

UML: Schritt für Schritt

5/2005

Der Autor

Dipl.-Ing. Peter Siwon, geboren 1958, hat die Embedded Welt schon aus vielen Blickwinkeln betrachtet: In der Forschung, als Trainer, als Entwickler, als Projektleiter, in Vertrieb und Marketing. Heute ist er einer der Geschäftsführer der MicroConsult GmbH. MicroConsult begleitet seit vielen Jahren ihre Kunden in der Industrie mit Training, Coaching und Engineering für Hard- und Softwareentwickler. MicroConsult gehört zu den Pionieren, wenn es um den Einsatz der UML für Embedded Lösungen geht.

Peter Siwon, p.siwon@microconsult.com, 089/450617-44

Inhalt

UML: Schritt für Schritt	1
Der Autor	1
Inhalt	1
UML: Schritt für Schritt	2
Indizien: Der Wink mit dem Zaunpfahl	2
Orientierungsgespräch: Katze ohne Sack!	4
Analyse: Ursachenforschung	4
Konzept: Weg und Ziel	5
Basiskompetenz: A Fool with a Tool	6
Teil2: Die Projektpraxis	7
Toolauswahl: Schöne Bildchen	7
Das erste Projekt: Kernteam und Coach	8
Wissensmultiplikation: UML für alle	8
Zusammenfassung:	9
Tabelle: UML Schritt für Schritt	10

UML: Schritt für Schritt

Die entscheidende Frage für viele Entwicklungsteams ist heute nicht mehr, ob sie UML einsetzen werden. Vielmehr stehen folgende Fragen im Vordergrund: Wann ist der richtige Zeitpunkt für den Einstieg gekommen? und wie gelingt dieser Einstieg am erfolgreichsten? In diesem zweiteiligen Artikel habe ich die Erfahrungen, die wir in den letzten Jahren bei Trainings und Projekten gesammelt haben, zusammengefasst.

Teil 1) Voraussetzungen schaffen:

1. Indizien
2. Orientierungsgespräch
3. Analyse
4. Konzept
5. Basiskompetenz

Teil 2) Projektpraxis:

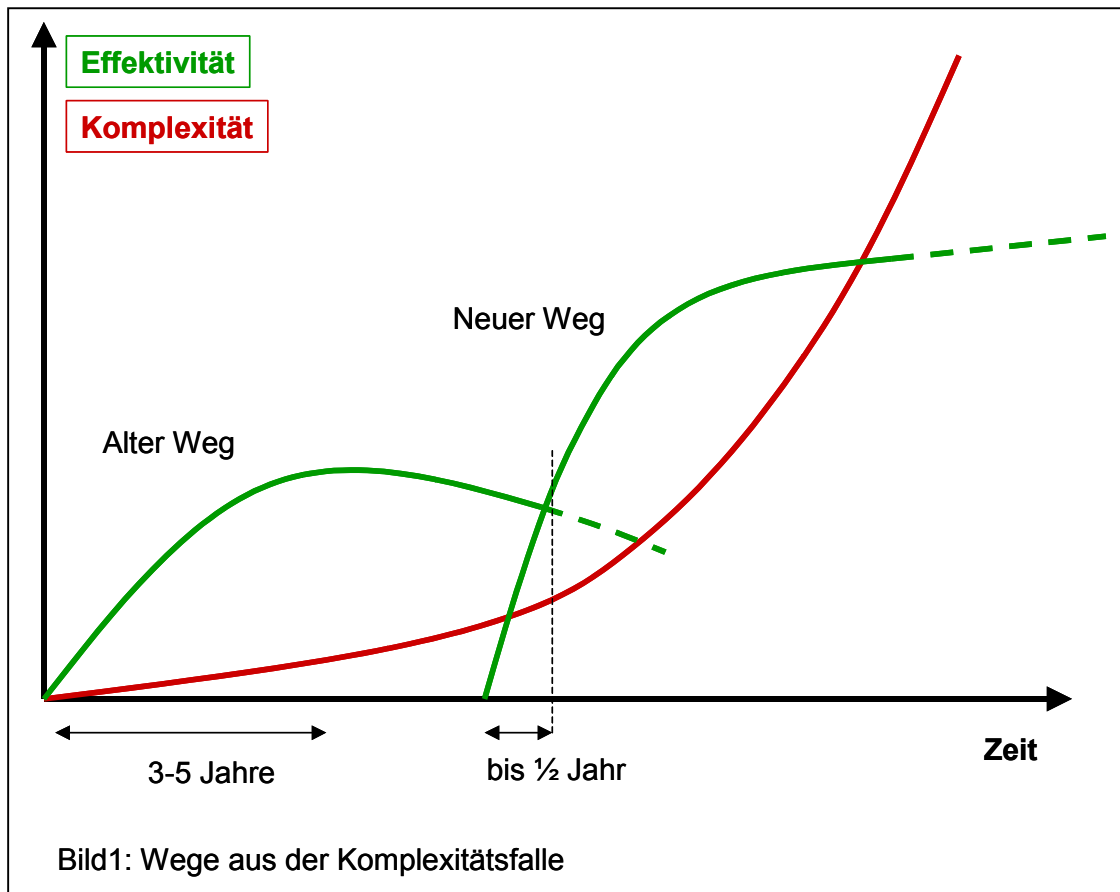
6. Toolauswahl
7. Das erste Projekt
8. Wissensmultiplikation
9. Zusammenfassung
10. Übersichtstabelle

