

Typische Projektlügen, Teil 2

Warum Projektplanung scheitert

Nachdem wir uns im ersten Teil unserer Reihe mit falschen Herangehensweisen bei Projektmanagementprozessen beschäftigt haben, werfen wir diesmal einen Blick auf die oft falsch interpretierte Rolle des Projektleiters.



Ralf C. Adam

arbeitet als freier Producer, Autor und Game Design Consultant.

Ralf ist seit 14 Jahren in der Spielebranche tätig und hat in dieser Zeit als Producer und Projektleiter für Firmen wie Sunflowers, Infogrames/Atari, JoWood und Phenomic gearbeitet. Seit nunmehr dreieinhalb Jahren bietet er seine Dienste als »Gun for hire« an und berät dabei sowohl Entwicklerteams auf der einen als auch Publisher auf der anderen Seite - aktuell unter anderem Atari, Blue Byte, dtp und GameForge. ralf@tigerteam-productions.de

Wenn man in einem durchschnittlichen Spiele-Entwicklerteam die Frage stellt, wer für die Leitung des Projekts die Verantwortung trägt, ertönt als Antwort meist im Chor »Na, der Projektleiter« (wenn ein solcher denn überhaupt im Team existiert)! Hierzu gibt es einen schönen Spruch aus dem klassischen Projektmanagement: »Wer glaubt, dass ein Projektleiter Projekte leitet, der glaubt auch, dass ein Zitronenfalter Zitronen faltet«. Was auf den ersten Blick nämlich wie die logische und wichtigste Aufgabe eines Projektleiters aussieht, führt in Wahrheit bei den Mitarbeitern zu einem Zustand, den die Psychologie »Anerzogene Hilflosigkeit« nennt. Typische Anzeichen dafür sind Teammitglieder, die...

- ... bei auftretenden Problemen die Arme verschränken und sich passiv verhalten.
- ... nur am Nörgeln und Jammern sind.
- ... Konflikte nicht lösen, sondern aussitzen.
- ... nicht über den Tellerrand schauen, sondern sich nur um ihren eigenen kleinen Bereich kümmern (»Was geht mich fremdes Elend an?«).
- ... ihre Versprechungen (und damit auch die Zeiteinschätzungen) nicht halten.

■ ... mit jedem kleinen Problem zum Projektleiter rennen, damit er das bitteschön für sie lösen soll.

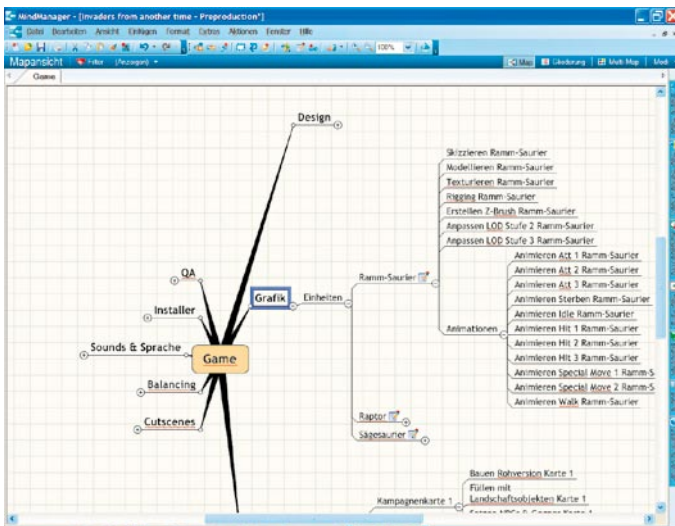
Anders formuliert: Ein Projektleiter, der jedem im Team sagt, was der zu tun und zu lassen hat, nimmt die Mitarbeiter aus der Verantwortung - was dazu führt, dass diese schlicht das Denken einstellen und automatisch im wahrsten Sinne des Wortes »verantwortungslos« handeln. Denn die Aufgabe des Projektleiters ist es eben nicht...

- ... für die Teammitglieder zu denken.
- ... jedem Mitarbeiter seine Arbeit nachzutragen.
- ... die Methoden und Tools vorzugeben.

