



**Contract Walk-Through – Release 3:  
Effects on Quality of Life**

**February 2006**

## Table of Contents

<i>Introduction</i> _____	3
<i>Contract Essays</i> _____	4
<b>Strategic Use of Planning: Technology Evaluation, Milestone Dates, and Description of Deliverables</b> _____	4
<b>Managing Change Orders and “Feature Creep”</b> _____	9
<b>Responsibility for Delay</b> _____	13
<b>Marketing Requests</b> _____	16
<b>Outsourced Assets and Potential Effect on Developer’s Quality of Life</b> _____	20
<b>Acceptance of Deliverables</b> _____	24
<i>Contributors</i> _____	28

### About the IGDA

The International Game Developers Association is a non-profit professional society that is committed to advancing the careers and enhancing the lives of game developers by connecting members with their peers, promoting professional development, and advocating on issues that affect the developer community. For more information on the IGDA, please visit [www.igda.org](http://www.igda.org) or e-mail [info@igda.org](mailto:info@igda.org).

## Introduction

### Welcome to Release 3 of the IGDA Contract Walk-Through!

The IGDA Quality of Life Committee is the sponsor of this third release, designed to help development studios address quality of life issues in their game dev deals.

This release differs from earlier material in that it tends to deal with broader issues that are not necessarily easy to isolate in one or several contract clauses. The articles here frequently deal with strategies and broader approaches that can be applied in the overview when negotiating game development contracts.

Each article in our new release is written by a volunteer lawyer with experience in negotiating these deals.

Each article focuses on quality of life issues from a different perspective – examining different aspects of game deals – in a way that can help a studio understand, negotiate, and live with its deal --- and maintain a personal life for staff!

Since the first and second releases of the Contract Walk-Through were published on igda.org, beginning in May 2003, over 4,000 developers have downloaded these materials!

In reading the Contract Walk-Through, it should be remembered that this is presented for general informational and educational purposes and is not intended to be legal advice. Every situation is unique and developers are always best served by working alongside their own experienced legal counsel.



Jim Charne  
Chair, Attorneys Panel, IGDA Contract Walk-Through  
charne@usa.net

Law Offices James I Charne  
Santa Monica, CA

## **Strategic Use of Planning: Technology Evaluation, Milestone Dates, and Description of Deliverables**

By Jim Charne, © 2006

### **Introduction**

Any discussion of the use of the Statement of Work, Technical Design Doc, Game Design Doc, milestone dates, and description of deliverables as tools to enhance developer quality of life differs fundamentally from Contract Walk-Through Release 1 and Release 2 topics.

This is because these documents and contract exhibits are not standard publisher “boilerplate” clauses, like reps, warranties, indemnities, breach clauses, and others to be negotiated, but rather highly fact-oriented descriptors of work unique to your proposed game project.

These documents are the standard against which your work will be judged. By carefully crafting their contents, you can help advance the quality of life for your development team and improve the chances of a happy publisher who will see the game come through on time and on budget.

### **The Problem**

We all know game development makes for a very demanding career. Schedules are long and difficult, teams keep getting larger, there is never enough time or money, every project goes down to the wire, and every aspect of a game is subject to the review and absolute subjective approval of the game publisher, content licensor, and in the case of console or handheld games, the console hardware manufacturer.

Approvals and corresponding payments are based on achievement of deliverables set forth on the milestone schedule. The milestones are based on realizing the technical and creative elements of the design as agreed upon by the developer and publisher.

Development contracts are long, detailed, and weigh heavily in favor of the publisher. The result is an industry in which extraordinarily talented and motivated people work countless hours in high-pressure conditions throughout the life of a project, with very little room for error, in order to meet the requirements of the job.

While game development can be personally rewarding, the demands placed on studio personnel may also lead to burnout, frustration, battered personal lives, health problems, personal relationships stressed to the breaking point, talent departing for less physically

and emotionally demanding careers, and the inevitable rise of quality of life as an issue in the game development community.