Indizien: Der Wink mit dem Zaunpfahl

In vielen Fällen entstehen Schwachpunkte durch die gestiegene Komplexität unserer Entwicklungsaufgaben. Wir sind schleichend an die Grenzen bewährter Vorgehensweisen gestoßen und geraten mehr und mehr in Gefahr in der Komplexitätsfalle gefangen zu werden. Die Indizien, dass diese Grenzen erreicht wurden, zeigen sich zunächst sporadisch, dann allerdings immer deutlicher:

- Die Testabdeckung sinkt ab. Beim Kunden werden immer mehr Fehler entdeckt. Die Kosten steigen aufgrund von Rückrufaktionen, „Feuerwehreinsätzen“, etc.
- Das Verhältnis von real umgesetzter zu ursprünglich geplanter Funktionalität sinkt.
- Die Anzahl der Projekte, die durch massiven Ressourceneinsatz in der Endphase „gerettet“ werden, nimmt zu.
- Es kommt vermehrt zu Fehleinschätzung von Projektlaufzeiten und Kosten.
- Die Einarbeitungszeiten für neue Mitarbeiter steigen stark an.
- Es wird immer häufiger reagiert statt agiert.
- Der Code, den sich keiner mehr anpacken traut, wächst.
- Spezifikation, Code und Dokumentation weichen immer mehr voneinander ab.
- Immer mehr Hilfskonstruktionen, die ihre Vorgehensweisen erweitern oder ergänzen, werden eingesetzt (Bilder ersetzen Code).
- Der Zeitaufwand für Test und Korrektur steigt überproportional zum Entwicklungsaufwand.

Ein sinnvoller Bezugspunkt für diese Bewertungen sind Erfahrungen, die nach dem Einsatz einer Methode, von Tools oder eines Prozesses in der Softwareentwicklung nach etwa 3 Jahren vorliegen. Das ist in etwa der Zeitraum, der notwendig ist, um eine Entwicklungsumgebung optimal anzupassen und zu beherrschen.



Im Prinzip ganz einfach

Steigende Komplexität lässt sich im Prinzip durch drei Ansatzpunkte beherrschen:

- Verbesserung der Entwicklungs-/Arbeitsumgebung (Tools, Beseitigung von Störfaktoren)
- Optimierung des Ressourceneinsatzes (Prozess, Kommunikation)
- Verbesserung der Abstraktion (Methode, Darstellungsmittel)

Die UML liefert für alle diese Ansatzpunkte sinnvolle Lösungsbeiträge:

- Gute UML-Tools übernehmen lästige Detailaufgaben und machen Fehler früher sichtbar.
- Die UML unterstützt die Optimierung von Entwicklungsprozessen und erleichtert das Projektmanagement.
- Die hilfreichen Darstellungsformen der UML erlauben neue Sichten auf Systemarchitektur und -verhalten, die der Abstraktion auf Programmiersprachenebene (C, C++, Java, etc.) überlegen sind.

