OO-Projekte im Vergleich - Erfahrungsbericht

Andres Koch, dipl. El.Ing HTL, M.Math
Object Engineering GmbH, Zürich
Kontaktformular

Vortrag anlässlich der SAQ Tagung 1996 zum Thema "Qalitätssicherung und Objektorientierung"

Erfahrungen mit der objektorientierten Entwicklungstechnik sind heute vorhanden. Der vorliegende Beitrag soll vier konkrete Projekte, welche mit OO in fast ähnlicher Umgebung gemacht wurden, miteinander vergleichen. Dies wird anhand von Kriterien im Bereich von Projektmanagement, Entwicklungsprozess und Teamzusammensetzung gemacht. Die Ergebnisse sind unterschiedlich. Es mag nicht überraschen, dass die eingesetzte Technologie eine eher untergeordnete Rolle gegenüber der Entwicklungskultur und dem Teamverhalten spielt. Eine klare Software-Architektur und eine nachgeführte Dokumentation kristallisiert sich jedoch für alle OO-Projekte als gemeinsamer Nenner für den Erfolg unterstützt durch gute Ausbildung und ein gutes Team heraus.

1 Einleitung

Die seit 1989 gesammelte Erfahrung beim Begleiten von OO-Projekten sind einerseits durch die in dieser Zeit erst entstandenen OO-Methoden geprägt. Anderseits können aber gewisse Projekt-Grundprobleme unabhängig von den verwendeten Methoden ausgemacht werden, welche beim Einsatz der Objektorientierten Technik noch einen ausgeprägteren Einfluss auf das Gesamtprojekt haben. Im Folgenden werden vier Projekte aus jüngerer Zeit vorgestellt und gegenüber den gewählten Kriterien verglichen.

1.1 Kriterien

Die gewählten Kriterien wurden ihrerseits wie folgt gruppiert:

1.1.1 Projekt-Management-Kriterien

Dabei sollen die bei der Organisation und Leitung eines Projektes auftretenden Faktoren betrachtet werden. Oft werden die Grundsteine für den Erfolg eines Projektes bereits zu Beginn gelegt, im Speziellen bei der Auswahl des Teams, des Projektorganigramms und beim Vorlegen der genügend hohen Anfangsgeschwindigkeit. Natürlich spielen die Ressourcen (Geld, Zeit, ausgebildetes Personal) eine grosse Rolle.
Da das Team den Motor des Projektes bildet wird dies als einen wichtigen Faktor ebenfalls beurteilt respektive verglichen. Vom Ausbildungstand aber vorallem auch Lernwillen hängt die Qualität der Lösung im Wesentlichen ab. Der Erfahrungsschatz war in der Regel in der noch jungen Technologie bei allen relativ klein. Das Teamverhalten, die vom Einzelnen übernommene Verantwortung und Eigeninitiative bestimmt dann mehr die Einhaltung der in der Regel zu knappen Termine und damit in der hektischen Schlussphase das Verhältnis im Team.

Der eigentliche technische Entwicklungsprozess wird in die Phasen:

- Requirement-Analyse/Domain-Analyse
- Architektur
- Design

- Implementation
- Tests
- Qualitätssicherung
- Dokumentation

untersucht, um die Unterschiedlichkeit der Projekte in den einzelnen Phasen aufzuzeigen. Dabei wird auch die Iterationshäufigkeit aufgezeigt, womit die Wiederholung oder das Zurückkehren in eine vorangehende Entwicklungsphase gemeint ist. Dies kann auch sehr wohl als Qualitätsmerkmal gewertet werden, im Speziellen wenn den früheren Phasen wenig Aufmerksamkeit geschenkt wurde.
Schlussendlich wird das Resultat verglichen, was schlussendlich das ist, was den Kunden interessiert und über den Wert des Produktes bestimmt.

1.2 Verglichene Projekte

Die verglichenen Projekte wurden aufgrund Ihrer Art gewählt.

1.2.1 Projekt A - Integrations-Projekt

Integration eines eingekauften, stark parameterisierbaren Applikations-Paketes, welches in die bestehende System-Umgebung des Unternehmens integriert werden musste. Der beschriebene Teil des Projektes betraf hauptsächlich den technischen Teil der Integration mit Hostanbindung, Kommunikation über Telex und anderen Plattformen. Realisiert wurde es mit UNIX, C++ und einer eigens dafür erstellten meldungsorientierter Middleware. Da die objektorientierten Methoden erst im Aufkommen waren, wurden sie noch nicht von Anfang an konsequent angewendet doch es wurde laufend versucht sich in dieser Beziehung zu verbessern.

