	8
3.3	Nichtklassisches Verhältnis und Rollen von Lehrpersonen und Lernenden	8
3.4	Durchführung	8
3.5	Ergänzende Aufträge für Tutoren.....	9
3.6	Tutoren-Anreiz-System.....	9
4.	Fazit	9
5.	Quellenverzeichnis	10
6.	Abschlussbemerkung.....	10
7.	Anhang.....	11

1. Einleitung

Im Zuge der Ausbildung am EHB als Berufskunde-Lehrperson habe ich im Januar 2019 mit dem Modul **Berufsfelddidaktik** begonnen wo als Kompetenznachweis ein Berufsfeld-didaktisches Entwicklungsprojekt zu machen ist. Per Ende Januar 2019 ging bei uns an der Technischen Berufsschule Zürich TBZ das Semester zu Ende und gleichzeitig kam die Idee und Erkenntnis auf, dass dieses soeben abgehaltene Unterrichtskonzept derart exotisch ist, dass sich dies als Entwicklungsprojekt im Sinne des Moduls 9 in der Berufsfelddidaktik eignet.

1.1 Bedingungen als Kompetenznachweis

Mehrere Bedingungen sind erfüllt. Das Konzept wurde an unserer Schule noch nicht angewandt, es ist neu und **viele meiner Lehrerkollegen** sind am Ausgang dieses Konzeptes interessiert. Daneben sind auch mehrere **Dozenten** am EHB **an den Resultaten interessiert** und es ist ein Aspekt meiner Ausbildung, dass es sich hier lohnt, dieses System erstens zu dokumentieren und zweitens entsprechende Empfehlungen oder Weiterentwicklungsmöglichkeiten am Schluss anzudeuten.

1.2 Übersicht über den Kompetenznachweis

Im ersten Teil wird gezeigt, wie ich den soeben absolvierten Unterricht aufgezogen habe und wie sich diese Idee entwickelt hat. Dies wird im **Erfahrungsbericht** mit der Beschreibung, wie es zu dieser Idee kam und wie dieses Konzept angewandt und aufgesetzt wurde und endet mit dem Fazit der abgehaltenen Lehrmethode. Zum Erfahrungsbericht gehört auch die im Anhang befindliche Klassenevaluation und -feedbackantworten.

Im Weiteren Kapitel der **Konzeptvorlage** werde ich dann Bedingungen und Vorschläge geben für all jene, die dieses Konzept auch mal ausprobieren wollen.

2. Erfahrungsbericht

Für nicht eingeweihte der Informatik sei hier vorweggenommen, dass aus der ICT-Berufsbildung her, der Fachunterricht an Berufsschulen für alle Informatiker Berufe (Applikationsentwicklung, Systemtechnik, Betriebsinformatik und ICT-Fachleute) in Modulen zu 40 Lektionen aufgegliedert ist. Ein IT-Modul wird an der Technischen Berufsschule Zürich normalerweise in 10 Wochen zu 4 Lektionen abgehalten. In anderen Berufsschulen, wie z.B. in der Berufsfachschule Uster werden die Module in 20 Wochen zu 2 Lektionen gehalten.

Der Inhalt aller Informatik-Module sind wohl definiert und auch für die Durchführung gibt es in Form von mehreren Leistungsbeurteilungsvorgaben (LBV) eine Vorgabe, auf welche Art ein Modul abgehalten werden kann, oder gar muss. Glücklicherweise haben wir in der Technischen Berufsschule Zürich recht freie Hand, wie der Kurs, das Modul abzuhalten ist.

Hier die Links zu den **Moduldefinitionen** und **Leistungsbeurteilungsvorgaben**:

<https://cf.ict-berufsbildung.ch/modules.php?name=Mbk&a=20101&cmodnr=226a&noheader=1>
M226a: Klassenbasiert (ohne Vererbung) implementieren (M226a, 2019)

<https://cf.ict-berufsbildung.ch/modules.php?name=Mbk&a=20101&cmodnr=226b&noheader=1>
M226b: Objektorientiert (mit Vererbung) implementieren (M226b, 2019)