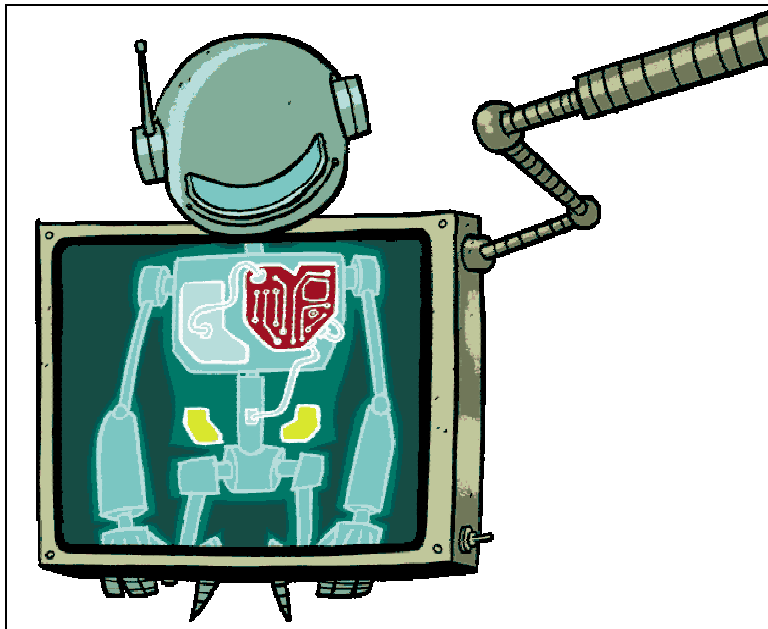
Wenn die Indizien auf die Vorteile der UML hindeuten, dann ist es Zeit, sich nach einem erfahrenen Partner umzusehen.

Orientierungsgespräch: Katze ohne Sack!

Dieser Partner sollte über mehrjährige Branchenerfahrung verfügen, damit er in der Lage ist, Ihre Projektanforderungen auf Tools, Methoden und Prozessvarianten zu projizieren. Die Sicht von Außen erhöht die Wahrscheinlichkeit, „blinde Flecken“ in der eigenen Wahrnehmung zu entdecken und zusätzliche Impulse für einen erfolgreichen neuen Weg zu erhalten. Ein Orientierungsgespräch von 2 bis 3 Stunden hilft beiden Seiten zudem festzustellen, inwieweit die menschlichen und fachlichen Voraussetzungen für eine Zusammenarbeit gegeben sind. Gerade hier gilt: Kaufe die Katze nicht im Sack.

Analyse: Ursachenforschung

Die Analyse hilft uns, die möglichen Ursachen für die Phänomene herauszufinden, auf die unsere Indizien hindeuten (Bild2: Analyse). Der Projektcoach (s. Kasten) führt diese Analyse gemeinsam mit einem Personenkreis durch, der repräsentativ für die Projektbeteiligten (Management, Projektleiter, Softwareentwickler, Systemarchitekt, etc.) ist. Je nach Betrachtungsschwerpunkt und Analyseziel wird der Personenkreis angepasst. Der Coach bringt sowohl ein sehr umfangreiches Fachwissen als auch ein gutes Fingerspitzengefühl für die zwischenmenschliche Brisanz einer solchen Analyse mit. Hier einige wichtige Blickwinkel:



Blick auf Technik und Organisation:

- Art, Ziel, Aufgaben
- Termine, Budget, Qualitätsanforderungen
- Typische Risiken und Abweichungen von Vorgaben
- Vorgegebener und gelebter Prozess
- Eingesetzte Methoden und Tools
- „Altlasten“ wie z.B. vorhandener Code, Kompatibilität
- Verfügbare Ressourcen (Personal, Geld, Zeit)

Blick auf die beteiligten Menschen

- Beteiligte Personen und ihre Rollen
- Wissen, Fähigkeiten, Erfahrungen der einzelnen Personen
- Wissen, Fähigkeiten, Erfahrungen des Teams
- Akzeptanz und verfügbare Potenziale für Veränderungen

Konzept: Weg und Ziel

Aus der Analyse werden gemeinsam die möglichen Ansätze für Verbesserungen abgeleitet und anschließend nach Aufwand und Wirkung bewertet.