1.2.2 Projekt B - Backendprozess-Projekt

Der Bau einer Auswertungsapplikation mit grosser Datenmenge und hohe Anforderungen an Präsentationscharakter für die gedruckten Auswertungen. Ebenso waren hohe Anforderungen an Durchsatz gefordert, um die Auswertungen in angemessener Zeit den Kundenbetreuern zur Verfügung zu stellen. Es wurde ausschliesslich in C++ und auf UNIX-Plattformen gearbeitet.

1.2.3 Projekt C - Frontendprozess-Projekt

In diesem Projekt wurde eine eingeschränkte, stark manuelle Prozesse unterstützte Applikation für die Erbringung von Kunden-Dienstleistungen durch eine neue Applikation ersetzt. Es waren zwei Plattformen (Host/MVS und Server/Unix) im Einsatz und es wurde sowohl auf dem Host wie auf dem Server mit C++ gearbeitet.

1.2.4 Projekt D - Migrations-Projekt

In einer typischen 3-Tire-Architektur in einem Grossunternehmen mit mehr als 10 Aussenstellen und 2500 Benutzern musste eine nicht mehr erhältliche Hardware durch neue Hard-/Software ersetzt werden, ohne dass die Applikations-Software massgebend verändert wurde. Mit einer schrittweisen Migrations-Strategie konnte mit sehr geringem Aufwand ein wichtiger Teil des Gesamtsystems erfolgreich auf eine modernere CORBA-basierte und objektorientierte Architektur überführt werden. Bei den involvierten Programmen handelte es sich um PL/1- und Assembler-Programme auf proprietären Plattformen (u.a MVS). Für die Migration wurde mit C später mit C++ gearbeitet und durch die Aufteilung in verschiedene Prozesse wurde eine hohe Flexibilität erreicht. Neben der bestehenden Plattform, wurde das abzulösende System durch einen modernen RISC-Server mit UNIX ersetzt und die Bildschirme durch die üblichen PCs.

2.1.1 Projektmanagement

Projekt A: Das Projekt wurde durch eine praxiserfahrene Person initialisiert. Mit eher managementorientierter statt technischer Einstellung wurde nur gerade die Integrations-Schicht im Hause entwickelt und die Haupt-Applikations-Komponente eingekauft. Damit konnte nicht nur Zeit sondern auch Erfahrung gewonnen werden.
Projekt B: Ein eher projektunerfahrenes und technisch orientiertes Projektmanagement, hat gerade die gegenteilige Einstellung vorangetrieben, nämlich alles im Hause zu entwickeln. Wiederverwendung was an vorhandenen Plattformen oder Komponenten innerhalb des Unternehmens bereits vorhanden war, wurde mit der Begründung "unfertig" in den Wind geschlagen, was sich später als fatal auswirken sollte.
Projekt C: Ein konventionell aufgezogenes Projektmanagement, welches sich aber mit internen Vertuscherei, Politik und fast intriegenähnlichen Zuständen auseinandersetzen musste. Die unglückliche Zusammensetzung der Teammitglieder, war eines der grossen Handicaps dieses Projektes.
Projekt D: Ein Schatten-Projektmanagement arbeitete auf die schlanke Art, da es sich lange um "kein" Projekt sondern um einen "Notbehelf" gehandelt hat. Durch ein sehr kleines Team, welches gut kooperierte, erreichte mehr, als wenn eine grosse Projektstruktur aufgebaut worden wäre. Für die Ausbreitung wurde dann richtigerweise ein zu 100% engagierter Projektleiter eingesetzt.