2.1 Vorgeschichte

Im Vorgängermodul M226a konnte ich schon ganz am Anfang beobachten, dass bei den selbst zu lesenden Theorieabschnitte des Lehrmittels, aber auch nachher bei den Aufgabenstellungen, sich eine Schülerin intensiv mit einem der Schüler mit Asperger-Syndrom **erklärend beschäftigte**. Als ich sie in der Pause drauf angesprochen habe, meinte sie, dass sie das schon im ersten Lehrjahr mit ihm gemacht hatte und dass sie sowas gerne mache. Ich habe ihr dann gesagt, dass das zwar schön sei, aber ich meinerseits sie darauf auf aufmerksam machen müsse, dass wir aufzupassen haben, dass sie sich damit nicht überfordere und ich die Aufgabe habe, sie zu schützen. Grade auch, wenn sie auch ihre Arbeiten und Aufgaben noch in entsprechender Zeit zu machen habe.

In diesem Modul M226a konnte ich auch sehen, dass ein paar wenige bedeutend weiter in dem Thema waren als die anderen. **Vier bis sechs Lernende** arbeiten schon seit Lehrbeginn vor einem Jahr mit Java. Zwei davon auch schon länger und **kennen die objektorientierten Konzepte aus der täglichen Anwendung**. Die anderen der Klasse hatten mit Java noch keine oder nur wenig Erfahrung, weil sie im Lehrbetrieb mit anderen Programmiersprachen arbeiten.

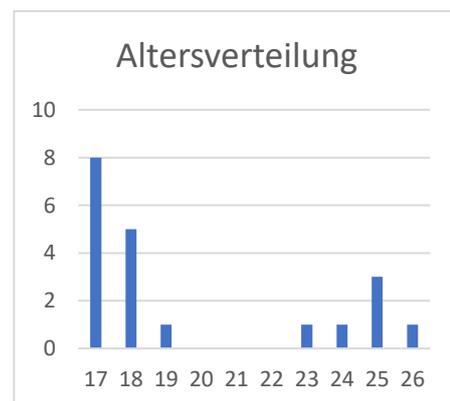
2.2 Ausgangslage

Die Klasse bestand aus **24 Applikationsentwickler** und sind eben im **2. Lehrjahr** gekommen. Sie sind sehr heterogen zusammengestellt. Die Altersstruktur ist von **17 bis 25 Jahren**. Die Vorkenntnisse unter den Lernenden sind sehr unterschiedlich in Bezug auf die Java-Programmierung. Es gibt **Quereinsteiger**, die eben erst **mit der Lehre begonnen** und schon einen Beruf (Maurer), bzw. vor zwei Jahren eine Berufsmatur im Pflegebereich abgeschlossen haben. Diese beiden Lernenden sind blutige Anfänger in der Softwareentwicklung und machen eine abgekürzte Lehre, was sich derart äussert, dass die beiden das erste und zweite Lehrjahr schulisch zusammen absolvieren.

Ein Teil der Lernenden hat ein **Basislehrjahr** absolviert. Waren also noch nicht in einem Lehrbetrieb, sondern in einer fachschulischen Organisation. Andere hingegen, arbeiten seit Lehrbeginn in einem Lehrbetrieb und sind schon in Teams in produktiven Kundenprojekten eingebunden und arbeiten an grossen Applikationen mit.

Weiter waren in der Klasse **6 Asperger**, die in einem entsprechend geschützten Lehrbetrieb arbeiten und keinen Kunden- oder Projektkontakt haben, dafür aber während den Nichtschultagen auch an den schulischen Projekten arbeiten können, wenn sie das wollen.

Als dann das Modul M226b startete, gab es einen Ausfall eines Asperger-Lernenden (Lehrabbruch) und ich startete mit 23 Schüler. Also 5 Asperger, 6 Schüler über 23 Jahren und 17 unter 19 Jahren.



Im Laufe des Quartals mit den 10 Wochen sind aus Krankheitsgründen 2 Asperger ausgefallen.

2.3 Entwicklung der Idee

Gegen Ende des Moduls M226a beginnt man als Lehrperson natürlich auf das nächste Quartal zu schauen und in diesem Zug dann auch das Folgemodul, hier das M226b konzeptionell zu überlegen.

In den verschiedenen existierenden Leistungsbeurteilungsvorgaben ist demnach auch eine vertiefende Projektarbeit vorgesehen.

Die LBV, die wir in der Schule anwenden, sagt mir, dass die Module in SOL-Stil mit einem Kompetenzraster abgehalten werden kann. In der Methodik bin ich an der TBZ relativ frei und kann die für mich und die Lernenden optimale Didaktik und Methodik wählen oder anpassen.