Folgende Ansatzpunkte deuten auf Verbesserungen durch den Einsatz von UML-Werkzeugen hin:

- Arbeitsteilung
- Kommunikation
- Ergonomie
- Fehlererkennung
- Automatisierung
- Wiederverwendung
- Änderbarkeit
- Konsistenz Code/Dokumentation
- Projektmanagement

Es kann jedoch auch sein, dass die primären Ansatzpunkte für eine verbesserte Arbeitsteilung oder Kommunikation mehr in den Menschen und der Team- bzw. Unternehmenskultur liegen. Es ist also nützlich, zwischen primären und sekundären Ursachen zu unterscheiden. Die wichtigsten Grundfragen bei der Erstellung eines Konzepts sind:

- Dürfen: Welche Veränderungen sind (vom Management) erlaubt?
- Wollen: Welche Veränderungen sind (vom Entwicklerteam) erwünscht?
- Können: Welche Veränderungen sind (aufgrund der Ressourcen) realistisch machbar?

Wenn es hier im beteiligten Personenkreis unterschiedliche Meinungen gibt, ist der erste und wichtigste Schritt, einen Konsens zu finden. Hier ist der Nutzen eines externen Moderators nicht zu unterschätzen. Maßnahmen zur Herbeiführung dieses Konsens sind bereits Teil des Konzeptes.

Veränderungsprozesse sind je nach Umfang Teilprojekte innerhalb von Projekten oder übergeordnete Projekte. Deshalb spiegelt das Konzept in seiner Darstellung die Logik einer Projektplanung wider: Ziele, Maßnahmen, Termine, Kosten, Meilensteine, Qualitätsmerkmale, Kosten-/Nutzenanalyse. In vielen Fällen erfolgt die Durchführung der Analyse und der Erstellung des Konzepts bereits mit Personen aus dem Kernteam für die Einführung der UML.

Auswahl der Kernteams

Die Auswahl des Teams orientiert sich an fachlichen, rollenspezifischen und emotionalen Faktoren. Es sollten folgende Rollen (eine Person kann auch mehrere Rollen haben!) besetzt sein:

- der engagierte Pate, der das Projekt im Management vertritt
- der nüchterne Controller, der Kosten, Zeit und Qualität im Auge behält
- der kontaktfreudige Kommunikator, der dafür sorgt, dass die Informationen verteilt werden
- die engagierten Umsetzer, die das Projekt realisieren

Das Kernteam kennt die Ziele und ist von der Durchführbarkeit und dem Nutzen überzeugt. Die Kollegen sind über die Bedeutung dieser Aufgabe informiert und das Management hat seine Unterstützung bekräftigt. „U-Boote“ haben bei solch wichtigen Themen wenig Sinn, weil die Gefahr des Scheiterns aufgrund zu geringer Ressourcen und mangelnder Unterstützung zu groß ist. Die Kommunikation zwischen Management und Entwicklung spielt eine wichtige Rolle.

Basiskompetenz: A Fool with a Tool

Zunächst braucht das Team ein ausreichendes Basiswissen zum Thema UML. Leider wird nicht selten viel Geld in ein Tool investiert und bei der Ausbildung gespart. Bei der UML gilt der schöne Spruch „A Fool with a Tool is still a Fool“ ganz besonders. Es ist empfehlenswert, ein Methodentraining eines erfahrenen Anbieters zu besuchen. Achten Sie darauf, dass er Ihrer Branche nahe steht, damit die Beispiele zu Ihrer Gedankenwelt passen. Auch sollten Sie unbedingt sicherstellen, dass die Methode im Vordergrund steht und nicht die Toolbedienung. Typischerweise dauert ein Training mit praktischen Übungen für einen brauchbaren Lernerfolg je nach Vorkenntnissen 3-5 Tage. Danach besteht eine gute gemeinsame Kommunikationsbasis für die weiteren Schritte.