Publishers generally control the content of development agreements. But there are certain contract provisions driven by the developer. By carefully planning, negotiating, and then managing those contracts points, developers may be able to improve quality of life over the life of game development.

By careful planning when the statement of work (technical design doc and game design doc), milestone dates, and descriptions of deliverables are being put together, and by exercising control where and when it is available, developers can moderate some of the pressures that lead to impossible work loads, hours, and cash flow crisis.

## **The Strategy**

### **1. Managing Publisher Expectations.**

The road to improving quality of life starts with the earliest discussions between developer and publisher. Managing expectations is the developer's job. Every developer wants to make the greatest game possible. But if quality of life is a priority for your team, it is important to maintain a realistic connection between the game design, technical issues, development calendar, development strategy, and dollars and cents cost of the project.

Early discussions with your publisher should focus on setting expectations and a preliminary budget. Get to understand what kind of game the publisher wants.

Understand the budget your publisher has set aside in its planning. Calculate whether it is possible to achieve the results that are expected for the budget that is proposed. Evaluate the skill sets of your own team and in-house technology. Review what may be available in the middleware market. Determine the fastest, easiest, and least expensive way to achieve the results that you and the publisher want. If you determine the game as defined can't be delivered at the price or in the time frame the publisher has quoted, it is important to make that clear early on. The only way a developer can make up for its failure to set realistic parameters is to pressure the publisher for more money and time, or by demanding over-the-top efforts on the part of its staff. Neither alternative is good for long-term health of the developer.

If the publisher is insistent on a very elaborate game at a rock bottom, impossible price, it may be best to walk away. In the alternative, if it is appropriate and you feel comfortable with the idea, work with the publisher to identify outside resources that may be tapped to deliver portions of the project. It may be possible to obtain selected assets from third parties at lower cost than is possible internally. Planning for outsourcing may take pressure off your team.

Just as the publisher may have a “green light” process after it reviews the preliminary documents prepared by your team, you should be careful not to be automatically committed to delivering the game in the event you are not able to agree on the design parameters, budget, and milestones.

2. Look for the most established, lowest tech, fastest, easiest, least risky way to achieve the design.

Developers by their nature love to rise to a challenge. Your technical staff will certainly look at any project as an opportunity to expand the technology of the studio. The project will be seen as a way to showcase the technical ability of the organization.

However, implementing new technology, no matter how elegant or advanced, can add to the development time and cost of the project. New technology also brings with it an element of risk. New technology has to be perfected, debugged, and integrated into the development engine or developer tools catalogue.

While a game may appear to require additions or enhancements to the developer technology base, it is good practice to undertake an internal critical assessment to determine how much new tech is really required to do the game. Once that assessment is complete, a determination can be made as to the least intrusive means to source the capability – be it internal development, engaging a subcontractor (with all the requisite issues of management), or licensing third party tools or code.

The less new technology, the less chance there will be technology-based delays or distractions during development.

If the goal is to improve quality of life, the tried and true method is the best bet.

3. Set Milestones Based on Developer Cashflow Needs.

Developers have to be experts at managing their cash flow. In order to do so, it is important to fully understand the milestone process, going beyond all dates for review, approval, and processing of payments. Always give yourself a cushion. There is little more damaging to studio morale and quality of life than telling team members who have been working so hard on a project that their paychecks may be delayed pending receipt of a publisher milestone check.

Developers should always prepare and manage internal cash flow projections based on best and worst case scenarios under their development deals. Invoices must be tracked through the approval process and the publisher accounts payable department. No milestone should be considered complete until it has been approved in writing by the publisher, invoiced, the check issued, and payment received and deposited.

While it may be tempting to set a few large milestones that represent major turning points for the game (such as first playable, alpha, beta, gold master), it is dangerous for a developer to go so long between checks.