Eine tolle Idee

Als die erste Idee geboren war, dass die besten Schüler andere anleiten könnten, habe ich Rücksprechen getroffen. Einerseits mit meinen Lehrerkollegen im Schulhaus, die ähnliche Module unterrichten. Aber ich habe auch Absprache und Beratung meines Dozenten am EHB herbeigezogen. Zuerst meinte er, dass es problematisch sei, Schüler andere Schüler anzuleiten währenddem sie selbst noch schwierige Aufgaben zeitgerecht abzuliefern haben. Daraufhin habe ich das Konzept noch etwas verfeinert und dann fand es auch mein Dozent am EHB eine tolle Idee, die verfolgt werden kann.

Das Lern-Setting sind 5 "Lerninseln" (Tische) mit 4 bis 5 Lernenden. Einer davon ist ein Tutor, der die anderen anleitet und zur Seite stehen kann. Jeder Lerninsel stand also ein Tutor als sog. "Chief-Programmer" (so unser Name) vor.

Wenn es stärkere Schüler gibt, dann sind sie Tutor-Kandidaten. Diesen Tutoren wird eine gewisse Verantwortung übertragen, dass sie mit bestem Wissen und Gewissen den anderen Schülern zur Hilfe stehen. Andererseits haben die anderen Schüler auch eine Verantwortung ihr Tutor entsprechend respektvoll zu behandeln.

2.4 Blick in die Literatur und theoretischer Hintergrund

Dass Schüler anderen Schülern helfen ist sicherlich nicht neu. So zitiert (Hanel, 1991) in deinem Aufsatz über «Lernen durch Lehren» dass schon Hugo Gaudig, 1922 (ohne Quellenangabe) fordert, dass Zitat: "den Schüler aus dem Passivum in das Aktivum zu übersetzen" sei. Und weiter zitiert er ohne Jahres- und Quellenangabe einen Peter Petersens (1884-1952), der gar das völlige Zurücktreten der Lehrperson fordert und nur dann in Erscheinung tritt, wenn ein Schüler ihn fragt.

Weiter zurück findet gar (Feldmann, 2002) S. 4 eine unreferenzierte Quelle von Bell und Lancaster, die schon 1796 ein «Schüler helfen Schüler»-Konzept beschrieben haben sollen.

Bei beiden Autoren kommen die folgenden Stichworte vor:

Schüler kann Erklären – Lernen durch Lehren - Erklären können - Schüler helfen Schülern – Schüler unterrichten Schüler – Schüler als Tutoren – Schüler als Lehrer.

In der Hochschuldidaktik von Hamburg, Osnabrück, Paderborn, München und weiteren wird von Einsatz von **Tutoren als Multiplikatoren** gesprochen. Einerseits sind die Tutoren von der Altersstruktur, aber auch von der sprachlichen und emotionalen Konstellation her näher am Lernenden, als es die Lehrperson sein kann. Siehe dazu

<https://www.google.ch/search?q=Tutoren+als+Multiplikatoren&ie=&oe=> (mehrere Stellen, 4.2.2019)

2.5 Organisation

Die Organisation ist also so aufgezogen, dass immer zwei Tische als Lerninseln zusammengestellt sind. Tutor und die anderen Lernenden setzen sich rund herum und bilden so eine Lerngruppe.

Im Laufe der Projektarbeit, also ab dem 2. Modul-Halbtage, fand pro Team unter Leitung der Lehrperson regelmässig für den Informationsaustausch ein **5-minütiges Standup-Meeting** statt, so wie es auch in vielen Projekten in der Praxis durchgeführt wird. An diesem Meeting kamen die folgenden drei Fragen für jede Person zur Sprache:

- Was habe ich bisher gemacht
- Was wird als nächstes drankommen
- Wo hatte ich Probleme

2.6 Umsetzung

Im Vormodul M226a hatten wir in Theorie das Lehrmittel (Ruggerio, 2012) bis Kap. 14 behandelt. Analog einer Aufgabe im Kap. 15, wo es um Flugzeuge und Gates geht, war nun eine Einzelaufgabe namens "Flixbus" zu realisieren. Dabei konnte jeder Lernende seinen eigenen Aspekt der "Flixbus"-Aufgabe umsetzen. Die meisten haben das Online-Buchungssystem / Buchungs-App gewählt.

