

Software-Engineering objektorientiert - Eine Herausforderung für die Praxis

Horst Lichter
Schweizerische Bankgesellschaft Zürich

Zusammenfassung

In diesem Beitrag formuliere ich am Anfang einige wesentliche Gründe, warum die objektorientierte Technologie (OO-Technologie) in der industriellen Software-Entwicklung eingesetzt wird oder werden wird. Anschließend werden Randbedingungen und Einflußfaktoren vorgestellt und diskutiert, die berücksichtigt werden müssen, wenn die OO-Technologie in einem Unternehmen eingeführt werden soll. Darauf aufbauend wird am Beispiel der Schweizerischen Bankgesellschaft (SBG) erläutert, wie dies organisatorisch durchgeführt werden kann.

1. Warum Objektorientierung? - Einige Gründe

Es gibt verschiedene Gründe, warum die OO-Technologie zur Zeit in vielen industriellen Software-Unternehmen eingeführt oder zum Teil bereits eingesetzt wird. Es sollen nur einige genannt werden:

- Der Aspekt „time to market“ wird auch oder ist gerade auch bei Software-Produkten immer wichtiger. Je schneller und kostengünstiger neue Produkte erstellt oder existierende den Marktanforderungen angepaßt werden können, je besser kann man sich im Markt behaupten.

Man hofft, mit den Mitteln der OO-Technologie – insbesondere durch die Wiederverwendung von Bausteinen – kürzere Entwicklungszeiten zu erzielen.

- In den Entwicklungsabteilungen großer Firmen werden mehr und mehr Ressourcen verbraucht, um die bestehenden operativen Systeme zu warten und zu pflegen. Schuld daran sind unter anderem eine extrem hohe Integration der einzelnen Anwendungen; eine saubere Schnittstellenarchitektur zwischen den Anwendungen fehlt. Änderungen können dementsprechend nicht lokal begrenzt ausgeführt werden, sondern schlagen in das gesamte Netz der Anwendungen durch.

Hier hofft man, mit den Mitteln der Datenkapselung Bausteine mit hohem inneren Zusammenhalt konstruieren zu können (Datenstrukturen und deren Operationen), die über exakt definierte Schnittstellen miteinander verbunden sind.

- Ein weiteres Manko der konventionellen Software-Entwicklung, das sich sowohl in den Entwicklungszeiten als auch im Wartungsaufwand niederschlägt, besteht darin, daß nur sehr selten Bausteine wiederverwendet werden, wenn neue „ähnliche“ Anwendungen erstellt werden müssen.

Hier hofft man, spezielle Klassenbibliotheken oder sogar Frameworks erstellen und einsetzen zu können, die einen großen Teil der immer wieder benötigten Funktionalität einer Anwendungsklasse zur Verfügung stellen.

Die Gründe, warum die OO-Technologie in der industriellen Software-Entwicklung eingesetzt werden soll, können zusammenfassend auf den folgenden Nenner gebracht werden: Software soll *schneller*, *billiger* und *qualitativ* hochwertiger erstellt werden.

Die Ansprüche, denen die OO-Technologie gerecht werden muß, sind demnach sehr hoch. Die in sie gesetzten Hoffnungen ruhen vor allem auf den Techniken Datenkapselung mit sauberer Schnittstellenarchitektur und Vererbung als Mittel zur Wiederverwendung. Daß die Hoffnungen, die hinter den genannten Gründen stehen, nicht unberechtigt sind, zeigen Beispiele industrieller Software-Entwicklungen (siehe z.B. Bürkle, 1992)

2. Einführung der OO-Technologie

In Kilberth et al. (1993) wird die OO-Technologie und ihre Einführung unter verschiedenen Gesichtspunkten – technisch, organisatorisch und wirtschaftlich – beschrieben. Ich möchte besonders auf die folgenden Aspekte eingehen: Welche Randbedingungen sind bei der Einführung der OO-Technologie zu beachten und welche Konsequenzen müssen daraus gezogen werden. Die dazu gemachten Aussagen gelten zum Teil nicht nur speziell für die Einführung der OO-Technologie, sondern gelten generell, wenn eine neue Technologie eingeführt werden soll.

Die Einführung der OO-Technologie stellt in verschiedener Hinsicht ein Risiko dar: Auf der einen Seite birgt die Technologie selbst erheblichen Risiken in sich, auf der anderen Seite kann der Schaden, der dadurch entstehen kann, daß die Technologie ungeplant und unsystematisch eingeführt wurde, ebenfalls erheblich sein. Die zuletzt genannte Risikogruppe wird gemildert, wenn die Einführung sinnvoll geplant und systematisch durchgeführt wird. Die nachfolgende Abbildung zeigt wesentliche Einflußfaktoren und Randbedingungen, die zu beachten sind.