Statt auf jeden Aspekt des Projekts die richtigen Antworten zu geben, muss der Projektleiter vielmehr die richtigen Fragen stellen. Denn nur wenn die Mitarbeiter selbst Verantwortung übernehmen, werden sie auch voll hinter getroffenen Aussagen und Vereinbarungen stehen. Das gilt für nahezu alle Projektarbeiten, egal in welcher Branche. In der Spieleentwicklung gibt es jedoch ein paar besondere Aspekte zu beachten.

Do it the SMART way

Was wird wohl ein Tool-Coder denken, wenn er am Montagmorgen eine Task-Liste mit Zeiten



Mindjets MindManager ist ein hervorragendes Tool, um Tasks zu sortieren. Per Knopfdruck wird die MindMap nach MS-Project exportiert (hier direkt nach dem Export).

von seinem Projektleiter in die Hand gedrückt bekommt, die nicht er, sondern der Chef (oder wahlweise der Lead-Coder) für ihn geschätzt hat. Und was wird derselbe Coder denken, wenn er am Freitagnachmittag die Arbeiten nicht fertig gestellt hat? Ganz einfach: Es wird ihn nicht weiter interessieren! Er wird sich sagen: »Ich wusste von Anfang an, dass diese Zeiten unrealistisch sind, aber mich hat ja keiner gefragt«. Und es wird ihm auch egal sein, dass die Level-Designer jetzt Überstunden schieben müssen, weil die wichtige Funktion im Editor nicht fertig geworden ist - eben weil er gar nicht wusste, dass die Content-Abteilung dringend darauf gewartet hat.

Deshalb ist es so wichtig, dass ein Projektleiter immer die richtigen Fragen stellt, er die Antworten darauf aber die betroffenen Teammitgliedern selbst geben lässt. Bei der Taskplanung lauten diese Fragen:

Wer (Ausführender) macht...

- ... was (Aufgabe, Arbeitspaket)
- ... bis wann (Abgabetermin)
- ... mit welchem wie gemessenem Ergebnis (Ziel- und Erfolgskriterien)
- ... wozu (wird diese Aufgabe später gebraucht)

Dabei handelt es sich um die so genannte S.M.A.R.T.-Zielformulierung, wobei die Abkürzung für specific (genau, exakt beschrieben?), measurable (messbar?), attainable (erreichbar?), relevant (Ziel auch wirklich wichtig?) und timed (wann?) steht. In der Spiele-Entwicklung erlebe ich es sehr häufig, dass nur die ersten drei Fragen gestellt werden, die Definition der Qualitätsziele und -messbarkeit sowie der Zusammenhänge (wer soll später mit dem Ergebnis weiterarbeiten?) aber meist hinten runterfallen.

Cpt. Picard vs. Cpt. Kirk

Natürlich ist es wichtig, dass der Projektleiter die Antworten auf diese Fragen kennt - wie soll er sonst überprüfen, ob der Mitarbeiter die gestellte Aufgabe auch richtig verstanden und nichts übersehen hat. Wie aber kann ein Projektleiter zum Beispiel im Detail die Glaubhaftigkeit einer Zeiteinschätzung eines Engine-Coders zu einem technischen Programmierauftrag überprüfen? Einfache Antwort: gar nicht. Hierfür gibt es schließlich die jeweiligen Abteilungs-Leads, die beim Abschätzen der einzelnen Tasks mit am Tisch sitzen müssen. Und auch für sie gilt die Projektleiter-Regel: Wer fragt, führt! Auch sie sollen also nicht die Zeiteinschätzungen vorgeben, sondern diese hinterfragen, falls sie verdächtig hoch oder auch zu niedrig gegriffen erscheinen. Grundsätzlich empfiehlt es sich deshalb immer, einen Mitarbeiter eine gestellte Aufgabe noch einmal in seinen eigenen Worten zusammenfassen zu lassen, um festzustellen, ob er die Taskbeschreibung so verstanden hat, wie sie gemeint war. Oftmals kann es sogar passieren, dass dem Mitarbeiter noch Teilaspekte auffallen, die selbst dem Projektleiter oder dem entsprechenden Lead bislang entgangen waren. Wir sehen also: Es ist ein fataler Denkfehler,



Coder-Meeting: Gemeinsam werden die einzelnen Tasks und die Arbeitsaufwände besprochen.

dass ein starker Projektleiter gleichbedeutend für ein starkes Team ist – tatsächlich verhält es sich eher genau umgekehrt. Was uns zu der Preisfrage führt: Wer wäre der bessere Projektleiter: Cpt. Jean-Luc Picard (»Machen sie es so!«) oder Cpt. James T. Kirk (»Phaserkanonen feuerbereit machen! Auf mein Kommando!«)...?