Das Design, die Detailideen und die UML Diagramme (Hierarchie- und Klassendiagramme, Anwendungsfälle und Sequenzdiagramme) wurden pro Team an den Lerninseln entwickelt.

In der Softwareempfehlung aus dem agilen Projektmanagement von (Beck & Andres, 2004) S.58 wird im Zuge des "Extreme-Programming" (XP) das "**pair programming**" empfohlen, bei dem der Eine die Tastatur hat und den Code schreibt, währenddem der andere die strategischen Überlegungen macht und die sofort auf Probleme im Code reagiert. Weiter soll jeder seinen eignen Programm-Code erstellen. Und immer, wenn Probleme auftreten, und das kann für Anfänger sehr häufig vorkommen, ist der Tutor gleich nebenan um ihn zu fragen. Stichworte: **Schüler kann Erklären- Erklären können**.

Auf der anderen Seite hatten die **Tutoren keinen Programmierauftrag**. Vermutlich hätten die Tutoren ohnehin eine Maximalnote in kürzerer Zeit erreicht, wenn sie die gleiche Aufgabe bekommen hätten. Aber indem die Tutoren ihre drei bis vier Klassenkameraden betreuten, mussten sie erstens über deren Software bescheid wissen, wie auch Anleitungen geben, wie gewisse Dinge, wie z.B. die Verbindung zur Datenbank, zu machen ist. Es kam vor, dass sich die Tutoren hingegesetzt haben und Prototypen für spezielle Probleme, die sie nachher weitererzählen und zeigen konnten, gemacht haben. Stichworte: **Lernen durch Lehren - Schüler helfen Schülern – Schüler unterrichten Schüler**

Die Tutoren als Chief-Programmer hatten also keinen eigenen Programmierauftrag, sondern waren nur zu Erklärungen und Unterstützungen zuständig. Dennoch bekamen sie **einen fachlichen Auftrag**, nämlich auf den zweiten Modul-Halbtage einen **Kurzvortrag zu machen**, wie Extreme-Programming und insbesondere, wie "pair-programming" funktioniert und welche Vor- und Nachteile zu erwarten sind. Dies wurde auf zwei **Flip-Charts** festgehalten und im Zimmer permanent aufgehängt.

Stichworte: **Lernen durch Lehren - Schüler unterrichten Schüler – Schüler als Tutoren – Schüler als Lehrer**.

Weiter ist noch ein weiteres Stichwort hier zu erwähnen. Nämlich **Tutoren als Multiplikatoren**, wobei die Schüler die Lehrperson vervielfachen und ständig den anderen Schülern zugegen sind.

2.7 Zusammensetzung der Modulnote

Nach der Leistungsbeurteilungsvorgabe der technischen Berufsschule Zürich kann die Notenvergabe zu 100% zugunsten dem Projekt erfolgen. Ich habe dann eine Aufteilung vorgenommen indem eine

schriftliche Prüfung über gewisse angekündigte Themen, die 34% zählt, von allen gemacht werden muss. 66% der Note erfolgt dann aufgrund des Lernfortschritts und der Projektbeurteilung.

2.8 Modul-Fazit

Die Gruppenzusammenstellung erfolgte in erster Linie nach Sympathie zum CP und erst in zweiter Linie nach der Stärke (hätte ich umdrehen sollen, gemäss Feedback). Es gab demnach stärkere und schwächere Teams. Die Lernenden hätten lieber ein Gefälle innerhalb der Teams gehabt. Aber die genauere Stärken-Verteilung war zu Beginn des Modules noch nicht so genau zu sehen.

Ausnahmslos war ein sehr hoher Einsatz aller festzustellen. In persönlichen Gesprächen wurde mehrheitlich gesagt, dass sie sehr gut arbeiten konnten, sehr viel gelernt haben und dass man das Konzept wieder anwenden könne, weil immer sofort jemand für Hilfe da war. Auch die Nicht-CP haben mit der Zeit gegen Ende der 10 Modul-Halbtage den Kollegen nebenan oder am anderen Teams Unterstützung gegeben.

Es gab Zeiten, in denen die Tutoren, gehäuft gegen Ende der Modul-Halbtage "wenig zu tun" hatten. Ich hätte denen von Anfang an noch eine alternative Programmieraufgabe geben können für solche Zeiten, z.B. ein "Lehrbeispiel" einer Polymorphie-Anwendung oder sowas ähnliches zu erstellen und das dann der Klasse vorzustellen.

