



# Auswertung von objekt-orientierten Softwareentwicklungsmethoden

József Tick

Seit 10 Jahren besteht die Software Engineer Ausbildung im Fach Informatik an der Technischen Hochschule Kálmán Kandó. In diesem Aufsatz werden die Erfahrungen der letzten Jahre, besonders im Einsatz von objekt-orientierter Softwareentwicklung zusammengefaßt.

## Ursprüngliche Zielsetzung bei der Gestaltung des Ausbildungsprogrammes

Die Erwartungen bezüglich des Ausbildungsprogramms können folgendermaßen zusammengefaßt werden:

- Vermittlung methodologischer Grundkenntnisse zur Softwareentwicklung
- Förderung der Problemlösungsfähigkeit bei komplexen Großprojekten
- Erarbeitung von Kenntnissen im Software Projekt Management
- Förderung der Teamarbeit (Rollen in Teams)
- Entwicklung von Fähigkeiten hinsichtlich des Umgangs mit CASE Werkzeugen
- Vermittlung praxisorientierter, anwendbarer und marktfähiger Kenntnisse

## Ausbildungsform zur Erfüllung der Zielsetzung

In Anbetracht der Prämisse, daß alle Informatik-Studenten bestimmte Grundkenntnisse aus dem Bereich Software Engineering benötigen, aber nur ein Teil der Studenten Software Engineering als Fachrichtung wählt, um Überschneidungen zu vermeiden, wurde der Lehrstoff in 2 Module gegliedert: So entstand die in nachstehender Tabelle aufgeführte Gliederung der Lehrinhalte.

Fach/Wochenstunden im Semester	III.	IV.	V.	VI.
Vorlesungen + Praktika				
Für alle Informatik Studenten:				
Einführung in das Software Engineering	2+0	0+2		
Für die Studenten der Fachrichtung „Software Engineering“:				
Software Engineering		3+2	2+1	
CASE Werkzeuge				2+2

Die Themen der einzelnen Fächer bauen aufeinander auf, so daß diejenigen Studenten, die die Fachrichtung „Software Engineering“ studieren, ein kompaktes, einheitliches Modul kennenlernen. Es wurden folgende Themen festgelegt:

- Software-Lebenszyklus-Modelle

- Software-Projekt-Management
- Systemanalyse
- Software-Requirement-Analyse
- Softwareplanung
- Strukturierte Software-Entwicklungsmethoden
- Objekt-Orientierte Software-Entwicklungsmethoden
- Entwicklung von Benutzeroberflächen
- Implementation von Software
- Qualitätssicherung von Softwarelösungen

In den Übungen werden zur Vertiefung und Festigung von Inhalten der Vorlesungen praktische, strukturierte und objekt-orientierten Methoden durch den Einsatz von CASE-Werkzeugen trainiert. Die Studenten bilden Softwareentwicklerteams zu je 4-6 Personen, die komplexe Aufgaben (Projekte) lösen sollen.

## Unterrichtsspezifische Kriterien zur Auswertung und Auswahl von objekt-orientierten Softwareentwicklungsmethoden

Bis vor 6 Jahren waren die objekt-orientierten Softwareentwicklungsmethoden im Ausbildungsprogramm der oben genannten Module nicht vorgesehen. Stattdessen wurden den Studenten die durchaus bekannteren strukturierten Methoden (Yourdan, Jackson, Warnier, SSADM, usw.) beigebracht. Infolge der ständigen Weiterentwicklung der objekt-orientierten Techniken (OOA und OOD), nahm die Objektorientierung in unserer Fachrichtung „Software Engineering“ langsam den Platz der strukturierten Methoden ein.

Ein wichtiges Kriterium bestand sowohl bei der Weiterentwicklung der vorhandenen Studienfächer des Software-Engineering als auch bei der Auswahl der objekt-orientierten Softwareentwicklungsmethoden darin, daß diejenigen Studenten, die die Studienrichtung „Software Engineering“ gewählt haben, bereits über umfangreiche Softwarekenntnisse verfügen sollten. Dies bedeutet, daß die theoretischen Grundlagen und die praktischen Fähigkeiten für das strukturierte und objekt-orientierte Programmieren bereits vorhanden sein sollten.

Vor der Auswahl der objekt-orientierten Methoden sollen zunächst die Auswahlkriterien festgelegt werden. Diese könnten wie folgt bestimmt sein:

- die ausgewählte Methode soll in das Ausbildungsprogramm harmonisch hineinpassen
- die Methode soll in der Praxis gut anwendbar und weit verbreitet sein
- die Methode soll eine gut durchschaubare und klare Darstellungstechnik haben



- handelsübliche und finanzierbare CASE-Werkzeuge sollen für die Methode vorhanden sein
- die Methode soll das Modellieren mit einfachen Mitteln unterstützen
- die Methode soll eine gute Möglichkeit zum statischen und dynamischen Modellieren gewährleisten

### Auswertung und Auswahl der objekt-orientierten Methoden

Die Fachrichtung „Software Engineering“ wurde unter Berücksichtigung der oben angegebenen Auswahlkriterien ständig weiterentwickelt. Hinsichtlich der „Marktfähigkeit“ des Ausbildungsprogrammes und der unterrichtsspezifischen Kriterien sowie der thematischen Aufbau, und unter Berücksichtigung der Langfristigkeit wurden die verschiedenen Methoden im Ausbildungsprogramm nach und nach gewechselt. Um die Erfahrungen korrekt auswerten zu können, soll hier das bereits mit Diplom abgeschlossene Studienjahr 1996/97 betrachtet werden. Aufgrund der angegebenen Auswahlkriterien enthielt das Ausbildungsprogramm für Software Engineering drei verschiedene objekt-orientierte Softwareentwicklungsmethoden.