Warum brauchen wir immer doppelt so lang wie geplant?

Obwohl die meisten Teams ihre Mitarbeiter inzwischen bei der Taskplanung mit in die Verantwortung nehmen, jammern sie dennoch, dass ihre Zeitpläne nach wie vor gesprengt werden, dass die Budgets sowieso grundsätzlich zu knapp sind und dass man die Deadlines (wieder einmal) zu eng kalkuliert hat. Interessanterweise ist es dabei übrigens egal, ob es sich um eine 200.000-Dollar- oder eine 20-Mio.-Dollar-Entwicklung handelt – die Beschwerden kommen immer, und die Aussagen gleichen sich dabei wie ein Ei dem anderen. Wenn also Budget und Ressourcen augenscheinlich nicht der massgebliche Faktor sind, bleibt die Frage, warum gerade in der Spielebranche anscheinend der Zeitaufwand regelmäßig so gnadenlos unterschätzt (bzw. falsch eingeschätzt) wird?

Hierfür lassen sich drei Gründe ausmachen:

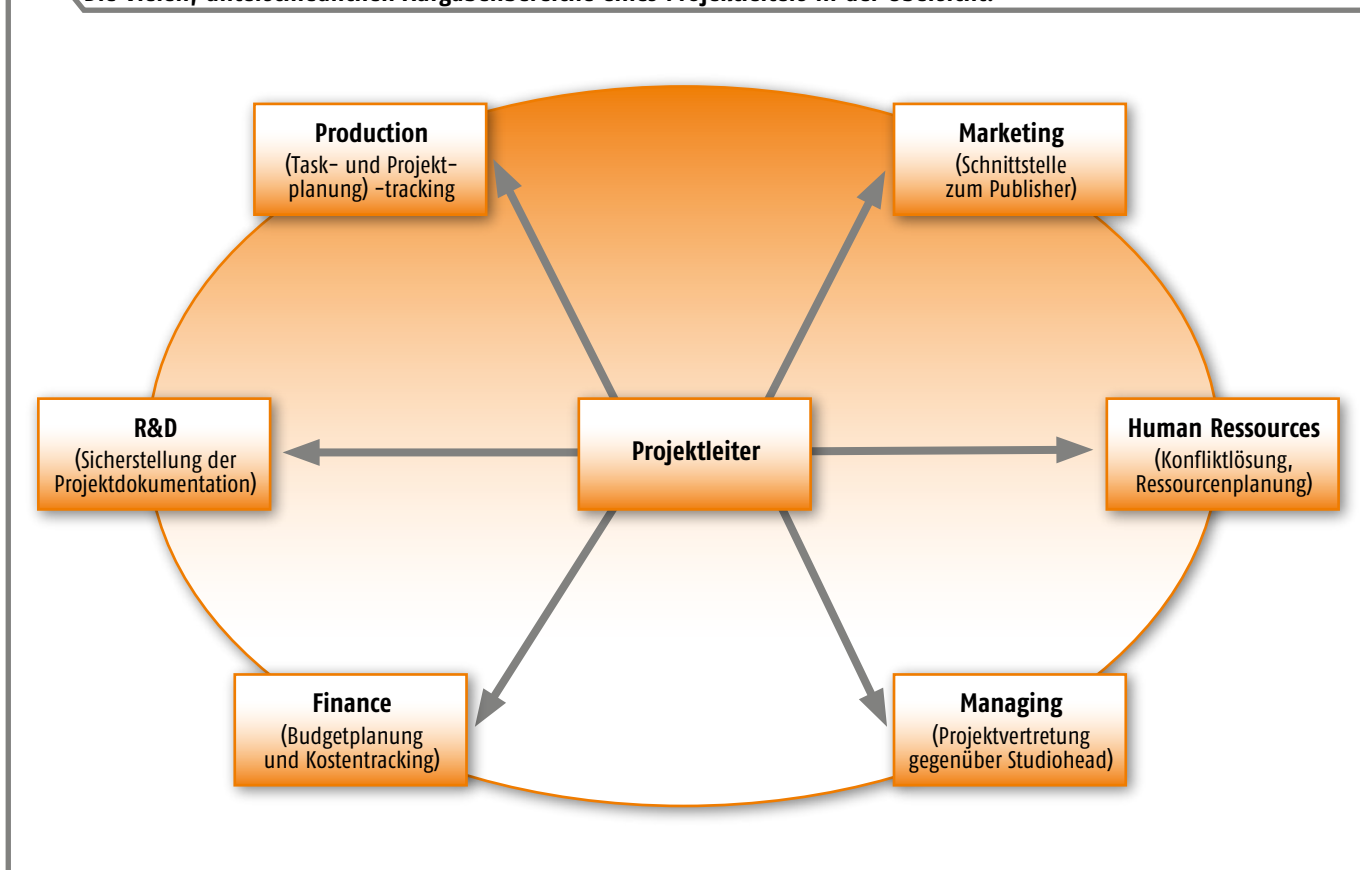
- Die Aufwandsschätzungen sind zu grob.
- Statt der Arbeitsdauer wird ausschließlich der Aufwand geschätzt.
- Es wird mit der falschen Methode geschätzt.