Von den bei (Hanel, 1991) am Schluss aufgezählten 14 Erfahrungen kann ich mindestens 12 bestätigen. Herausheben möchte ich den letzten Punkt, den ich auch im Evaluations-Fragebogen bestätigt bekam, dass nämlich Zitat: "Aus zahlreichen Schülerbefragungen geht hervor, daß diese Methode von den Betroffenen selbst weitgehend positiv beurteilt wird".

3. Konzeptvorlage für eine weitere Durchführung

Das ist ein Beispiel dafür, wie auf eine ganz spezielle Anfangskonstellation im Sinne einer Situations- und Bedingungsanalyse eingegangen wurde. Immer wenn eine solche ähnliche Konstellation vorliegt, kann man dieses Konzept auf Anwendung prüfen.

3.1 Vorbedingungen

Wichtig ist die **Freiwilligkeit**. Das ist für viele Ansporn. In der Regel machen das die «Köner» auch gern, auch wenn sie Informatiker sind. Den der Mensch zeigt grundsätzlich sein Können gern, wenn es gefragt ist.

Gut anwendbar, wenn sich **Kurse aufeinander aufbauen** und ein Vorkenntnis-Gefälle der Schüler vorhanden ist. Zudem müssten sich die «besseren» Schüler als Tutoren, insbesondere als Chief-Programmer eignen. Aus Sicht der Nicht-Tutoren-Schülern werden sie zu Experten.

Von grossem Vorteil ist, wenn man als Lehrperson **die Klasse kennt**. Das ist beim System der Technischen Berufsschule Zürich wo die Stundenpläne nicht grundsätzlich in Klassenzügen zusammengestellt werden, nicht so einfach gegeben.

Ist überall dort geeignet, wo **Projektarbeit vorgegeben oder empfohlen ist**

Bei uns in der IT drängt sich von den beiden Modulen M226a und **M226b** das letztere auf. Aber vermutlich ist auch das Modul **M404** als Folgemodul von M403 dafür geeignet.

Jedoch muss es in erster Linie bestimmte und genügend Personen sich zum Experten eignen geben. Das kann natürlich am besten aus der Erfahrung eines Vormoduls eruiert werden.

3.2 Teameinteilung

Die Teameinteilung habe ich nach einer Pause gemacht, nachdem die Tutoren bestimmt und das Konzept kommuniziert war. Und zwar nach Sympathie und persönlicher Präferenz. In der Evaluation wurde von einem Schüler gesagt, dass es besser gewesen wäre, wenn immer pro Team ein grosses Vorkennnisgefälle vorhanden sein sollte. Bei uns gab es «bessere» und «weniger starke» Teams.

Die Teamaufteilung kann auch aufgrund der eigentlichen Projekt-Präferenz gemacht werden. Eine Aussage aus der Befragung war, dass mehr Projektideen hätten zur Auswahl gestellt werden sollen. So drängt sich natürlich eine zielproduktartige Aufteilung auf.

3.3 Nichtklassisches Verhältnis und Rollen von Lehrpersonen und Lernenden

Eine der wichtigsten Erkenntnisse ist, dass wenn die Lerninseln mal am arbeiten sind, dass sich die Lehrperson stark zurücknimmt. Es geht vielmehr darum, die Klassen-Inputs nur noch selten oder gar nicht mehr zu geben um den Arbeitsfluss (den «flow») nicht zu stören. Die Lehrperson wird zum Coach und geht nur noch von Team zu Team um interessiert zuzuschauen, die Arbeiten positiv bestätigen und das was gearbeitet wird, zu loben und zu verstärken. Siehe auch [2.4](#) Blick in die Literatur und theoretischer Hintergrund

Auf Fragen eingehen, das ist klar, aber nicht mit Sofortlösungen aufdringlich werden, das ist die zentrale Empfehlung. Zweitens ist streng darauf zu achten, dass die Aufgabe und Autorität der Chief-Programmer / Tutoren nicht untergraben wird.