2.1.2 Entwicklungsmanagement

Projekt A: Die Entwicklungsphasen wurden noch eher traditionell im Wasserfall-Modell organisiert. Mit dem Erstellen der nötigen Dokumente in den einzelnen Phasen nach einem vorgegebenen Dokumentations-Plan wurde die nötige Transparenz für die angrenzenden Systeme und Organisations-Einheiten geschaffen. CASE-Tools wurden je nach Bedarf aber nicht konsequent eingesetzt. Es wurde zu Beginn eher objektbasiert als objektorientiert gearbeitet.
Projekt B: Objektorientierung war gross geschrieben und auch konsequent durchgeführt worden. Es ergab dann aber eine hauptsächliche Konzentration auf die Design- und Implementation-Phase wobei die Dokumentation und Versionenkontrolle sträflich vernachlässigt wurden.
Projekt C: Auch hier wurde Objektorientierung angesagt, doch fiel man dann wieder in die konventionellen Entwicklungsphasen nach dem Wasserfallprinzip zurück, was aber noch korrigiert werden konnte. Die Vision auf ganz oberster Direktionsstufe konnte nicht durch die eher überschätzte Basis getragen werden, wodurch der berechtigte Eindruck entsteht, ob es für das Unternehmen ein zu hoch gestecktes Ziel gewesen sei.
Projekt D: Durch die Notwendigkeit ein funktionierendes System innert kurzer Zeit abzulösen, hat gar keine Illusionen zugelassen. Man organisierte sich produkteorientiert dafür aber in flexibler modularer Weise. Wenn die Pragmatik auch oft etwas überhand nahm, konnte mit gezielt von der technischen Seite eingebrachten Systematik trotzdem ein flexibles System gebaut werden.

2.1.2.1 Ressourcen

Finanzen

In Projekt A, B und C waren diese gut budgetiert und genügend vorhanden. Projekt D, das als "Schattenprojekt" aufgezogen war, hatte eher knapp bemessene Mittel, welche aber trotzdem für die kleine Organisation angemessen genügten.

Zeit

In Projekt A, B und C knapp aber nicht unrealistisch bemessen. Trotzdem kam es bei allen drei Projekten zu kleineren (A und B) bis grossen (C) Terminverschiebungen.
Projekt D hatte weniger offiziellen Zeitdruck, da "nichts" erwartet wurde. Dafür hat war die

Notwendigkeit, den Betrieb aufrecht zu halten und der Erfolgsdruck um so grösser. Es gab hier kleinere Terminverschiebungen.

Personal

In Projekt A standen etwa 15 Personen (intern und extern), in Projekt B und C etwa 20-25 Personen und in Projekt D etwa 6 Personen im Einsatz.

2.1.3 Vision - Illusion - Realität

Projekt A war von Pragmatik geprägt, aber die Systematik und auch das Qualitätsbewusstsein für den Entwicklungsprozess wurde vom Management mit dem nötigen Nachdruck gefördert.
Projekt B dagegen war eher eine Illusion von technisch denkenden Projektverantwortlichen, welche den wichtigen Entwicklungsphasen wie Dokumentation und Qualitätssicherung wenig beimassen.
In Projekt C wurde versucht eine Vision in die Realität umzusetzen, welche eher als Steckenpferd zu werten war, als dass sie vom Kunden goutiert worden wäre. Wie bereits erwähnt, war die Basis dafür nicht gewachsen.
In Projekt D stand die Realität respektive das zu erreichende Resultat klar und deutlich im Vordergrund, wobei aber die Vision das simulierte Altsystem mit der Zeit ganz abzulösen immer wegweisend vorhanden war.

2.2 Team-Kriterien

2.2.1 Ausbildungsstand und Lernwille

In Projekt A waren die rekrutierten Mitarbeiter recht gut ausgebildet und brachten aber teilweise auch Praxiserfahrung mit. Sie waren aber eher zurückhaltend in Bezug des Startens des Projektes, da die unternehmerische Wichtigkeit sie etwas beeindruckte. Um so lernwilliger und aufnahmebereiter waren sie für neue Methoden und Vorgehensweisen. Man darf diese Einstellung sicher als eher ideal einstufen.
In Projekt B war die technische Erfahrung recht gut, dafür bestand aber auch ein hohes Bewusstsein "es gut zu beherrschen". Ideen und Vorschläge aus anderen Projekten und Teams wurden oft mit Rechtfertigungen verworfen oder gar nicht ernst genommen.
In Projekt C war die Erfahrung und das Know-How der neu eingesetzten Technologie nicht vorhanden und mangels Akzeptanz des vom Projektmanagement organisierten umfangreichen Ausbildungsprogramms nie richtig angewendet worden. Die Teammitglieder, welche die Ausbildung besucht hatten, hätten die Methode noch angewendet, wären nicht neue Ressourcen dazugekommen, die sich als Gurus wähnten und das schon gar nicht "nötig" hatten.
In Projekt D war die Notwendigkeit neue Methoden zu lernen und vorallem anzuwenden von Anfang an bekannt, obwohl mehr als 12 Jahre Erfahrung in Systemprogrammierung vorhanden war. Durch das kleine Team war Lernwille nie ein Problem.