Defining monthly or even semi-monthly milestones also assures the developer cannot stray too far off course. The publisher is forced to provide feedback and approvals regularly. Cash can flow to the developer on a steadier basis. Provided that developer has correctly forecast its cash flow needs, there is less likelihood of a late paycheck crisis.

4. Take enough time to write a milestone schedule you can really live with.

Publishers generally defer to their developers when it is time to write the milestone schedule. My experience is that developers do not put enough time into this important document.

The milestone schedule – both its dates and deliverables – is the standard against which the developer’s performance will be judged.

When writing up milestones, be specific as possible. Don’t give the publisher room to reinterpret the deliverables. Remember, this is the standard against which the developer will be paid. Including bi-weekly milestone completion checklists as part of the planning documentation creates a detailed road map for both developer and publisher.

Game development historically takes much longer than even the most conservative estimates. While development studios have become much better at accurately predicting achievement of development goals, it is always better to err on the side of caution. Give yourself breathing room in formulating the milestone schedule!

Delivering a milestone two weeks early makes the developer a hero. Delivering it two weeks late may result in material breach, under certain circumstances give the publisher an excuse to terminate (this issue was discussed in Contract Walk-Through Release 2), and generally causes a quality of life and cash flow catastrophe for the developer.

Completion of milestones can be delayed for many reasons; not all of which are the fault of the developer. But planning for these inevitable delays when constructing the schedule can help the developer weather the storm and deliver a great game on time and on budget.

And in doing so, in planning for the unforeseen but predictable delays, pressure can be taken off the development team that will translate into improved quality of life.

## **Conclusion**

Game development will never be easy or pressure free. Great creative expression, whether motion pictures, literature, recorded music, fine art, or interactive games, requires intense

concentration, singular focus, hard work, imagination, inspiration, and dedication. All of these are present in the best game development studios.

All these traits can better flourish where personnel do not get ground down by extreme working conditions. By managing the elements discussed above, studio personnel may be able to enhance the quality of life for their development teams. Any improvement can only lead to better games.

## **Managing Change Orders and “Feature Creep”**

By David S. Rosenbaum, © 2006

### **Introduction**

Every game development agreement requires the developer to develop and deliver the game to the publisher pursuant to agreed upon specifications and an agreed upon milestone schedule. Often, during production, the specifications change and the challenge for both developer and publisher is to manage those changes to meet the parties’ expectations for the project. In these instances, the development agreement can provide some guidance, but it will not provide absolution from any problems that may be engendered during the process.

The challenge of reaching the agreed upon specifications and milestone schedule can be framed in the following questions which should be asked when the development agreement is being negotiated:

1. What are the specifications for the game?
2. If the specifications are not fully known when development starts, when will they be known?
3. What constitutes a “change” and is a Change Order required?
4. What are the costs of a change?
5. How will the parties handle the change vis-à-vis their contractual obligations?

### **Discussion**

1. What are the specifications for the game? Every development agreement provides as a basic principle that the “Developer shall develop, produce, complete and deliver to Publisher the game and each version of the game in accordance with the Title Specifications and the Milestone Delivery Schedule, and such development shall be completed at Developer’s sole cost and expense, subject to the terms of this Agreement.”

Terms like “Title Specifications” and “Milestone Delivery Schedule” are defined terms in each development agreement. The names of the defined terms vary from publisher to publisher, but they cover the game design document, technical design document and delivery schedule which establishes the roadmap for the developer to meet during development to satisfy the publisher’s requirements and, most importantly, get paid for the work performed. Lawyers don’t develop games so they have to rely on their clients (publishers and developers) to confirm to them that the specifications attached to the agreement are as the parties agreed. So lawyers will ask: Are the milestone descriptions correct? Are they manageable? How much of the text of the game concept is “puffery”

used to interest the publisher in the concept in the first place and needs to be revised so as not to have the roadmap end on a cliff?