Zu grobe Zeiteinschätzungen sind relativ offensichtlich – und kommen trotzdem nach wie vor sehr häufig vor. Statt echter Tasks – im ursprünglichsten Sinne ihres Wortes und ihrer Definition (Task = Teil eines Projekts, der nicht mehr weiter heruntergebrochen werden kann) – werden Komponenten oder Projekt-Teile geschätzt. Dass aber die Komponente »Wegfindung« mit einer Aussage des verantwortlichen Coders à la »Brauche ich so ungefähr drei Monate für!« nicht hinlänglich und genau genug erfasst sein dürfte, versteht sich eigentlich von

Best Practise Projektplanung

Zu einem Task in einem Projektplan (z.B. in MS-Project) gehört immer ein beschreibendes Verb! Beispiel: Die Bezeichnung »Ork-Krieger 1« unter dem Oberbegriff »Grafikerstellung« ist nicht aussagekräftig genug. Was bedeutet »Ork-Krieger 1«? Die Skizzenerstellung? Die Modellierung? Das Rigging? Die bessere Alternative ist deshalb »Texturieren Ork-Krieger 1« oder »Erstellen Grundtextur Ork-Krieger 1« bzw. »Erstellen Texturvariante Drachenclan für Ork-Krieger 1«.

Die vielen, unterschiedlichen Aufgabenbereiche eines Projektleiters in der Übersicht.



alleine. Ähnlich verhält es sich mit einer Grafiker-Aussage im Stil von »Für die 30 Models brauche ich knapp 60 Tage!«. Bei beiden Definitionen fehlen überdies komplett die messbaren Aussagen hinsichtlich der Ziel- und Qualitätskriterien. Was wird die Wegfindung denn alles beinhalten (und was nicht)? Und wie verhält es sich mit den 30 Modellen des Grafikers – sind die mit Textur oder ohne? Bereits geriggt? Mit Animationen? Mit Normal-Maps? Gibt es LOD-Stufen?

Dauer ist nicht gleich Aufwand

Ein weiterer gern gemachter Fehler bei der Zeitabschätzung ist die Annahme, dass ein Mitarbeiter acht Stunden täglich fünf Tage die Woche nur vor seinem Task sitzt. Der Tool-Coder, der für das eine, spezifische Editor-Feature fünf Arbeitstage veranschlagt, hat an sich durchaus Recht: Er wird daran ungefähr 40 Arbeitsstunden programmieren. Er wird allerdings auch telefonieren, Zeit in Meetings verbringen, Bugs an zuvor erledigten Arbeitspaketen fixen, einen neuen Mitarbeiter einweisen, sich mit dem zweiten Tool-Coder über ein anderes Editor-Feature abstimmen sowie Research betreiben. Aus dem von ihm geschätzten Aufwand von fünf Arbeitstagen wird dann am Ende eine effektive Arbeitsdauer von sechseinhalb oder gar sieben Arbeitstagen. Die Faustregel lautet, dass die Arbeitsdauer normalerweise im Schnitt Arbeitsaufwand plus 30% beträgt. Hinzu kommt die oftmals in der Planung ebenfalls