Seien Sie sich bewusst, dass dieses Basistraining so etwas wie der Anfängerhügel beim Skifahren ist. Sie wissen jetzt - im Prinzip - wie es geht und können ein paar Stembögen fahren. Zum eleganten Slalom durch die steilen Softwarepisten sind einige Wochen Praxis notwendig. Es ist sehr vorteilhaft, wenigstens einen Mitarbeiter im Team zu haben, der schon über fundierte Erfahrung mit der UML verfügt. Dies kann auch ein externer Mitarbeiter sein. Auch die Kombination der Anwendungskompetenz „alter Hasen“ und der Methodenkompetenz von Hochschulabsolventen, bietet für beide Seiten gute Entwicklungsmöglichkeiten, sofern die Teamkultur das zulässt.

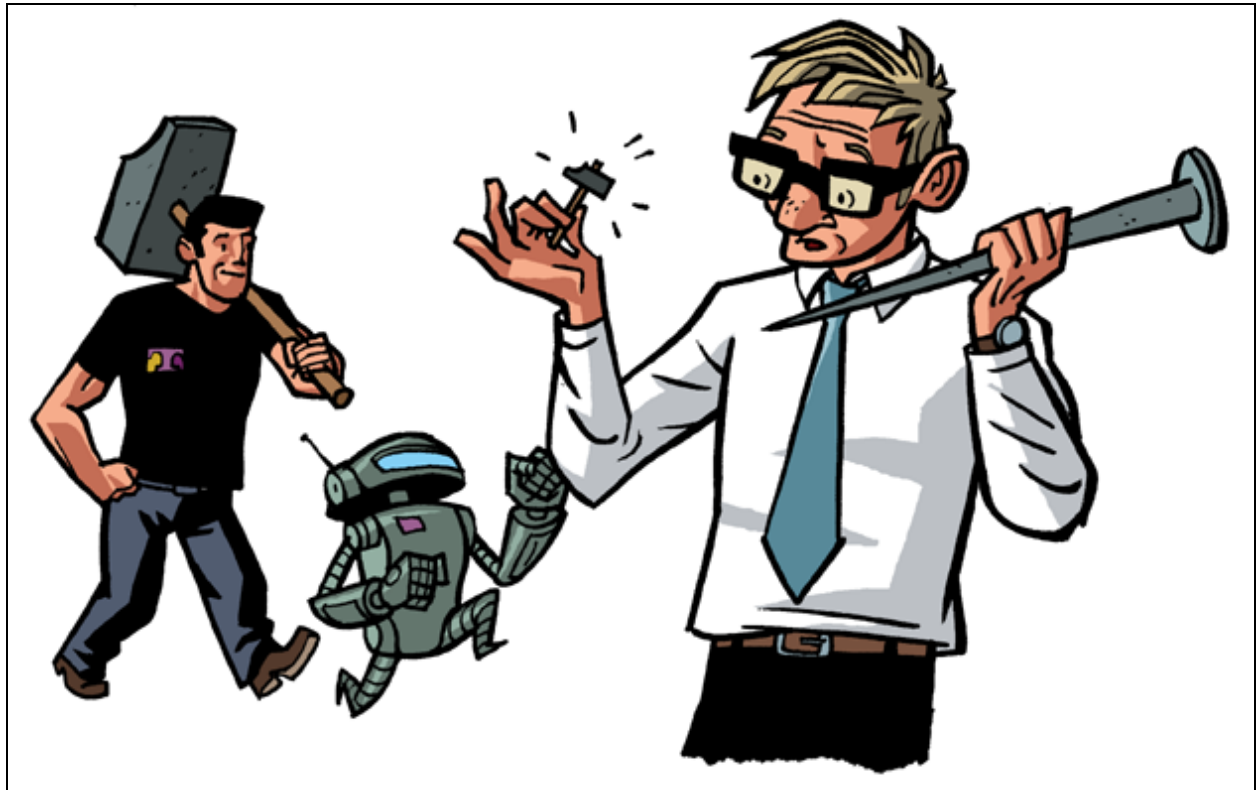
Die Ergebnisse der Analyse bieten auch eine hervorragende Grundlage, um das Basiswissen auf die Bedürfnisse des Projektteams zuzuschneiden. Statt eines 0815-Trainings kann beispielsweise eine mehrstufige Ausbildung abhängig von Wissensstand und Rolle erfolgen. Gleichzeitig ist diese Ausbildung die erste Gelegenheit für das Team, die Projektanforderungen und die Möglichkeiten der UML gedanklich zu verknüpfen.