Weiter sind nach Bedarf, die Tutoren von der Lehrperson zu unterstützen. Durch die Standup-Meetings könnten Probleme angesprochen werden, auf die die Lehrperson reagieren muss. Aber oft sind Probleme nicht grad offenkundig. Um nichtoffenkundige Probleme zu erkennen, geht die Lehrperson unregelmässig bei den Tutoren vorbei und erkundet in informellen Gesprächen, ob die Tutoren Hilfen brauchen, falls diese nicht angesprochen werden.

3.4 Durchführung

Die Durchführung würde ich wieder gleich machen wie oben im Erfahrungsbericht beschrieben. Es ist jedoch darauf zu achten, dass die Design-Arbeit, die zu Beginn vom ganzen Team gemacht wird, von allen getragen werden kann.

Im Laufe der Projektarbeit sind die regelmässigen, kurzen Standup-Meetings pro Team und eines unten den Tutoren gemacht worden. Daran soll unbedingt festgehalten werden.

Als Lehrperson ist darauf zu achten, nicht aufdringlich zu werden. In einem gewissen zeitlichen Abstand vor und/oder nach den Standup-Meetings kann bei den Lernenden vorbeigegangen werden um zu fragen, wie es so läuft. Oft wollen die Lernenden von Ihren Erfolgen erzählen. Das soll unbedingt angehört werden. Aber es ist immer darauf achten, dass nicht gleich mit Lösungen die Gedankengänge blockiert werden. Was man machen kann ist, dass Vor- und Nachteile gewisser Konzepte diskutiert werden können. Solches unbedingt in kleinem Rahmen auch wenn das Risiko besteht, dass man dasselbe mit einem anderen Team nochmals macht.

3.5 Ergänzende Aufträge für Tutoren

Es kam gegen Ende des Moduls (7. bis 9. Modul-Halbtage) mehrfach vor, dass die Tutoren «arbeitslos» waren. Oder dass, wie es eine Tutorin machte, «ihre» Lernenden am Tisch zuerst ein «Online-Tutorial» zu einem spezifischen Thema machen liess, mit dem sie zuvor auch sehr gute Erfahrungen gemacht hat. Mit dem Effekt, dass sie dann für mehrere Stunden keine Aufgaben mehr hatte und sich entsprechend selber bildete (oder beschäftigte, eventuell sogar fach- und artfremd?). Es ist also darauf zu achten, dass die Tutoren neben dem Anreizsystem auch Pausenaufträge bekommen, wenn sie «zu wenig» zu tun haben. Für diese Fälle sollen die Tutoren eine Arbeit haben, die sie einerseits etwas fordert, wie auch für die Klasse von Nutzen sein kann in dem die Arbeit zum Input für die Klasse wird und sie dann dadurch zu zusätzlichen Teil-Experten werden. Solche Themen könnten sein:

- Anbindung von Datenbanken und Verwendung von JSON-Files für die Datenhaltung
- REST-API, REST-Aufruf- und Behandlungsstruktur
- Service-Architektur
- Polymorphismus
- Ausgedehnte Schnittstellenarchitektur (interfaces)
- Saubere Gestaltung von Backend mit 2 unterschiedlichen Frontends (Mobile & Desktop)
- Spezielle JUnit-Tests

Alle diese Themen eignen sich dafür, als Vorträge gehalten zu werden.

3.6 Tutoren-Anreiz-System

Die Tutoren sind so zu bestimmen, dass die Besten der Klasse diese Rollen erfüllen können. Es ist zu erwarten, dass diese Schüler ohnehin eine Note über eine 5 machen werden. Dadurch kann man diesen Schülern auch eine **Minimalnote** für das «Projekt» von einer **5.5** versprechen, wenn sie ihren Job zufriedenstellend erfüllen.

Dazu ist dann ein Fragebogen für die Lerninsel gegen Ende des Projektes und der Projektarbeit, bzw. der Modul-Halbtage auszufüllen und der Lehrperson zu kommunizieren. Alle der Lerninsel sollen diesen gewissenhaft ausfüllen, inkl. Selbsteinschätzung der Tutoren. Dieser könnte folgendermassen aussehen:

Ausgangspunkt ist die Note der Projektarbeit als Tutor: Projekt-Note = 5.5

- | | | |
|---------------------------------|-------------------|-------------------------------|
| - ungenügende Unterstützung | Betreuungs-Note 3 | → Projekt-Note geht auf 5.3 |
| - genügende Unterstützung | Betreuungs-Note 4 | → Projekt-Note bleibt bei 5.5 |
| - gute hilfreiche Unterstützung | Betreuungs-Note 5 | → Projekt-Note geht auf 5.7 |
| - souveräne Unterstützung | Betreuungs-Note 6 | → Projekt-Note geht auf 6.0 |

Natürlich können die Vorträge der Tutoren mit in die Bewertung einfließen.