unterschlagene dumme Angewohnheit von Mitarbeitern, hin und wieder auch mal Urlaub zu nehmen oder gar krank zu werden – laut statistischem Bundesamt betrug die Anzahl der durchschnittlichen Krankheitstage in Deutschland im Jahr 2005 immerhin 11,5 Tage. Summiert man diese Zahlen und Faktoren für alle Tasks entsprechend auf, wird man sehr schnell feststellen, dass man mit einer reinen Abschätzung von Arbeitsaufwänden bei der Planung sich selbige auch schenken kann, da sie nichts mehr mit der Realität zu tun haben wird.

Pi mal Daumen hat noch nie funktioniert

Nicht nur die Dauer, auch der Aufwand selbst werden oft falsch eingeschätzt. Sogar mit 30%-igem Puffer blasen sich dann fünf projektierte Arbeitstage zu stolzen 15 auf. Woran liegt das? Auch wenn die Über-den-Daumen-Methode nach wie vor die am weitesten verbreitete Schätzmethode ist, die beste und zuverlässigste ist sie mit Sicherheit nicht. Eine in der IT-Branche gängige Art, Zeiten zu schätzen, ist die Vergleichsmethode. Hier werden durch Vergabe von Kennzahlen die Erfahrungswerte aus vergangenen Projekten herangezogen. Dummerweise mögen sich zwar bei der SAP-Datenbankprogrammierung Arbeitspakete in Projekten wiederholen, bei der Spiele-Entwicklung ist das allerdings eher selten der Fall. Da kann die Schätzklausur schon lohnenswerter sein – hier

Dreifach-Schätzung

Best Case - Einschätzung: 10 Tage

Most Likley - Einschätzung: 15 Tage

Worst Case - Einschätzung: 25 Tage

Formel zur Bestimmung eines realistischen Durchschnittswertes (basierend auf den Erfahrungen aus der Teststatistik, dass eine Aufwandsschätzung in mehr als 85% der Fälle in Richtung Worst-Case abweicht):

$$(2 \times \text{Best Case}) + (3 \times \text{Worst Case}) + \text{Most Likely} = X / 6$$

In diesem Fall:

$$(2 \times 10) + (3 \times 25) + 15 = 110 / 6 = 18,33 \text{ Tage}$$

Quelle:

»Dan Irish: The Game Producer's Handbook«

sitzen dann zum Beispiel alle Coder am Tisch und lassen ihre jeweiligen Erfahrungswerte beim Abschätzen jedes Tasks mit einfließen (BioWare hat laut Vortrag auf der diesjährigen GDC daraus inzwischen schon eine Art internes, pokerähnliches Spiel entwickelt). Die Aufwandsabschätzung wird auf diese Weise wesentlich genauer, auf der anderen Seite ist diese Art der Planung natürlich auch entsprechend aufwändiger und zeitintensiver.

Die letzte Möglichkeit besteht in der sogenannten Dreifach-Schätzung. Bei dieser überlegt jeder Mitarbeiter bei einem Task wie lange er a) idealerweise dafür brauchen wird (best case), b) normalerweise (most likely) und c) im schlimmsten Fall (worst case). Mit diesen drei Werten lassen sich mit Hilfe einer einfachen Formel (siehe Extra-Kasten) relativ verlässliche Durchschnittswerte ermitteln. Genauer werden diese Werte sogar noch, wenn Sie über einige Monate hinweg beobachten, wie gut jeder einzelne Mitarbeiter sich selbst einschätzen kann. Dann können Sie nämlich nach einiger Zeit individuell für jeden Mitarbeiter die Formel anpassen, je nachdem wie stark seine Schätzungen (und in welche Richtung) im Schnitt vom Most-Likely-Fall abweichen.

Zeitschätzung ist Zeitverschwendung!?