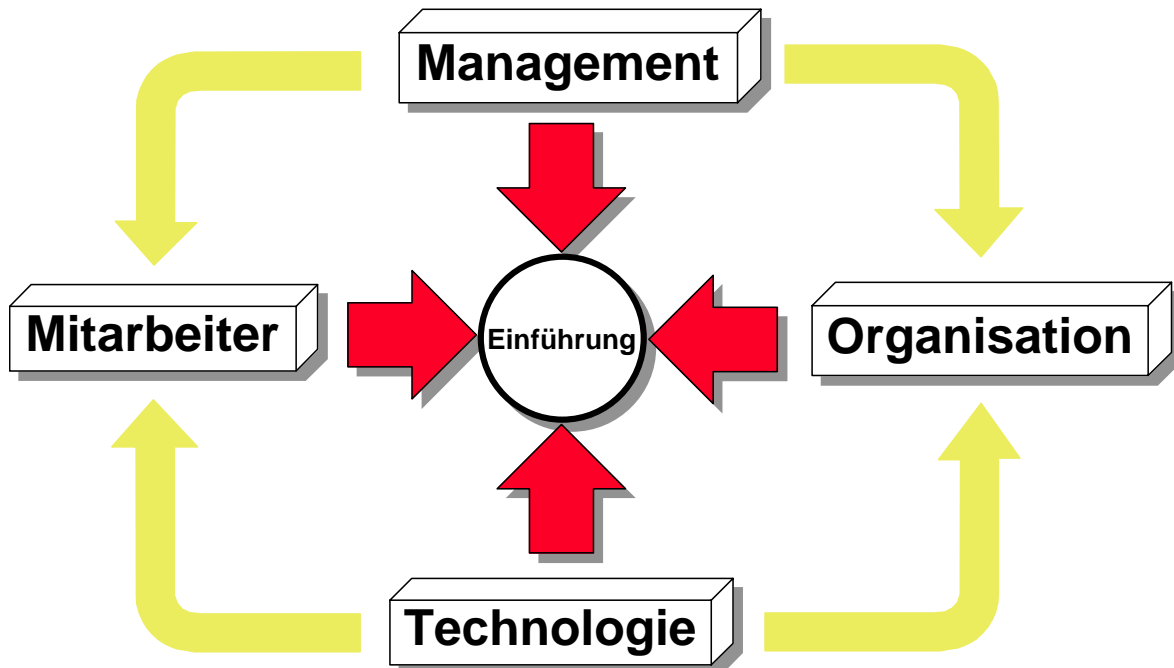


Abb. 1: Einflußfaktoren

Das *Management* muß die Einführung der OO-Technologie vollumfänglich tragen und mitverantworten. In diesem Zusammenhang müssen dem Management die Chancen, aber auch die Risiken, die in der OO-Technologie liegen, bekannt sein. Letzteres ist – bei der Flut von Lobpreisungen – besonders wichtig. Weiterhin muß dem Management klar sein, welcher Aufwand zu leisten ist, damit ein konsolidiertes OO-Engineering in der Unternehmung entstehen kann; ihm muß klar sein, daß sich die vorhandenen Potentiale der OO-Technologie sowie ein messbarer Nutzen in Form von projektübergreifender Wiederverwendung nicht kurzfristig einstellen werden, sondern erst mittelfristig zu erwarten sind. Kurz gesagt: Die Einführung der OO-Technologie muß "Chefsache" sein.

Die *Mitarbeiter* müssen die OO-Technologie in ihren Projekten umsetzen. Damit dies erfolgreich möglich sein kann, müssen sie motiviert werden, Neues zu lernen. Da dieses mit nicht unerheblichem Aufwand für jeden einzelnen Mitarbeiter verbunden ist, ist dies nicht immer einfach („Wir machen das doch schon 15 Jahre so und die Anwendungen laufen doch prima“). Wird der Aufwand von den Mitarbeitern investiert, so muß dieser „belohnt“ werden. Dies kann in extrinsischer oder intrinsischer Form geschehen. Da die OO-Technologie nicht in einer „softwarekulturfreien“ Umgebung eingeführt wird, die häufig durch die traditionelle Host-Entwicklung geprägt ist, muß verhindert werden, daß eine sich gegenseitig hemmende Zwei-Welten-Kultur entsteht. Die eine – die alte Welt – darf nicht als die

ewig-gestrige, die zweite - die OO-Welt - nicht als die allein-seeligmachende Welt dargestellt werden. Dazu muß, und dies ist Sache des Managements, der Stellenwert beider Technologien klar formuliert sein und das Mit- und Nebeneinander der Technologien deutlich gemacht werden.