4. Fazit

Dieses Tutoren-Konzept (auch Chief-Programmer-Konzept) ist eine sehr interessante Durchführungsart für projektbasiertem Unterricht, bei dem jeder einzelne viel lernen kann. Es ist sicher keine Alltagsdidaktik und Methodik, weil recht viele Bedingungen erfüllt sein müssen, bis an

dieses Konzept herangegangen werden kann. Aber wenn die Vorbedingungen erfüllt sind, ist dieses Konzept sehr zu empfehlen. Das Konzept ist nicht neu, sondern es wurde schon vor über 200 Jahren erwähnt. Weitere Studien und Erfahrungen zählt uns (Feldmann, 2002) unter dem Stichwort «Schüler helfen Schüler» auf. Unter anderem wird da erwähnt, dass etwas erst richtig verstanden wurde, wenn es erklärt werden kann. Als weiteres ist hier eine Sozialform des Gruppenunterrichts aufgesetzt, indem nicht primär die Lehrperson spricht, sondern die Schüler mehrheitlich zu Wort kommen.

5. Quellenverzeichnis

Beck, K., & Andres, C. (2004). *Extreme Programming Explained : Embrace Change*. New Jersey, United States: Addison-Wesley Educational Publishers Inc.

Feldmann, K. (2002). Von http://www.feldmann-k.de/tl_files/kfeldmann/pdf/bildung-erziehung/schueler_helfen_schuelern.pdf abgerufen

Hanel, P. (1991). Von <https://www.lidl.de/Material/EmpfLiteratur/hanel.htm> abgerufen

M226a, i.-b. (2019). *M226a: Klassenbasiert (ohne Vererbung) implementieren*. Von <https://cf.ict-berufsbildung.ch/modules.php?name=Mbk&a=20101&cmodnr=226a&noheader=1> abgerufen

M226b, i.-b. (2019). *M226b: Objektorientiert (mit Vererbung) implementieren*. Von <https://cf.ict-berufsbildung.ch/modules.php?name=Mbk&a=20101&cmodnr=226b&noheader=1> abgerufen

Ruggerio, M. (2012). *Modul 226: Objektorientiert Programmieren*. Zürich: Compendio Bildungsmedien.

6. Abschlussbemerkung

Ich bestätige hiermit, dass ich die vorliegende Arbeit ohne fremde Hilfe und Verwendung anderer als angegebener Hilfsmittel verfasst habe. Sämtliche Textstellen, die nicht von mir stammen, sind als Zitate gekennzeichnet und mit dem genauen Hinweis auf ihre Herkunft versehen. Die verwendeten Quellen sind im Literaturverzeichnis aufgeführt.

Uster, den 11. Februar 2019



Harald G. Müller

7. Anhang

Feedback-Formular

Anonymisierte Antworten. Rücklauf: 17 von 19 Aufforderungen. Zeitpunkt der Befragung: 30 min vor Modulende am 29.1.2019.

Die Lehrerin bzw. der Lehrer hat den Unterrichtsstoff kompetent vermittelt	Ich habe viel gelernt	Die Lehrerin bzw. der Lehrer hat bei Verständnisschwierigkeiten zusätzliche Hilfen angeboten	Die Unterrichtsmethoden waren abwechslungsreich	Die Arbeitsatmosphäre, die von der Lehrerin bzw. von dem Lehrer ausging, war gut	Die Arbeitsatmosphäre, die von den Schülerinnen und Schülern ausging, war gut	Die Notengebung war nachvollziehbar	Mein Arbeitseinsatz war hoch	Das sollte die Lehrperson ändern	Das sollte die Lehrperson beibehalten	Besonders, Bemerkungen
Punkte 1-6	Punkte 1-6	Punkte 1-6	Punkte 1-6	Punkte 1-6	Punkte 1-6	Punkte 1-6	Punkte 1-6			
3	5	5	4	6	4	5	4	Mehr Auswahl Möglichkeiten bei den Projekten, weil eigentlich alle das gleiche haben (Flix Bus).	Frei arbeiten an Projekten!	-
2	3	6	6	4	6	4	6	Schüler nach stärken und schwächen einteilen und dann Gruppen ausgeglichen bilden.	Es war eine coole Idee, dies so zu machen.	-