Teil2: Die Projektpraxis

Toolauswahl: Schöne Bildchen

Mit diesem Wissensfundament steigen wir nun etwas gelassener, aber in jedem Falle mit mehr Selbstvertrauen in die Toolauswahl ein. Oft fehlen Ihnen allerdings die Anwendungserfahrungen von Tools dieser Art in Ihrem Umfeld. Deshalb ist auch hier der Einsatz eines erfahrenen Beraters oder Coaches empfehlenswert.



Wie so oft zeigen sich die wahren Stärken - aber auch Tücken - eines Tools erst bei längerer Anwendung. Eine Demo ist zwar sinnvoll, aber aus Gründen der Praktikabilität auf einfache Beispiele beschränkt. Die Ergebnisse der Analyse und die ersten Erfahrungen im UML-Training sind sehr hilfreich um die Ansprüche an das Tool zu definieren. Lassen Sie sich nicht von schönen Oberflächen, netten Gimmicks oder eindrucksvollen Referenzen von Ihren Prioritäten abbringen. Es geht um mehr als nur darum, ein paar schöne Bildchen zu zeichnen! Falls Sie niemanden im Team haben, der hier über praktische Erfahrung verfügt, dann suchen Sie jemanden. Vielleicht kann Ihnen ja auch der Hersteller einen Kontakt zu einem Kunden vermitteln, der ähnliche Anforderungen bereits in der Praxis umgesetzt hat. Aufschlussreich ist auch der Besuch einer User Group, um aktive Anwender bezüglich ihrer Erfahrungen zu befragen. In jedem Falle sollten die Personen, die das Tool evaluieren, vorher eine Toolschulung besuchen, um eine realistische Vorstellung der Möglichkeiten zu haben. Die Anzahl der Tools im Embedded Bereich hält sich in Grenzen und eine Vorauswahl lässt sich mithilfe eines erfahrenen Beraters treffen. Daher ist es weniger riskant, eine Schulung zu besuchen, selbst wenn man dann feststellt, dass das gewählte Tool ungeeignet ist. Andererseits ist eine „unnötige Schulung“ immer noch billiger als der Kauf eines falschen Tools. Also: Erst wissen, dann wählen!

In der Regel stellen Ihnen die Hersteller das Tool für eine Evaluierung kostenlos zur Verfügung. Dank Ihrer Analyse sind Sie nun in der Lage, sehr zielstrebig die Dinge auszuprobieren, auf die es ankommt. Stellen Sie dabei auch gleich die Kompetenz und Verfügbarkeit des Toolsupports auf die Probe. Die Toolevaluierung ist auch gleichzeitig das erste Teilprojekt, das Sie auf Ihrem Weg zum UML-Profi ein Stück weiterbringt.

Das erste Projekt: Kernteam und Coach

Jetzt geht's los! Das erste Projekt mit UML und dem ausgewähltem Tool wird umgesetzt. Genau genommen handelt es sich zunächst um ein weiteres Teilprojekt, das Sie ausgewählt haben. Bei der Auswahl der ersten Aufgaben ist es sinnvoll, darauf zu achten, dass das Einüben der Methode und des Tools nicht von weiteren Hürden behindert wird, beispielsweise durch eine völlig neue Technologie oder politisch brisante Projektelemente. Wenn die Aufgabe die Möglichkeit bietet, den gesamten Entwicklungsprozess von der Anforderungsanalyse bis zum Test zu durchlaufen, bietet sie ideale Voraussetzungen. Eine zentrale Rolle in dieser Phase spielt der sogenannte Projektcoach.

Seine Aufgabe ist es, das Kernteam möglichst schnell zur eigenständigen routinierten Umsetzung der Ziele zu entwickeln. Wenn auf diese Weise nun mehr Routine in dem Kernteam vorliegt, können im nächsten Schritt kritische Pfade - z.B. bezüglich Echtzeitverhalten - angepackt werden.

So beweist sich schnell in der Praxis, welche Erfahrungen und welches Wissen für die Teamkollegen am hilfreichsten ist.