In diesem Zusammenhang sei am Rande erwähnt, daß die Umsetzung der OO-Technologie nicht - wie ab und an fälschlich zu lesen oder zu hören ist - dazu führt, daß die vorhandenen Kenntnisse der Mitarbeiter nichts mehr Wert sind und nicht mehr gebraucht werden. Im Gegenteil bilden diese doch die Basis und den Grundstock, um die neuen Konzepte der OO-Technologie zu schulen und ihre Vorteile gegenüber älteren bekannten Konzepten zu erläutern.

Die *Organisationsform* der Projekte muß unter Umständen an die Bedürfnisse der OO-Technologie angepaßt werden. In Kilberth et al. (1993) wird festgestellt, daß sich in objektorientierten Projekten gezeigt hat, daß ein objektorientiertes Anwendungssystem nicht völlig unabhängig von der Organisationsform entwickelt werden kann. Die vorhandene Aufteilung in Geschäftsbereiche mit ihren jeweiligen Zuständigkeiten kann sich gelegentlich als sperrig erweisen. So ist etwa zu berücksichtigen, daß anwendungsnah arbeitende Systemanalytiker oder DV-Berater viel stärker als vorher üblich in ein Entwicklungsprojekt integriert werden müssen. Dies wirft dort Probleme auf, wo fachliche Verantwortung und personelle Zuständigkeit, bedingt durch die Organisationsstruktur, auseinanderfallen. Als Lösung empfiehlt sich meist ein Matrixmanagement, das Mitarbeiter aus verschiedenen Abteilungen für den Ablauf eines Projektes fachlich einer Projektleitung unterstellt.

Soll objektorientierte Entwicklung nicht nur in einem einzelnen Projekt, sondern in zeitlich und thematisch parallelen Projekten erfolgen, dann erfordert das sowohl umfassende Werkzeugunterstützung als auch personelle und organisatorische Voraussetzungen. Von herausragender Bedeutung für die mittelfristige Realisierung der Aspekte Flexibilität, Wiederverwendbarkeit und Offenheit ist dabei, daß die in den entstehenden Klassenbibliotheken und Frameworks dokumentierten Konzepte kontinuierlich gepflegt und weiterentwickelt werden. Dazu muß nicht nur bekannt sein, wie OO-Programmtexte technisch zu verwalten sind, sondern dies bedeutet auch, daß kontinuierlich Wissen und Erfahrung weitergegeben werden muß.

Nicht zuletzt müssen die Elemente der einzuführenden *OO-Technologie* bekannt und auf die Bedürfnisse der Projekte abgestimmt sein. Da es zur Zeit eine Vielzahl von methodischen Ansätzen im Bereich der objektorientierten Analyse und des Entwurfs gibt, verschiedene Sprachen- und Werkzeugalternativen existieren, müssen die Elemente ausgewählt und zusammengestellt werden, die den Anforderungen der ersten OO-Projekte am besten gerecht werden. Bei der Auswahl dieser Projekte muß darauf

geachtet werden, daß diese von der Aufgabenstellung besonders geeignet sind, um sie in der OO-Technologie durchzuführen. Positive Erfahrungen existieren zur Zeit im Bereich der interaktiven Auskunftssysteme und Beratungssysteme.

3. OO-Technologie bei der SBG - Strategie der Einführung

In diesem Abschnitt möchte ich kurz beschreiben, welche Strategie die SBG umsetzen will, um die OO-Technologie erfolgreich einzuführen und zu etablieren.

In der SBG existiert bereits seit ca. fünf Jahren in Form des eigenen Informatik-Forschungslabors UBILAB eine Keimzelle für die OO-Technologie. In den letzten drei Jahren wurden neben den doch eher labor-orientierten Arbeiten des UBILAB's einzelne Projekte in OO-Technologie entwickelt. Erst in diesem Jahr ist jedoch geplant, die OO-Technologie systematisch einzuführen

Marty (1994) beschreibt die folgenden Aspekte für die Einführung der OO-Technologie bei der SBG:

- Es ist ein gradueller Übergang hin zur OO-Technologie geplant, der mindestens fünf Jahre dauern wird.
- Die Einführung soll in kleinen nachvollziehbaren Schritten durchgeführt werden.
- Erste Erfahrungen sollen an speziellen Pilotprojekten erhalten werden.
- Um eigene SBG-angepaßte Klassenbibliotheken oder Frameworks zu erstellen, soll verstärkt auf zugekauften „Halb-Produkten“ aufgebaut werden.