Zum Abschluss dieses Artikels würde ich gerne noch mit zwei weit verbreiteten Vorurteilen im Zusammenhang von Aufwandsschätzungen bei Spieleprojekten aufräumen. Das eine Vorurteil besagt, dass Mitarbeiter grundsätzlich eigene Zeiteinschätzungen niemals unterbieten. Wenn jemand also fünf Tage für einen Task schätzt, dann benötigt er auch mindestens fünf, selbst wenn er es eigentlich in zwei geschafft hätte.

Hierzu finde ich einen Spruch aus »Eric Bethke: Game Design & Production« sehr passend: Set goals, not hours! Ich denke, in einem Entwicklerteam, in dem die Mitarbeiter ihre Zeit »absitzen«, ist grundsätzlich der Wurm drin. Wenn ein Teammitglied eine Aufgabe schneller als geplant abgeschlossen hat und nicht eigenständig den nächsten Task beginnt, Bugs aus der Datenbank fixt oder einen Kollegen, der vielleicht hinten dran hängt, unterstützt, dann gibt es zwei Möglichkeiten: Entweder er ist tatsächlich so eine faule und unkollegiale Socke – dann gehört er nicht ins Team. Oder aber er wurde von außen dazu gemacht (Stichwort: anerzogene Hilfslosigkeit – siehe oben). In beiden Fällen tragen die Verantwortung dafür Projektleitung und die verantwortlichen Leads – getreu der alten Bauernregel: Der Fisch stinkt vom Kopf her!

Das zweite Vorurteil bezieht sich auf den Projektplan als solchen und besagt, dass ein Plan bereits zu dem Zeitpunkt veraltet ist, an dem er erstellt wurde. Und da es so einen enormen Aufwand darstellt, ihn zu pflegen, macht sich ohnehin niemand die Mühe, so dass er spätestens nach vier Wochen nichts mehr mit der Realität zu tun hat. Hier trifft meiner Meinung nach falsches Vorgehen bei der Projektplanung selbst



Wer vergisst, Urlaube oder Puffer für Krankheitstage im Projektplan zu berücksichtigen, kann böse Überraschungen im Projektverlauf erleben, etwa **leere Schreibtische**.

mit mangelnder Konsequenz bei der täglichen Arbeit aufeinander (und das Zauberwort hier heißt »Konsequenz«). Bei konstanter Pflege dauert es selbst mit MS-Project nicht länger als eine halbe Stunde täglich, um den Projektplan aktuell zu halten. Und selbst, wenn man ihn nur einmal die Woche nach den Lead-Meetings (in denen der komplette Zeitabgleich für die letzten sieben Tage besprochen wurde) aktualisiert, sollte dies nicht mehr als einen halben Tag Zeit in Anspruch nehmen – vorausgesetzt, die Leads kommen wiederum ihren Rollen nach und der Projektleiter muss nicht jedem einzelnen Task nachlaufen. Dieser Aktualisierungs-Aufwand nimmt allerdings drastisch zu, wenn man die Projektplanpflege schleifen lässt und sich nur alle paar Wochen die Mühe macht, den Plan up-zudaten. Ein Plan, der vier Wochen lang nicht bearbeitet wurde, ist in der Tat schnell veraltet – und der Aufwand, ihn wieder auf Vordermann zu bringen ist unverhältnismäßig hoch. Und einen Plan, der mehr als zwei Monate lang nicht aktualisiert wurde, kann man auch gleich wegwerfen und neu aufsetzen. **Ralf C. Adam**

Weiterführende Quellen

- Andrew Rollings, Dave Morris: »Game Architecture and Design«, Coriolis Technology Press 2000 (ISBN: 1-57610-425-7)
- Heather Chandler: »The Game Production Handbook«, Thomson 2006 (ISBN: 1-58450-416-1)
- John Hight, Jeannie Novak: »Game Development Essentials: Game Project Management«, Thomson 2007 (ISBN: 1-4180-1541-5)

In der nächsten Ausgabe...

Den für diese Ausgabe versprochenen Abschnitt über typische Lügen bei Projektverhandlungen und der anschließenden Zusammenarbeit mit Publishern reichen wir aus Platzgründen im dritten Teil unserer Serie in der nächsten Ausgabe nach. Außerdem beschäftigen wir uns dann mit unkalkulierbaren Risiken (und: nein, damit sind nicht Publisher gemeint...).