2. If the specifications are not fully known when development starts, when will they be known? Very often, given the long development cycle for some projects, when the agreement is negotiated, very little of the actual game specifications are fully known and the milestone schedule attached to the contract is full of “TBDs”. So the publisher will fund the developer’s pre-production process during which the developer will produce both a final game design document and a final technical design document, along with a playable prototype or proof-of-concept demo, upon which the publisher will then decide whether or not to greenlight full production of the game. Preparation of the final design documents will affect how the remaining milestone schedule will look so when submitting the design documents, developers will also need to submit a revised milestone schedule that will show how the design documents will be implemented to complete the game project. If the publisher greenlights the project to completion, then the development agreement will need to be amended to remove the original exhibits or schedules (which have mostly “TBDs”) and replace them with the final design documents and amended milestone delivery schedule.

3. What constitutes a “change” and is a Change Order required? Development agreements will vest in the publisher final editorial control in all areas of development of the game. Development agreements provide that the publisher reserves the right to request changes to the agreed specifications taking into account commercial and creative considerations and requirements of hardware manufacturer or a third party licensor, if any, with respect to the game. It is when the requested change becomes “significant” that trouble can often be lurking.

There is no commonly accepted standard for what constitutes a “significant design change.” To paraphrase the opinion of a famous Supreme Court justice, Publishers like to think they know a significant change when they see one, which they concede will require a “change order,” which is an amendment to the development agreement; as a result of this belief, publishers will be reluctant to agree to contractual limits on how changes are to be dealt with when they occur. If a change isn’t deemed “significant,” and many changes aren’t significant, then publishers expect developers to accommodate the requested change (for no additional fees) and stay on schedule. The challenge for the developer is to protect against not being paid for changes that are significant and have a material affect on the agreed specifications and the delivery schedule.

So what is a “significant” design change? On some levels, a change is significant if it causes the actual specifications to materially differ from the agreed specifications and requires either “significant” reprogramming of the game code, or “significant” amounts of new artwork and animation. It is admittedly a tautology, because it doesn’t fully answer the question of what “significant” means. If the hardware manufacturer conditions granting concept approval on changing elements in the game as submitted, that will affect

both the programming and artwork quotients. If the publisher obtains a license to use a property in connection with a game after the game has started development, that may also affect both the programming and artwork quotients.

In practical terms, when dealing with a significant design change, the developer needs to look at the schedule first and determine how much extra time or resources, if any, will be required to accommodate the requested changes. In some instances, development budgets include a generous contingency amount to cover a developer for additional costs for many changes; this occurs more often in games that are financed by third parties, e.g., banks or investment groups who prefer to build in a higher contingency from the outset of the project rather than later on when problems arise. In the more typical publisher-funded game, the budget will not likely include as high a contingency (because the budget is a largely non-refundable advance and publishers don't want to "overpay" for a project) in which event the developer must then convince the publisher that the change is significant enough to overcome the publisher's institutional bias against renegotiating development fees during the project or postponing the expected (read as "announced to Wall Street") release date. In some instances, publishers and developers will agree from the outset that if a change causes the project to fall behind by an agreed period of time, e.g., one month, then the change will be deemed significant and trigger the change order process.

4. What are the costs of a change? Game development is largely an exercise in resource and time management – the project will require a team of X members Y months to complete at a burn rate of \$Z per month. Changes dictated by the requirements of concept approval are likely to be viewed differently from changes that the publisher agrees will make a better (read as "better selling") game; both should require a change order, but in the former, the publisher will want to keep cost increases to a minimum (because some executives within the publisher will view concept approval as a developer failure rather than a publisher challenge) and in the latter, the publisher will be more willing to "spend money to make (more) money."

If the publisher is looking first to stay on schedule, then the developer needs to quantify what resources must be added to the project to implement the change and stay on schedule. If the publisher is willing to postpone the release, then resources may also need to be added, but also, the additional time to accommodate newly agreed specifications would need to be quantified and agreed.