3.1 Die OO-Gruppe - Selbstverständnis und Ziele

In der SBG wird zur Zeit eine Gruppe aufgebaut, die mit der Einführung der OO-Technologie und deren Umsetzung in den Projekten betraut sein wird. Die Etablierung dieser OO-Gruppe ist letztlich eine Konsequenz der im vorangegangenen Abschnitt für die OO-Technologie beschriebenen organisatorischen Voraussetzungen. Das Selbstverständnis und die Aufgaben der OO-Gruppe werden nachfolgend erläutert.

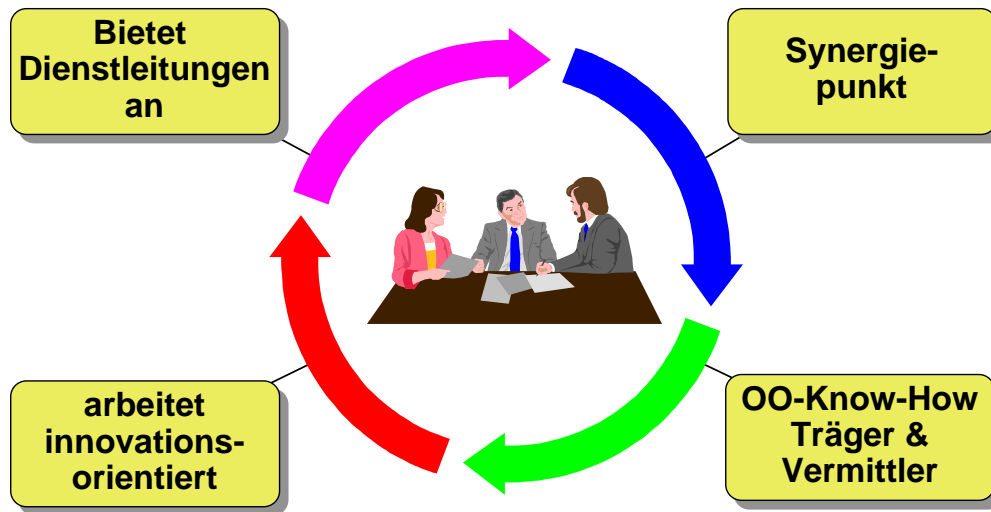


Abb. 2: Selbstverständnis der OO-Gruppe

Die OO-Gruppe ist ein Anbieter von Dienstleistungen.

Die OO-Gruppe versteht sich als ein Dienstleistungsanbieter. Die angebotenen Dienstleistungen müssen den Erfordernissen der OO-durchgeführten Projekte entsprechen. In diesem Zusammenhang ist es wichtig, daß die Projekte einen direkten messbaren Nutzen haben, wenn sie auf die Dienste der OO-Gruppe zurückgreifen. Damit dies erreicht werden kann, müssen die angebotenen Dienstleistungen bekannt sein und mehrheitlich akzeptiert werden.

Die OO-Gruppe ist der Synergiepunkt für die OO-Technologie.

Die OO-Gruppe ist die zentrale Stelle, die Ergebnisse, die in verschiedenen Projekten erarbeitet werden, aufbereitet, pflegt und anderen Projekten zur Verfügung stellt werden. Hierzu zählen insbesondere die eingesetzten Klassenbibliotheken, aber auch die verwendeten Werkzeugumgebungen. Die Chancen, die die OO-Technologie im Bereich Wiederverwendung bietet, sollen dadurch aktiv genutzt und zielführend umgesetzt werden.

Die OO-Gruppe arbeitet innovationsorientiert.

Die OO-Technologie ist noch lange nicht ausgereift, sondern unterliegt ständiger Veränderungen und Verbesserungen. Dieser Tatsache muß die OO-Gruppe dadurch Rechnung tragen, daß sie die sich stabilisierenden Neuerungen erkennt, bewertet und eventuell umsetzt. Die OO-Gruppe beobachtet dazu u.a. den sich rasch entwickelnden Markt für Methoden, Werkzeuge, Klassenbibliotheken und Frameworks.