- Die Kontinuität, die Verbindung von strukturierten und objekt-orientierten Methoden, die Darstellungsmöglichkeit der Weiterentwicklung der Methoden, und nicht zuletzt die einfache Darstellungstechnik sprachen dafür, daß der Coad/Yourdon OOA und OOD als erste im Ausbildungsprogramm aufgenommen wurden.
- Die gut durchdachte und ausgearbeitete Methodologie, die hier eingeführte „Verantwortungssicht“, die Verwendung von „hierarchy“ und „collaboration“ Graphen, das gute Modellieren, die einfachen und „preiswerten“ Techniken (wie z. B. Class Card) und die gute Anwendbarkeit in der Praxis waren die wichtigsten Gründe, um die Wirfs-Brock „Responsibility driven“ Methode als nächste in das Ausbildungsprogramm einzubeziehen.
- Als dritte Methode wurde die gut bekannte und immer populärer werdende Rumbaugh – Object Modelling Technique (OMT) in das Programm eingebaut. Die Wahl kann leicht begründet werden durch die gut ausgearbeitete Methodologie, die ausgezeichneten statischen und dynamischen Modellierungsmöglichkeiten, die einfachen, gut verwendbaren und sehr verbreiteten Darstellungstechniken und die CASE-Unterstützung.

### Erfahrungen in der Ausbildung

Die folgenden Erfahrungen beziehen sich nicht so sehr auf die Software-Engineering-Ausbildung, sondern viel mehr auf den Einsatz von objekt-orientierten Methoden, auf die Richtigkeit der Auswertung und der Auswahl und auf die Akzeptanz von Seiten der Studenten:

- Im allgemeinen soll festgehalten werden, daß in den letzten 5 Jahren ein bedeutender Anstieg des Interesses für das objekt-orientierte Programmieren zu verzeichnen ist. (Die umgekehrte Reihenfolge weist auf die Erscheinungsfolge hin.) Es ist ebenfalls fest-

zustellen, daß die früher eher als feindliche – bestenfalls neutrale – fachliche Umgebung heute an objekt-orientierten Techniken interessiert ist. Diese Tendenz ist auch in der Vermarktung stark zu spüren. Das Interesse an verschiedenen OMT und UML Kursen ist stärker als je zuvor.

- Die Akzeptanz der 3 ausgewählten Methoden seitens der Studenten ist eindeutig. Die schrittweise Einführung, die aufbauende, steigende Komplexität der Methoden wirkt sich sehr positiv auf das Verständnis aus. Nach vorliegenden Erfahrungen, wählen 1/3-1/2 der Studenten noch die „gute, alte strukturierte“ Methode, wenn die Methode angewendet werden soll, nicht vorgeschrieben ist. Selbstverständlich spielt der Aufgabentyp bei der Wahl eine wesentliche Rolle.
- In letzter Zeit ist festzustellen, daß sich die Studenten bei komplexen Aufgaben der objekt-orientierten Analyse zuwenden. Die angebotenen Darstellungstechniken werden bei kleineren Aufgaben gern verwendet. Besonders gilt dies für die einzelnen Elemente der ausgewählten Methoden. Aufgrund der verwendeten CASE-Werkzeuge sind die OMT bzw. das C/Y Objektmodell und das dynamische Modellieren der OMT sehr populär.
- Die Anwendung von CASE-Werkzeugen, die auf objekt-orientierten Methoden beruhen, ist nicht so verbreitet, wie dies in der Informatik-Ausbildung zu erwarten wäre. Man sieht oft Studenten in Laboren mit Bleistift und Papier modellieren. Der Grund hierfür besteht darin, daß die CASE-Systeme sehr groß und komplex sind, und daher die Einsatzmöglichkeiten sehr begrenzt sind, sie stehen den Studenten nur im Laboren zur Verfügung. So gibt es leider keine Möglichkeit, diese Systeme in der Freizeit oder zu Hause zu benutzen. Die schnellere Zugriffsmöglichkeit von zu Hause, der sogenannte „Minimaleditor“ wird vielleicht dieses Problem lösen können.
- In komplexeren Aufgaben, wie z. B. Diplomarbeiten, oder anderen wissenschaftlichen Studentenarbeiten sind objekt-orientierte Techniken wie z.B. Fallstudien, Aufarbeitung der verschiedenen Methoden oder Vergleichsanalyse der Methoden sehr oft zu finden. Der Erfolg dieser Arbeit wurde durch nationale und internationale Anerkennung schon mehrmals nachgewiesen.

Die mehrjährige Erfahrung, die positiven Rückkopplungen, das immer steigende Interesse der Studierenden weisen darauf hin, daß die stufenweise Einbeziehung von gut ausgewählten objekt-orientierten Methoden in das Ausbildungsprogramm des Software-Engineering erfolgreich ist. Dieser Prozeß ist noch nicht beendet: Die neuen Methoden, die neuen CASE-Werkzeuge und das ständig wachsende Interesse erfordern eine kontinuierliche Weiterentwicklung.

### Verfasser

Prof. Dr. József Tick

Technische Hochschule Kálmán Kandó, Budapest/Ungarn  
Tavasmező utca 17, H-1084 Budapest

Tel. (00 36 1) 2 10 14 37

E-mail: tick@nfs.jozsef.kando.hu