2.2.2 Teamverhalten

Im Projekt A hatte sich in kurzer Zeit ein kleines aber schlagkräftiges Team mit gutem sozialen Zusammenhalt gebildet. Die gegenseitige Akzeptanz war gross und die Zusammenarbeit mit dem Projektmanagement gut. Die Eigeninitiative und Uebernahme von Verantwortung für die zugeteilten Aufgaben (die in der Architektur ausgewählten Module wurden jeweils einem Teammitglied zum Design und Realisation übergeben) war sehr hoch.
In Projekt B schien der Zusammenhalt des Teams auch recht gut funktioniert zu haben, doch sobald die Fehler offensichtlich wurden, begann die Zusammenarbeit mit dem Auftraggeber eher einer kühlen Beziehung zu gleichen. Eigeninitiative und Verantwortung war unterschiedlich. Leider zeigte sich eine gewisse Apathie und sogar Aufgabe bei Teammitgliedern gegen Ende des Projektes.

Das Team in Projekt C war wie von einem Virus befallen, der vermutlich historisch aus dem Unternehmen herauskam. Dieser Virus wurde dann noch durch dazukommende "Parasiten" unterstützt respektive das Team wurde dadurch noch mehr als verkraftbar geschwächt. Das und andere Faktoren haben trotz grosser Bemühung der Projektleitung den inneren Zusammenhalt und Teamgeist auf ein Minimum sinken lassen. Was Verantwortung und Eigeninitiative anbelangt, waren auch hier unterschiedliche Verhaltensweisen zu beobachten. Einige strengten sich an irgendetwas zu bewegen, während andere mit wenig Aufwand (Bemerkungen) dies wieder zunichte machen konnten.
Die fast zufällig zusammengewürfelte Mannschaft hat sich durch grosse Toleranz und gegenseitige Akzeptanz (gleiche Wellenlänge) schnell gefunden und auch als schlagkräftig gezeigt. Mit Uebernahme von Verantwortung und die Eigeninitiative motivierten sich die Teammitglieder gegenseitig das gleiche zu tun. Keiner liess den anderen im Stich.

2.3 Entwicklungsprozess

Nachdem wir nun eher die organisatorischen und sozialen Aspekte betrachtet haben, möchten wir uns auf den eigentlichen Entwicklungsprozess konzentrieren.

2.3.1 Requirement-Analyse/Domain-Analyse

In Projekt A war die Anforderungs-Analyse des Bedarfs des Benutzers wohl das grösste Problem auf der applikatorischen Seite. Wie so oft stellte die Definition respektive die Anforderungen der Schnittstellen auf technischer Seite ein kleines Problem dar.
In Projekt B kann diese Phase mangels fehlender Informationen nicht beurteilt werden dafür wurde diese Phase in Projekt C mittels Use-Case-Technik beispielhaft erfüllt. Bei Projekt D waren die technischen Anforderungen durch das vorhandene System recht gut dokumentiert und damit auch nicht problematisch.

2.3.2 Architektur

Sowohl in Projekt A und D wurde dieser für objektorientierte Systeme sehr wichtige Teil sehr gut gelöst und darf wohl als ein weiterer Hauptfaktor für den Projekterfolg gewertet werden.
In den Projekten B und C waren Ideen, aber keine Architektur und diese wurden schon gar nicht oder erst im Nachhinhein dokumentiert. Es hätte genau eine Person gebraucht, welche die Rolle des Architekten übernommen hätte und auch als solcher akzeptiert worden wäre.

