

Typische Projektlügen, Teil 3

Projektplanung in der Pitchingphase

Im dritten Teil unserer Serie beschäftigen wir uns mit jener Art von Projektlügen, die meist gleich zu Projektbeginn auftauchen: mit denen in der Pitchingphase, wenn es darum geht, einen Publisher für die Finanzierung zu gewinnen. Außerdem werfen wir einen Blick auf vermeintlich unkalkulierbare Risiken.



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

Eine in der Spiele-Entwicklung ziemlich oft und häufig von Publishern gehörte Ausrede, wenn mal wieder ein Milestone komplett verfehlt oder der Gold-Master-Termin um sechs Monate verschoben wurde, lautet: »Es tut uns leid, ein unvorhersehbares Problem trat plötzlich auf!«

Ein unvorhersehbares Problem kann natürlich jeden von uns erwischen – da kann man nichts machen. Oder etwa doch? Interessanterweise habe ich noch nie an einem Spiel mitgearbeitet, dessen Entwicklung wirklich von Unvorhersehbarkeit sabotiert wurde. Wenn man den Projektleiter fragt, ob sich die Probleme nicht im Vorfeld bereits in irgendeiner Form abgezeichnet hätten, dann wird er – wenn er ehrlich ist – mit ziemlicher Sicherheit antworten: »Doch! Ich glaube sogar, mich zu erinnern, dass XY irgendwann mal in einem Meeting gesagt hat, dass das alles nicht so einfach funktionieren wird, wie wir uns das vorstellen!«

Es besteht also ein klarer Unterschied zwischen »unvorhersehbaren« Hürden und »unvorhergesehenen«. Solange man sich mit der Lüge zufrieden gibt, dass es unvorhersehbare Probleme gibt, wird man kein effektives Risk-Management betreiben.

Aktives Risk-Management in der Praxis

Wie aber tritt man nun aktiv möglichen Risiken im Projektverlauf entgegen? Grundsätz-

lich lässt sich Risk-Management in vier einfache Punkte untergliedern:

- 1 Risikoerfassung
- 2 Risikobewertung
- 3 Gegenmaßnahmen
- 4 Risk-Ownership

Die ersten beiden Punkte sind relativ klar. Bei der Risikoerfassung geht es darum, mögliche Projektrisiken zu sammeln. Das erste Mal sollte dies frühzeitig im Projekt geschehen, idealerweise direkt nach der Aufgaben- und Aufwandsplanung. Darüber hinaus kann es hilfreich sein, bereits in Design-Dokumenten mögliche Risiken bei der Umsetzung aufzuzeigen und in einem separaten Absatz festzuhalten. Außerdem habe ich in der Vergangenheit gute Erfahrungen damit gemacht, verschiedene Risiko-Listen für verschiedene Bereiche zu führen (Design, Grafik, Technik).

Bei der »Risikobewertung« gibt es die unterschiedlichsten Modelle, von einer einfachen A-, B-, C-Klassifizierung in einer Liste bis hin zu einer komplizierten Matrix (wer die bisherigen Artikel gelesen hat, weiß allerdings, dass ich nicht unbedingt ein Freund von »kompliziert gleich gut« bin, da komplizierte Prozesse von einem Team selten langfristig genutzt werden). Im Übrigen empfiehlt es sich, erst alle Risiken separat zu sammeln und anschließend gemein-

No.	Beschreibung	Lösungsvorschlag (im Vorfeld)	Plan B (falls Problem eintritt)	Verantwortlich
1.	Online-Multiplayer: Gamespy soll für den Multiplayer-Modus eingesetzt werden, aber niemand im Team hat bislang Erfahrung mit der Implementierung!	Gamespy zwecks Unterstützung bei der Lösung des Problems kontaktieren.	Es gibt eine externe Coding-Firma Code.net, die die Komplett-Implementierung von Gamespy übernimmt.	Joe Joystick (Lead Coder)
2.	Outsourcing Grafik: Fehlende Transparenz bei Projektfortschritt im Art-Bereich, nachdem externes Grafikstudio beauftragt wurde.	Wöchentliche Mini-Milestones und Reviews mit dem externen Dienstleister vereinbaren, abnehmen und kontrollieren.	Dienstleister wechseln.	Leo Lightpen (Art-Director)
3.	Feature Creep: Feature Creep bei den Einheiten und ihrer Bewaffnung.	Genauere Liste mit Must-Have, Should-Have und Nice-To-Have Waffen definieren.	Alle Nice-To-Have und Should-Have Waffen streichen.	John Joypad (Game Designer)
4.	And so on ...			