6	6	6	6	6	6	6	6	Bleiben Sie so wie Sie sind :D	Bleiben Sie so wie Sie sind :D	In diesem Modul habe ich mehr gelernt als in der Gaze Lehr! Diese Unterrichtsmethode sollte weitergeführt werden!
6	4	6	6	6	6	6	4	Sicherstellen das alle mitmachen & gefördert werden.	Freundlichkeit und Freude an der Arbeit.	Keine
5	6	5	6	5	4	6	5	nichts	Ich denke das mit den Chief Programmern ist eine gute Sache und sollte weitergeführt werden.	Mir hat das Modul sehr gut gefallen, insbesondere auch wegen der Selbstständigkeit.
6	6	6	6	6	6	6	6	-	Weiter Motiviert sein und weiter so gut erklären.	Ich habe bei Ihnen, viel mehr gelernt, als ich in BBW in den 1 Lehrjahr gelernt habe,
6	6	6	6	6	6	6	6	-Zeit nehmen, sich die fertigen Projekte anzuschauen (inkl. Doku etc.)	-freundlicher Umgang -kompetente Antworten -den Umständen angepasste Bewertungen (Zweitlehre etc.)	Vielen Dank für die Interessanten Stunden :)
5	5	3	5	5	4	5	4	Die Gruppen waren auch wenn sehr durchmischt, leider in meiner Gruppe bestehend aus einigen der weniger Starken. Der beobachtete Fortschritt von anderen Gruppen kam daher sehr kontrastreich daher im Vergleich zu uns.	Die Projektleitung / Teamleiter Idee gefiel mir sehr, dies können Sie so gerne weiterführen.	Probleme mit dem WLAN :)

4	4	4	3	5	5	6	4	- Das wöchentliche "Standup" fand ich eigentlich gut, allerdings würde ich es erst am Ende des Morgens machen, damit man als Schüler auch die Möglichkeit hat zu sagen, was man gerade macht und was man gerne in Zukunft erledigen möchte. - Sobald die Gruppe angefangen hat selbständig zu arbeiten, hatte man als "Chief Programmer" etwas wenig zu tun.	Die Organisation mit Kleingruppen fand ich angenehm zum Arbeiten, da so der Austausch unter Schülern höher ist, als wenn jeder sein eigenes Projekt durchzieht.	-
5	5	5	4	5	5	4	6	Die Gruppenverteilung sollte weniger nach Präferenz als nach individueller Stärke erfolgen, das heisst, in jeder Gruppe sollte mind. ein starker und ein schwacher Schüler sein.	Die Einteilung in selbst organisierte Gruppen mit CP ist interessant, weil so die CPs auch einiges über Projektmanagement lernen.	Die Handlungsziele/Kompetenzraster sollten früher verteilt werden, damit man sich besser auf die Anforderungen einstellen kann.
4	1	3	4	4	4	3	6	Einige Noten waren meiner Meinung nach etwas hoch für das gegebene Resultat.	Die Freiheit beim Arbeiten war sehr angenehm.	-

5	6	5	5	6	5	6	6	Habe momentan keinen Punkt zum ändern.	Eigentlich die gleiche Vorgehensweise, wie bei jetzt.	Ich finde das Chief-Programming System spannend.
4	5	4	4	5	5	5	5	-		
5	6	4	4	4	5	5	4	Intensiver über ein Projekt gewisse Workshops leiten, damit man eventuell mehr Funktionen implementieren kann.	Dass dieser sympatisch war und mir bei jeder Frage möglichst geholfen hat.	Als frischer Java-Anfänger kann es anfangs zu Frustrationen führen, bis man eine Lösung gefunden hat. Vorher jedoch ist es eine Hürde, das war meine Lektion in diesem Unterricht (zwar grösstenteils alleine.)
4	5	4	6	6	5	4	4	Projekt bei Noten geben anschauen	so cool zu sein	
6	6	5	5	6	4	5	6	Nichts	Alles	Ich finde es gut, dass er uns selbständig arbeiten liess.
5	6	5	6	5	5	6	6	Standup am Ende des Tages	Teams	Die Teams waren sehr gut.