2.3.3 Design

Im Projekt A wurde nach einem aus der Architektur folgenden Modularisierung die Design-Aufgaben auf die einzelnen Teammitglieder verteilt und von denen im Detail ausgearbeitet und dokumentiert.
In Projekt B wurde das Design wirklich objektorientiert und mit viel Aufmerksamkeit auf die technische Lösung sehr gut gelöst. Leider blieb die Dokumentation dessen sträflich vernachlässigt.
In Projekt C wurde das Design eigenhändig durch eine Person gemacht, aber schlecht oder gar nicht (ausser im Code) dokumentiert. Die Auswirkungen für das Verständnis und für die Wartung des Produktes werden sich vermutlich nicht als positiv erweisen.
In Projekt D wurde konsequent mit einem CASE-Tool für eine objektorientierte Methode gearbeitet. Es wurde fast ausschliesslich Fürwort-Engineering verwendet, das heisst auf Reverse-Engineering wurde verzichtet. Das CASE-Tool wurde auch eingesetzt für das Design der objektbasierten Module, die später in 'C' (nicht objektorientierte Sprache) realisiert wurden. Damit konnte auf einfache Weise die Dokumentation des Projektes auch sichergestellt werden, die aber auch durch technische Beschreibungen unterstützt wurde. Das Vorgehen hat sich vorallem später bei der Weiterentwicklung gut bewährt.

2.3.4 Implementation

In Projekt A wurde die Implementation von "Hand"in C++ durch die Modulverantwortlichen realisiert. Die Integration erfolgte in der Regel (eine Ausnahme beim eingekauften Produkt) durch die Interprozesskommunikation und hatte darum ausser bei den Meldungs-Klassen und der IPC-Middleware wenig Abhängigkeiten.
Bei Projekt B und C kamen monolythische Programme zustande und dies mit der modernsten Werkzeuge des Software-Engineerings. Entsprechend gross wurden die Abhängigkeiten und langwierig die Generierung (Compilation) des Systems. Daraus entstanden eigentlich unnötige organisatorische Behelfslösungen, die dieses Problem zu lindern versuchten. Die Wartung beider Programme zeichnet sich als problematisch ab.
Projekt D wurde für die C-Teile von Hand programmiert später aber durch das CASE-Tool bei der Verwendung des Code-Generators für C++ stark unterstützt. Die lose Kopplung durch die Verwendung von Plattform-IPC und später einer CORBA-Implementation hat sich analog wie in Projekt A als sehr flexible und leichtgewichtige Vorgehensweise erwiesen. Dadurch kam bei Folge-Modulen auch die Wiederverwendung zum Zug.

2.3.5 Tests

Die Tests wurden in Projekt A und D auf Modulebene mit Teststubs gemacht, bevor das System integriert und als Gesamtes getestet wurde. Dies hat eine zeitlich parallele Entwicklung erlaubt, was bei hohen Terminanforderungen positiv auszahlte.
Während in Projekt C ein grosser Aufwand für den Aufbau eines Changemanagement-Systems und Testsystems mit Hilfe eines Testtools aufgewendet wurde, hat man vorallem den Changemangement-Aspekt in Projekt B sträflich vernachlässigt. Bei beiden Systemen war das Testen ein sehr mühseliger Prozess, nicht zuletzt weil es tatsächlich "Entwickler" gab, die Ihre Module ungetestet dem Test-Team übergaben. Durch die grosse Abhängigkeit der einzelnen Module untereinander und die langwierige Generations-Phase wurde das Testen für die Entwickler zum mühseligen Arbeitsprozess.

2.3.6 Qualitätssicherung

In Projekt A wurden durch frühzeitige "Walkthroughs" in der Designphase Fehler auf der Ebene von Middleware, Schnittstellen und Protokollen gefunden und konnten mit wenig Aufwand behoben werden.
In Projekten B und C wurde dies in der Design-Phase eher vernachlässigt und da man in diesen Projekten eher auf die Codierung konzentriert war. Es mussten zeitaufwendige Code-Reviews durch eine Qualitätssicherung gemacht werden.
In Projekt D musste man durch die kleinen und klaren Implementationsschritte sowie ein gutes Qualitätsbewusstsein auf Entwicklungsseite keine grossen QS-Aktivitäten aufziehen. Es beschränkte sich durch einen gemeinsamen durchgeführten System-Test bei der Integration und danach den üblichen System-Test durch die QS-Abteilung des Unternehmens.

2.3.7 Dokumentation

In den Projekten A wurde die Dokumentation in angemessenem Umfang über alle Entwicklungsphasen gemacht. Projekt C war in der Anfangsphase (bis und mit Analyse) eher überschwenglich bis angemessen, hat aber in der Design- und Implementations-Phase die Dokumentation sträflich vernachlässigt. Projekt B war durchs Band nachlässig in Bezug auf Dokumentation. Projekt D konnte von der Verwendung des CASE-Tools profitieren, ansonsten war man vom Umfang her eher moderat.