Die OO-Gruppe ist der zentrale Know-How-Träger der OO-Technologie.

Damit die OO-Technologie systematisch eingeführt, verbreitet und stabilisiert werden kann, muß es eine zentrale Stelle geben, in der das eingesetzte Know-How zusammengefasst ist. Die OO-Gruppe muß alle aktuell verwendeten Sprachen, Methoden und Werkzeuge, sowie die praxis-orientierten QS-Massnahmen kennen, die speziell auf OO-Projekte zugeschnitten sind. Sie muß weiterhin in der Lage sein, dieses Wissen angemessen zu vermitteln.

3.2 Aufgabenfelder der OO-Gruppe

Aus den oben genannten Zielen lassen sich die folgenden zentralen Aufgabenbereiche der OO-Gruppe identifizieren:

A. Zeitliche begrenzte Mitarbeit in OO-Projekten

Die Arbeit der OO-Gruppe darf kein Selbstzweck sein. Im Sinn eines Dienstleistungsanbieters können deshalb die Mitarbeiter der OO-Gruppe in einem zeitliche beschränkten Maße in OO-durchgeführten Projekten mitarbeiten. Dies führt auf der einen Seite dazu, daß die OO-Gruppe ständig mit den Problemen und der Entwicklungssituation der Projekte vertraut bleibt und dementsprechend darauf reagieren kann. Auf der anderen Seite ist dies die Voraussetzung, um Gemeinsamkeiten im Sinn der Synergie und Wiederverwendung zu entdecken und umzusetzen.

Aus heutiger Sicht werden die folgenden Schwerpunkte beim Einsatz in einem Projekt gesehen:

- Unterstützung bei der Einarbeitung in die verwendete Entwicklungsumgebung, Sprache und Methode.
- Vermitteln der eingesetzten Klassenbibliotheken.
- Mitarbeit und Beratung im Bereich Analyse, Design- und Implementierung.
- Anstoß von QS-Aktivitäten.

B. Pflege der vorhandenen und entstehenden wiederverwendbaren Bausteine

Damit projektübergreifend nutzbare Klassenbibliotheken entstehen können, muß es eine Stelle geben, die sich intensiv mit den bestehenden eingekauften Klassenbibliotheken und mit den in den Projekten entwickelten Klassen auseinandersetzt. Dies können die einzelnen Projekte aus Termin- und Kostengründen nicht leisten. Die OO-Gruppe hat in diesem Zusammenhang die Aufgabe, in sich konsistente, aktuelle und

wiederverwendbare Klassen in Form von Bibliotheken oder sogar Frameworks zur Verfügung zu stellen.

Damit dieses Ziel erreicht werden kann, müssen seitens der OO-Gruppe die folgenden Aktivitäten durchgeführt werden:

- Die vorhandenen Klassen müssen den Projekten vermittelt werden. Beim Entwurf neuer Klassen muß darauf geachtet werden, daß die existierenden Klassen sinngemäß verwendet und eingesetzt werden.
- Entstehen in einem Projekt neue Klassen, die nicht problemspezifisch, sondern genereller Natur sind, so müssen diese in die Klassenbibliothek aufgenommen werden. In der Regel müssen solche Klassen aber einem Redesign unterzogen werden. Weiterhin kann es notwendig sein, daß existierende Klassen verändert und aktualisiert werden müssen, wenn neue Klasse in eine Bibliothek aufgenommen werden. Dies ist die Aufgabe der OO-Gruppe.

C. Aufbau eines OO-Engineerings

Die OO-Technologie ist nur auf der Basis eines entsprechenden Software-Engineerings umsetzbar. Dazu zählen alle Aspekte, die für den Entwicklungsprozeß relevant sind. Die OO-Gruppe muß in diesem Zusammenhang einen OO-Engineering-Ansatz erarbeiten, definieren und in die Projekte einfließen lassen. Dieser Prozeß wird iterativ sein, da sich das Engineering nicht ad hoc abschließend definieren läßt, sondern sich aufgrund der in den Projekten gemachten Erfahrungen anpassen und verändern wird.

Es lassen sich die folgenden groben Themengebiete im Bereich OO-Engineering identifizieren:

- Erarbeiten einer praxisnahen und praxisgerechten Qualitätssicherung für OO-Projekte (Reviews, Metriken, Testverfahren etc.).
- Erarbeiten von Standards und Konventionen (Programmierrichtlinien, Dokumentationsstandards, Bezeichnerkonzept etc.)
- Bedingt durch den Einsatz integrierter OO-Entwicklungsumgebungen wie etwa Smalltalk muß ein Konzept für ein geeignetes Versions- und Konfigurationsmanagement erarbeitet und umgesetzt werden.