Ein Beispiel für eine einfache und effektive **Risiko-Liste**, die – wenn sie konsequent gepflegt wird – viel Ärger im Projekt ersparen kann.

DRAFT - RISK MANAGEMENT PLAN					
Updated: 01/31/05					
courtesy of R Robinson					
PROBABILITY (P)	IMPACT (I)				
	20%	40%	60%	80%	100%
20%	4%	8%	12%	16%	20%
40%	8%	16%	24%	32%	40%
60%	12%	24%	36%	48%	60%
80%	16%	32%	48%	64%	80%
100%	20%	40%	60%	80%	100%
RISK	NOTES	OWNER	(P) FACTOR	(I) FACTOR	(PI) FACTOR
Development team depth / dependencies					0%
Compatibility					0%
Public Beta: maintenance					0%
Clear demographic target					0%
Technology fit					0%
Marketing Requests					0%
New/Discovered tasks					0%
Criticality of end date					0%
Art Optimization					0%
Established/Known gameplay					0%
Code Optimization					0%
Bug of Sink bugs					0%
Financial stability					0%
Stable personnel					0%
External vs Internal					0%
Well defined schedule / milestones					0%
Impact of contract negotiations (delays to production)					0%
Localization / Import of international development					0%
Existing Engine (low risk) vs Extensive R&D (high)					0%
Play Balance					0%
Installer					0%
Gameplay fit					0%

Eine komplexere Variante einer **Risiko-Liste** (Quelle: Dan Irish: The Game Producer's Handbook).

sam zu bewerten. Andernfalls dauern Team-Meetings erfahrungsgemäß doppelt so lange, da man sich bei der Evaluierung schnell in Details verliert und darüber weitere Gefahrenherde vergisst oder übersieht.

Bei der Auflistung der Gegenmaßnahmen sollte man diese in zumindest zwei Teilbereiche untergliedern: Prävention plus Reaktion sowie Notfallpläne. In der Präventionsspalte werden mögliche Abwehrmaßnahmen aufgeführt. Dies können größere Zeitpuffer bei besonders schwierigen Aufgaben sein oder auch ein verstärktes Beobachten bestimmter Bereiche unter Berücksichtigung von Frühwarnindikatoren. In der zweiten Spalte führt man die entsprechenden Notfallpläne auf, falls sich die beschlossenen Gegenmaßnahmen als zu schwach erwiesen haben (beispielsweise Streichung von Features, Verschiebung von Deadlines, mehr und externe Ressourcen).

Besonders wichtig: Jedem Risiko muss ein Verantwortlicher zugewiesen werden, andernfalls nutzt die beste Planung nichts. Dieser Verantwortliche ist dafür zuständig, das Risiko im Auge zu behalten (auf Neuhochdeutsch: zu tracken) und dafür zu sorgen, dass die Präventionsmaßnahmen – und im Falle des Falles auch der Notfallplan – umgesetzt werden.

Kein Hexenwerk

Weiter oben habe ich geschrieben, dass man die allererste Risiko-Liste zu Projektbeginn erstellen sollte. Im Verlaufe einer Spiele-Entwicklung tun sich aber fast täglich neue Risiken auf. Deshalb nutzt es wenig, die Risiken nur einmal zu sammeln. Risk-Management funktioniert nur als aktiver Prozess. Dementsprechend sollte man eine Risk-List wöchentlich auf den neuesten Stand bringen (»Change Control«). Hier hat es sich als sinnvoll erwiesen, möglichst proaktiv