2.3.8 Iterationshäufigkeit

Bei Projekt A war die Iterationshäufigkeit - also das Durchlaufen resp. Zurückkehren der einzelnen Phasen -

recht ausgeglichen. Bei Projekt C konnte eine starke Konzentration auf der Implementations und Testphase - nicht unerwartet - festgestellt werden. Im Projekt D hat es zwischen Design, Implementation und Test eine häufige Iteration gegeben, nicht zuletzt, da nicht alle Verhaltens-Details des bestehenden Systems, in der Analyse untersucht wurden.

3 Resultate

Projekt A ist heute erfolgreich eingeführt, gut dokumentiert, gut wartbar und sehr stabil.
Projekt B wurde vom Management suspendiert und durch ein eingekauftes Produkt ersetzt. Verluste durch Entwicklung und Einführungsverzögerung wird durch eine zweistellige Millionenzahl beziffert.
An Projekt C wird immer noch entwickelt und man verzeichnet bereits über ein Jahr Terminverzögerung. Die Wahrscheinlichkeit, dass das Produkt von Bedienerfreundlichkeit und Performanz her dem Benutzer zusagt, ist als gering einzustufen. Die Verluste bei Projekt-Misserfolg würde sich im gleichen Bereich wie Projekt B bewegen.
Projekt D ist eingeführt, gut wartbar und stabil. Auf dessen Basis wird weitermigriert und weiterentwickelt und es ist geplant, in 1-2 Jahren die alten Komponenten zu ersetzen, zu vereinfachen oder ganz zu eliminieren. Dank dem modularen Design wird eine Ablösung des Altsystems überhaupt erst möglich, ohne den Betrieb einzuschränken.

4 Zusammenfassung

Beim Einsatz der OO-Technologie kann die gewonnene Modularität so ausgenutzt werden, dass man die Arbeit auf Teams aufteilen kann. Dies erfordert aber ein sehr gut koordiniertes Team mit gutem Kommunikationsverhalten. Eine klare und weitsichtige Architektur und daraus entstandene einfache und vorallem verständliche Entwicklungsrichtlinien sowie eine gute Dokumentation sind Qualitäts-Merkmale auf dem Weg zum Erfolg.

4.1 Fazit

Folgende Schwerpunkte der gemachten Erfahrung können herausgenommen werden:

- Der Erfolg eines OO-Projektes (und anderer) ist fast ausschliesslich von den sozialen und fachlichen Qualität des Projekt-Managements und Mitarbeiter im Team abhängig.
- Ohne eine gute und durchgesetzte System- und Software-Architektur (und einem guten Architekten) ist ein OO-Projekte eher zum Scheitern verurteilt.
- Der grosse Vorteil des Gedankengutes der objektorientierten Entwicklung liegt vorallem in der Modularisierung, sofern diese auch konsequent auf die physikalische Ebene übertragen wird. Vererbung und Polymorphismus sind zwar hilfreiche Mechanismen, müssen aber trotzdem in angemessener Form eingesetzt werden.
- OO-Technologie setzt eine gute Projekt-Organisation bis hin zur Qualitätssicherung voraus, wird aber nicht unbedingt auf der Organisationsstruktur der bisherigen, konventionellen Entwicklung aufbauen können.
- Von der QS sind einige neue Anforderungen und viel Flexibilität gefordert.
- Qualitätssicherung ist eine Einstellung und keine Instanz, die vom frühesten Zeitpunkt des Projektes beginnt und mit viel Einfühlungsvermögen gepflegt werden muss.
- Zeitmangel darf nicht auf Kosten der Wartungsfreundlichkeit und auf Kosten der Dokumentation gehen.

Trotz aller technologischen Forschritte und neuen Methoden, steht der Mensch weiter und noch stärker im Mittelpunkt. Durch die anwachsende Mächtigkeit der verschiedenen Methoden und Tools bekommt jedes

Teammitglied einen grösseren Hebelarm, mit dem es positive zum Erfolg führende Aktionen erlangen kann. Anderseits aber kann der Fehler bei der Person sich ebenso stark auf den Misserfolg auswirken. Ueberlegen Sie bei der Wahl des Teams für ein terminkritisches, technologisch herausforderndes und das Image riskierende Projekt immer, ob Sie mit dem gleichen Team auch eine Expedition zum Nordpol wagen würden?