D. Erarbeiten eines OO-angepaßten Schulungskonzeptes

Damit die objektorientierte Technologie in den Projekten umgesetzt werden kann, müssen die Projektmitglieder entsprechend geschult sein. Dabei ist zu beachten, daß die

einzelnen Mitarbeiter einen unterschiedlichen Wissenstand haben. Weiterhin müssen die zu schulenden Inhalte so gewählt sein, daß sie den Anforderungen der Projekte genügen. Dies kann durch ein in sich abgestimmtes Schulungskonzept erreicht werden.

Die Aufgabe der OO-Gruppe besteht in diesem Zusammenhang darin, die Inhalte und die Reihenfolge der aufeinander aufbauenden Kurse mitzugestalten. Das Schulungskonzept kann grob in zwei Kategorien gegliedert werden. Die erste Kategorie enthält Kurse, die das der OO-Technologie zugrundeliegende softwaretechnische Basiswissen vermitteln. Dazu zählt im einzelnen das Modulkonzept, das Geheimnisprinzip, die Datenkapselung, Abstrakte Datentypen, Generizität, Polymorphie, Typsysteme, dynamisches Binden, evolutionäre Prozeßmodelle und Prototyping. In der zweiten Kategorie sind Kurse enthalten, die spezielle Themen die Objektorientierung vermitteln. Folgende Kurse sind denkbar:

- Überblick über die OO-Software-Entwicklung
- Objektorientiertes Programmieren
- QS-Massnahmen in OO-Projekten
- Objektorientierter Systementwurf (Methoden, Software-Architekturen)
- Objektorientierte Frameworks - Konzepte und Designmuster

E. Aufbau und Unterhalt des OO-Entwicklungs-Environments

Die OO-Software-Entwicklung findet in zunehmendem Maße auf speziellen, teils integrierten Entwicklungsumgebungen (Smalltalk, Visual C++) statt und wird durch speziell geeignete Werkzeuge unterstützt. Die Auswahl, das Know-How und die Pflege der eingesetzten Werkzeuge sollte sinnvollerweise an einem Ort zusammengefaßt sein.

Durch den Mix von Projektarbeit und Innovationstätigkeit der OO-Gruppe können die Werkzeuge so ausgewählt und notfalls adaptiert werden, daß sie den Erfordernissen der Projekte und den Erfordernissen des OO-Engineering-Ansatzes entsprechen.

3.3 Bewertung

In diesem Bericht ist der aktuelle Stand des Denkens dokumentiert, der bzgl. der Einführung der OO-Technologie in der SBG vorhanden ist. Da wir zur Zeit erst in der Initialphase der Einführung sind, können keine Erfahrungen berichtet werden. Wir glauben jedoch mit der gewählten Vorgehensweise folgende Ziele erreichen zu können:

- Die Technologie kann geplant, schrittweise, systematisch und nachvollziehbar eingeführt werden.
- Die vorhandenen noch knappen OO-Ressourcen werden in Form der OO-Gruppe zusammengefaßt, so daß eine kritische Masse entstehen kann, die notwendig ist, damit Erfolge erzielt werden können.
- Durch die Arbeit der OO-Gruppe besteht die Möglichkeit, eine unternehmensweite OO-Basis zu bilden. Dazu zählt neben den Klassenbibliotheken insbesondere auch das Know-How im Umgang mit OO-Projekten.
- Die OO-Technologie soll nicht überall und nicht zu jedem Preis eingesetzt werden. Es sollen die Projekte in dieser Technologie realisiert werden, die besonders dafür geeignet sind. Die OO-Technologie wird nicht zum Selbstzweck, sondern als „Nutzengenerator“ eingesetzt.

Literatur

Kilberth, K., G. Gryczan, H. Züllighoven (1993): **Objektorientierte Anwendungsentwicklung Konzepte, Strategien, Erfahrungen**, Vieweg Wiesbaden.

Bürkle, U., G. Gryczan, H. Züllighoven (1992): Erfahrungen mit der objektorientierten Vorgehensweise in einem Bankenprojekt, **Informatik-Spektrum**, Vol. 15, Nr. 5, pp 273-381.

Marty, R (1994): Klassische Entwicklungstechnologien ungenügend, **Computer Woche** Extra 1, Februar, pp 38-40.