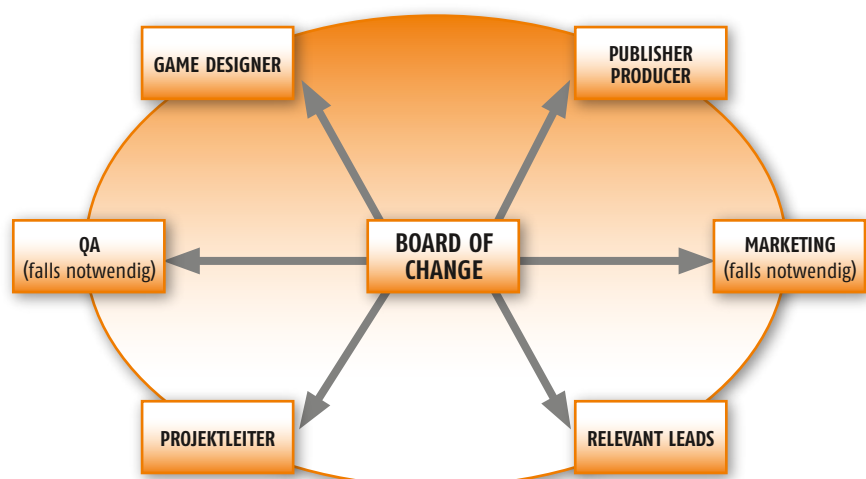
an die Sache heranzugehen und auf das gesammelte Wissen der Mitarbeiter zurückzugreifen. Ein Projektleiter ist deshalb gut damit beraten, immer wieder in den wöchentlichen Team-Meetings die anstehenden Aufgaben der nächsten Wochen daraufhin abzuklopfen, wo vielleicht noch nicht erkannte Gefahren lauern. Je erfahrener das Team, desto höher ist die Wahrscheinlichkeit, dass ein Mitarbeiter eine Risiko-Aufgabe schon einmal erledigen musste oder zumindest an einem Projekt beteiligt war, bei dem eine solche aufgetaucht ist. Die Erfahrung zeigt: Die Mitarbeiter wissen es immer am besten, man muss sie nur fragen.

Risk-Management ist weder Hexenwerk noch sonderlich zeitaufwändig, die Kunst besteht einzig und allein darin (wie bei praktisch jedem Projektmanagement-Prozess), es konsequent durchzuziehen.

Der Publisher schaut nur auf den Preis

Einer der Hauptgründe, warum viele der bisher aufgeführten Projektmanagement-Methoden in Teams nur so selten zur Anwendung kommen, liegt auch an dem weit verbreiteten Irrglauben, der da lautet: »Der Publisher wird nur einen Deal mit mir machen, wenn ich gnadenlos günstig bin«. Das Entwicklerteam kalkuliert unter dem eigentlich geschätzten Aufwand (von Zeitpuffern wollen wir hier erst gar nicht reden) und spekuliert auf eine Budget- und/oder eine Zeiterhöhung, wenn das Spiel bereits weit fortgeschritten und schon so viel Geld hineingeflossen ist, dass der Publisher nicht mehr zurück kann (»Point of no Return«). Dass bei einer derart mit der heißen Nadel gestrickten Planung vernünftiges Projektmanagement als Erstes auf der Strecke bleibt, ergibt sich fast zwangsläufig. In einer solchen Konstellation allerdings können am Ende alle Beteiligten nur

Die typische Zusammensetzung eines »Board of Change«.



Auf Messen wie der Games Convention, der Game Developers Conference und früher der E3 finden oftmals die ersten Pitching-Gespräche mit dem Publisher statt.

verlieren. Der Publisher wird angefressen sein, wenn das Team mitten in der Entwicklung ankommt und mehr oder weniger offen und unverblümt feststellt: »Es hätte doch von Anfang an klar sein müssen, dass man mit einem derartigen Mini-Budget solch ein galaktisch gutes Spiel gar nicht entwickeln kann.« Resultat: Ein Folge-Deal dürfte unwahrscheinlich sein, und da Publisher auch untereinander ihre Erfahrungen austauschen, ist der Ruf schnell ruiniert. Bei dem Entwickler wiederum wird die Erkenntnis reifen, dass Publisher nicht bereit sind, die realen Kosten für ein Spiel zu erkennen, geschweige denn, sie übernehmen zu wollen. Womit wir wieder beim anfänglichen Irrglauben wären.