Kasten: Der Projektcoach (Bildvorschlag: Coach)

Der Coach braucht die Fähigkeit loszulassen und den Beteiligten fast unbemerkt als Richtungsweiser zu wirken, ohne konkrete Schritte vorzugeben. Die meisten rein technisch orientierten Berater haben hierbei ihre liebe Not. In diesem Schritt ist das große Ziel, das gemeinsame Ergebnis im Projekt so zu erreichen, dass die Lorbeeren beim Projektteam bleiben und das gesamte Team dank neuer, eigener Erfahrungen kompetenter wird. Etwas aus seiner Sicht zu wissen und trotzdem einen eigenen Entwicklungsprozess zuzulassen - das ist die Herausforderung für die Profis unter den Projekt-Coaches. Bei entsprechenden Kompetenzen ist die parallele Ausbildung von Projektleitern bzw. Führungskräften zu Projekt-Coaches denkbar und in größeren Unternehmen sogar sinnvoll.

Er hilft dem Kernteam dabei, Basiswissen schneller in routiniertes Praxiswissen zu verwandeln. Dabei nimmt der Coach mehrere Funktionen und Rollen ein:

- Als Erfahrungsträger stellt er bereits beim Projektstart die Zusammenhänge zu weiteren Projektphasen her.
- Als Wächter über Methode und Prozess erinnert er immer wieder an Regeln und Konventionen und hilft bei der Interpretation. Er macht Ziel und Situation immer wieder bewusst.
- Als Diskussionspartner und Moderator unterstützt er die Entwicklung von Lösungsideen.
- Als Trainer füllt er die eine oder andere Wissenslücke und fördert den Erfahrungsaustausch im Team.

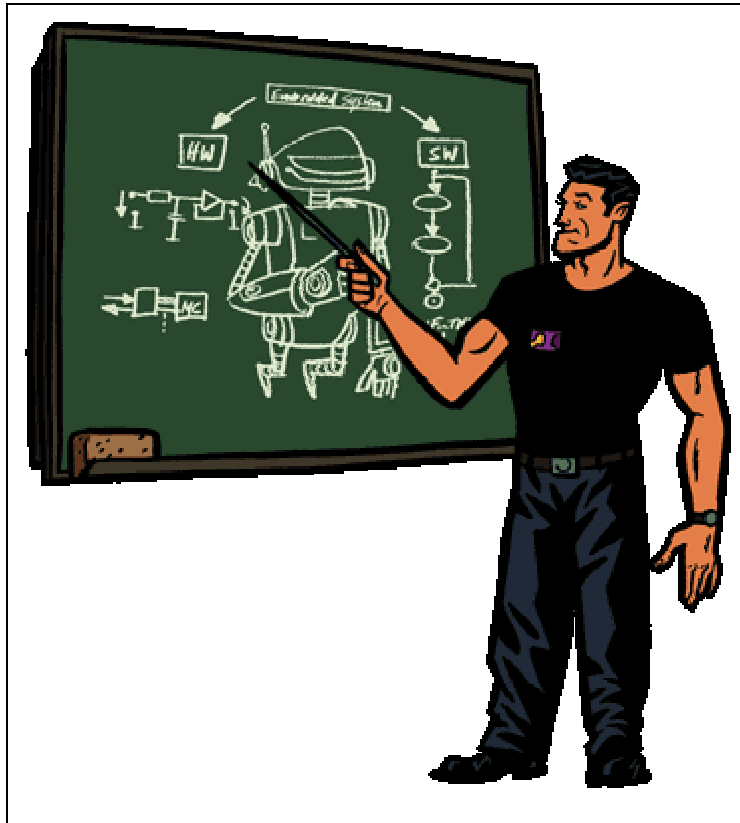
Wissensmultiplikation: UML für alle

Eine bewährte Form, weitere Entwickler ins Boot zu holen, ist der sogenannte Projektworkshop. Das Kernteam erarbeitet zusammen mit einem erfahrenen Trainer einen Workshop, in dem alle anderen Beteiligten die Erfahrungen des Kernteams in sehr kompakter Form in der Praxis erleben können. Auf diese Weise werden außerdem alle Erfahrungen und Erkenntnisse bewusst gemacht, bewertet und dokumentiert.

Bevor alle anderen Beteiligten diesen Projektworkshop besuchen, ist wie beim Kernteam eine fundierte Basisausbildung erforderlich. Natürlich lassen sich Basisausbildung und Projektworkshop auch sehr gut kombinieren.