Viele Publisher tragen eine gehörige Portion Mitschuld an dieser Misere. Vor einiger Zeit wurde ich als Berater zu einem Projekt hinzugezogen, das bereits ein Jahr in der Entwicklung war. Das Team hatte zu Beginn eine relativ genaue Vorstellung davon, wie lange und wie viele Ressourcen es zur Fertigstellung des Spiels benötigen würde. Der Publisher bot ihm 30 Prozent weniger an und wollte zudem noch weitere Inhalte. Obwohl noch über 15 Monate bis zum Gold-Master-Termin verblieben, war zu diesem Zeitpunkt schon klar, dass die Entwick-

lung länger dauern würde, als im Ursprungsplan veranschlagt. Selbstverständlich wurde sie so teuer und verfehlte trotzdem die Qualitätsvorgaben. Hätte der Publisher der Originalplanung von vornherein zugestimmt, dann hätten die Chancen auf einen erfolgreichen Projektabschluss wesentlich besser gestanden.

Zeit ist relativ

Ein weiteres Problem: Wenn ein Entwickler dem Publisher weiß macht, er brauche nur 18 statt der eigentlich kalkulierten 26 Monate, wird er bis zum ominösen Point of no Return (etwa in Monat 14 bei einer solchen Konstellation) erst mal versuchen, so viel Spiel wie möglich fertig zu stellen, um den Geldgeber zu überzeugen, dass es sich lohnt, weiter zu investieren. Erhält der Entwickler eine Verlängerung von weiteren vier Monaten, wird er in dieser Zeit mit hoher Wahrscheinlichkeit das Programm nicht weiter voranbringen, sondern erst mal die ganzen Bugs beheben müssen, die seine 14-Monats-Hetze zuvor verursacht hat. Und selbst, wenn er danach noch einmal vier weitere Monate zugestanden bekommt, ist es ohnehin zu spät, um noch grundlegende Änderungen und Verbesserungen vorzunehmen. Zumal viele Publisher auch noch den Kardinalfehler machen, mit einer Vertragsverlängerung zeitgleich auf mehr Inhalt zu pochen (»Gut, wir geben euch mehr Zeit, dann wollen wir aber auch mehr Spiel!«). Das dreht die Spirale gnadenlos weiter hin in Richtung Chaos. Hätte der Entwickler also von vornherein die 26 Monate auf den Tisch gelegt, hätte er ein wesentlich besseres und runderes Ergebnis abliefern können als bei einer »18 plus 4 plus 4 Monate Entwicklung«.

Was der Publisher wirklich will

Die Fronten scheinen hier also verhärtet zu sein. Auf der einen Seite das typische Entwicklerteam, das nie pünktlich ist und alle zwei Monate an die Tür des Publishers klopft, um neue unverschämte Forderungen zu stellen. Auf der anderen Seite der sture Publisher, der keine Ahnung von Spiele-Entwicklung hat und nicht bereit ist, den realen Problemen und damit verbunden Kosten ins Auge zu sehen.

Hier hilft es, einen Schritt zurückzutreten und sich klar zu machen, was beide Parteien überhaupt wollen. Was ein Entwicklerteam will, wissen wir alle recht gut, da wir alle Entwickler sind: Wir wollen ein geniales Spiel erschaffen. Und wenn am Ende dabei auch noch Geld in der Kasse übrig bleibt, umso besser. Was aber will der Publisher? Die Antwort darauf ist ähnlich einfach: Er will Geld verdienen. Dafür setzt er eine einfache Kosten-Nutzen-Rechnung an, überlegt sich zu Beginn, wie viele Einheiten er von einem Spiel wohl absetzen wird und kalkuliert danach den Preis, den er bereit ist, für die Entstehung zu bezahlen. Das bedeutet im Umkehrschluss, dass der Preis zunächst egal ist. Wichtig ist dem Publisher, dass er die Ware zum im Vertrag festgelegten Preis bekommt, damit seine

Kalkulation aufgeht. Ergo zahlt ein Publisher jeden Preis, wenn er an das Ergebnis und seine termingerechte Ablieferung glaubt. Kein vernünftiger Publisher entscheidet nur aufgrund des Budgets. Und von unvernünftigen Publishern sollte man als Entwicklerteam ohnehin die Finger lassen. Denn die Frage »Können wir das nicht auch billiger haben?« impliziert nichts anderes als »Wir vertrauen deiner Planung sowie so nicht, lieber Entwickler«. Und möchte ich als Entwickler eine oftmals mehrjährige Geschäftsbeziehung mit einem Partner eingehen, der mich gleich zu Beginn mit einer solchen Aussage konfrontiert?

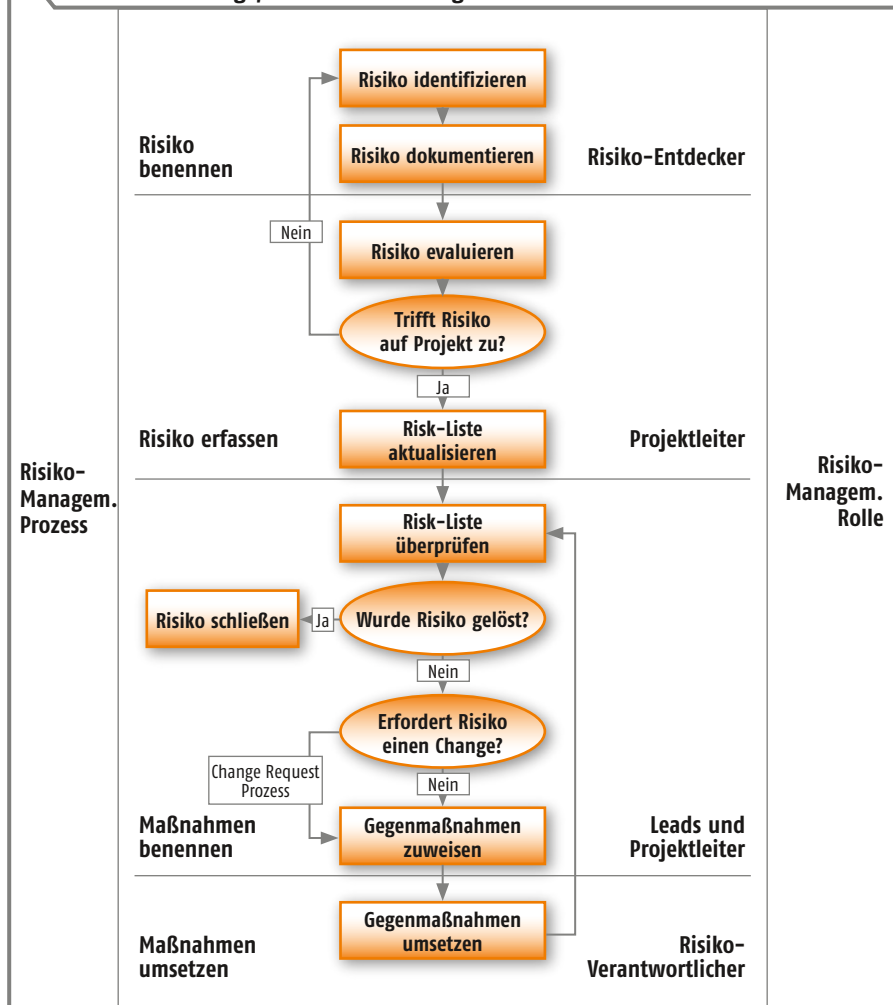
Wie bekomme ich mein Wunschbudget?

Genauso wenig, wie kein (vernünftiger) Publisher versuchen wird, ein Team im Preis zu drücken, wird er einem Entwickler einfach ein beliebiges Budget genehmigen. Wie also bekomme ich von einem Publisher das Geld und die Ressourcen, die ich zur Entwicklung eines genialen Spiels benötige?