Warum dieser Aufwand? Es gibt zu viele Softwareregelerwerke, die geduldig im Regal verstauben. Die beste Garantie für eine praktische Anwendung ist das Lernen unter Praxisbedingungen. Außerdem ist damit das Argument von „der grauen Theorie irgendwelcher Schlauberger, die keine Ahnung von der Praxis haben,“ wirkungsvoll widerlegt.

Anschließend werden die Projekte nach der neuen Vorgehensweise umgesetzt. Die Mitglieder des Kernteams können dabei als Coaches unterstützen.



Zusammenfassung:

Es steht außer Frage, dass Veränderungen ihren Preis haben. Der Umstieg auf UML ist von hoher strategischer Bedeutung für ein Unternehmen, weil die Einflüsse auf vielen Ebenen spürbar werden. Die systematische Analyse der Ist-Situation und der angestrebten Ziele ist der Ausgangspunkt für eine bewusste Entscheidung und für die Wahl der Maßnahmen. Der fundierte Aufbau von Wissen zu Methode und Tool erhöht die Sicherheit für eine erfolgreiche Einführung in die Projektarbeit. Beim Einstieg in die Projekte ist die Zusammenarbeit mit einem erfahrenen Projektcoach von großem Nutzen, weil dadurch die noch fehlende Routine ausgeglichen und der Lernprozess beschleunigt wird. Die Weitergabe der ersten Projekterfahrungen im Zuge eines Projektworkshops ist die effektivste Methode, um die gewonnenen Erfahrungen im Unternehmen zu vervielfachen. In der folgenden Tabelle sind alle Punkte nochmals zusammengefasst.

Tabelle: UML Schritt für Schritt

Schritt	Situation	Maßnahmen	Ziel	min. Zeit in Tagen	Investition typisch (in k€)	min. Investition in externe Unterstützung
Indizien sammeln	Vermehrte Abweichung von Zielen und Ergebnissen	Indizien sammeln und anhand der Projektergebnisse bewerten	Erkennen von Ansatzpunkten notwendiger Veränderungen Bereitschaft für Veränderungsprozess	Teil der Projekte		
Orientierung	Änderung erforderlich; Unklarheit über Ansatzpunkte	Orientierungsgespräch mit einem (externen) Berater	Auswahl eines geeigneten Partners, erste grobe Einschätzung weiterer Maßnahmen, Vertrauen zu Partnern aufbauen	0,5 für Team		kostenfrei
Analyse	Erste Ansatzpunkte sind bekannt	Analyse des Umfeldes; Tools, Methoden, Prozess, Mensch	Konkrete Ansatzpunkte für Maßnahmen, Ziele, Nutzen einer Veränderung, Vertrauen zu Partnern vertiefen	1 für Team mit Projektbegleiter		1500
Konzept	Ansatzpunkte sind konkretisiert	Festlegung der Vorgehensweise; Auswahl Kernteam	Maßnahmen, Termine, Kosten/Nutzen, Rollen	1 für Team mit Projektbegleiter		1500
Basisausbildung	Wissen zu UML fehlt; C++/C-Erfahrung	Training	Grundlagen sind bekannt, erste praktische Erfahrungen, Selbstvertrauen aufbauen	3 pro Person	1500	
Toolauswahl	Anforderungen bekannt	Toolschulung Evaluierung, Schritt 1	Toolentscheidung 1. Projekterfahrung	3 pro Person 10 für Team	1500	3000
1. Projekt mit Kernteam	Basiswissen vorhanden Toolentscheidung zu 70% getroffen	Erste Projekterfahrungen	Erfahrungen sammeln, Anpassung von Tool, Methode und Prozess	10 für Team		3000
	Lizenzbeschaffung	Toolentscheidung	Für Team mit 5 Entwicklern	Pro Entwickler	5000	
Einstieg weiterer Mitarbeiter	Projekt mit Kernteam abgeschlossen	Erfahrungsbericht, Entwicklung Projektworkshop, Basisausbildung (UML, Tool), Projektworkshop	Einführung von Tool, Methode, Prozess im gesamten Team	5 für Kernteam 9 pro Person	Ergebnis der Projektarbeit 3000	nach Bedarf