Das Zauberwort heißt auch hier wieder »Planung«. Viele Entwickler scheuen sich davor, dem Publisher einen genauen Einblick in ihre Planung zu geben – sei es, weil sie denken, das geht ihn nichts an (Fehler), oder weil sie schlicht keine haben (mächtig großer Fehler!). Dabei übersehen sie, dass sie die Planung nicht nur für ihre internen Prozesse und Abläufe dringend benötigen, sondern auch, dass eine Planung die mächtigste Waffe überhaupt gegen den »großen bösen Publisher« ist. Ein Publisher wird im Laufe einer Entwicklung todsicher immer ankommen und nach Änderungen oder Ergänzungen von Inhalten fragen. Dies geschieht meist nicht mit finsternen Hintergedanken, sondern schlicht, weil bei einem Publisher viele Parteien involviert sind (Marketing, Sales, PR, QA usw.), deren Interessen der Producer unter einen Hut bringen muss. Überdies sind die gemachten Vorschläge hin und wieder ganz sinnvoll.

Wie aber will sich ein Entwicklerteam gegen eine Kostenexplosion auf seiner Seite schützen, wenn es seine Kosten dem Publisher gegenüber nicht transparent hält, sondern versucht, sie zu verschleiern? Auf eine Feature-Anfrage des Publishers mit einem pauschalen »Das kostet 100.000 Euro mehr« zu antworten, ist weder seriös noch hilfreich. Der Publisher wird dies höchstwahrscheinlich mit einem »Ihr könnt mir doch nichts erzählen! So viel Aufwand ist das doch nicht. Das muss noch im Budget drin sein.« kontern. Und selbst, wenn der Publisher darauf eingeht: Was passiert, wenn man sich bei einer solchen aus der Luft gegriffenen Zahl sogar verschätzt hat, und das Ganze noch viel teurer wird? Wie viel hilfreicher ist es da, dem Publisher exakt belegen zu können: »Für Zusatzinhalt X müssen folgende Mitarbeiter folgende Aufgaben erledigen, das dauert so und so lange und es kostet diese Summe extra.« Ein weiterer Vorteil einer genauen Planung liegt darin, dass

Dieser Ablauf zeigt, wie Risiko-Management in der Praxis aussehen kann.



sich in ihr auch präzise festhalten lässt, welche Leistungen im Gesamtbudget nicht enthalten sind. Oft werden zum Beispiel Punkte wie FMV-Cutscenes, HiRes-Artwork fürs Marketing, Orchester-Vertonung, Sprachaufnahmen und Lokalisierung bei der Vertragsunterschrift nicht klar definiert. Der Publisher geht dann stillschweigend davon aus, dass der Entwickler die Kosten hierfür trägt, was dieser wiederum natürlich ganz anders sieht.

Nur eine genaue Planung und deren Offenlegung sorgen dafür, dass der Publisher die eventuell entstehenden Mehrkosten trägt, ohne das Gefühl zu haben, über den Tisch gezogen zu werden. Wir schlussfolgern: Entwicklerteams, die versuchen, sich als Billigheimer zu etablieren, werden über kurz oder lang in Probleme geraten und dem Publisher am Ende nicht als »günstiger Vertragspartner« in Erinnerung bleiben, sondern das Etikett »unorganisierter, unzuverlässiger Chaoshaufen« angehängt bekommen. Seien Sie deshalb clever: Kein Publisher erwartet, dass sie billig sind. Er möchte, dass Sie zuverlässig sind! Damit sind Sie automatisch besser als jedes andere Team, das er sonst unter Vertrag hat.

Ralf C. Adam

In der nächsten Ausgabe ...

Geschafft! Der Deal ist unter Dach und Fach, und unser Wunschbudget ist genehmigt. Leider enden damit die Probleme in der Zusammenarbeit mit dem Publisher nicht, sondern gehen jetzt erst richtig los. Im vierten Teil unserer Serie beleuchten wir die Missverständnisse und Projektlügen, die im Spannungsfeld der Zusammenarbeit zwischen Developer und Publisher auftreten können.