Problems can also arise where, over the course of the scheduled development, a series of non-material changes ("feature creep") have caused the project to get behind schedule. As the development agreement entitles the publisher to request changes and requires the developer to comply with those requests (unless the requested change rises to the level of a significant design change), the developer's project manager must analyze every requested change as a potentially significant change to determine whether the developer's resources aren't being overtaxed on the project and whether the project will miss delivery dates. Feature creep can result from poor project management on the part of both the publisher

and developer. Development agreements are largely fixed price contracts. The publisher asking for non-material changes isn't concerned about the agreed budget, so it becomes incumbent on the developer to manage the development process to ensure that feature creep doesn't erode the profit margin that was forecast at the outset of the project; not to mention the adverse affect that feature creep can cause to the team having to implement changes while staying on schedule which means longer hours and weekend work to stay on course.

5. How will the parties handle the change vis-à-vis their contractual obligations? During the course of development, publishers are always looking at the schedule, budget, market viability and product quality (often, it seems, in this order of importance). Accordingly, publishers expect developers to make changes in a way that will minimize both budget increases and schedule delays (some publisher form agreements expressly warn the developer not to use a change as an opportunity to increase its profit percentage on the project).

Once the parties agree that a change is significant, amendments to the development agreement need to be negotiated and an amendment executed. Some publishers want to know that a developer will be required to implement all changes even if all parties don't agree to the amendment. From the developer's point-of-view, the development agreement should provide that the developer would not be required to implement requested changes unless and until an amendment is negotiated by the parties in good faith. For developers and publishers who have established a good working relationship, reaching a mutual agreement is an easy task.

## **Conclusion**

The success of a development project depends on a clear set of project specifications from the outset, or at least a pre-production process that will lead to design documents and schedules that provide a clear set of specifications for both the publisher and developer to gauge the progress of the project. Although changes can be unanticipated, changes are inevitable in every project so the development agreement needs to set out a process for the parties to make informed decisions on how to handle changes and change orders.

## **Responsibility for Delay**

By Vincent Scheurer, © 2006 Sarassin LLP.

### **Introduction**

Getting a game into retail on time is critically important for the publisher. Delayed release of the game may result in wasted marketing expenditure and missing a critical sales window. It may even push the release date outside of the publisher's fiscal year, with grave long-term consequences for the publisher's reputation and share price. Publishers are naturally keen to ensure that the developer commits to deliver the game by a defined date. However, it is essential to ensure that the development contract deals with the possibility that delivery may be delayed for reasons other than the developer's failure to create the game on time.

### **Sample Clause**

This is an example of wording found in many development contracts:

*“A failure by the Developer to deliver a Deliverable by the agreed date shall constitute a material breach of this Agreement by the Developer.”*

### **Discussion**

The example wording has one fundamental flaw: it simply fails to recognize that delay can occur for reasons other than the developer's own breach of contract. Instead, it states that the developer is solely responsible for timely delivery. This overlooks an essential element of the game development process: that creating a game is a partnership between a number of different entities, including the developer, the publisher, the middleware owner, the IP licensor and the console owner. Any one of these parties could cause a substantial delay in completing a game. Worse still, the developer will not usually have any leverage or contractual control over many of these other parties.

Accordingly, if completion of the game is delayed because of the activities of another party, it is difficult to understand why the developer should automatically be liable for the consequences of this delay. As noted above, the consequences of delay can be very severe indeed.

In order to consider the way in which a development contract should address delay, one needs to consider three different types of causes of delay.

Firstly, there is delay that is caused by the developer, or an entity contracted to the developer for whom the developer has accepted responsibility (such as a sub-contractor).

It is broadly reasonable that the developer should be contractually responsible for delays that fall in this category (although this should be within certain limits, further described below).

Secondly, there is delay that is caused by the publisher. This includes delay caused by late delivery of game assets (including localization materials); late approval of deliverables; and late delivery of information necessary to create the game (including style guides and the like). In those circumstances the developer should not be responsible, or liable, for any such delay.

Finally, there is delay that is caused by a third party. The third party may be a licensor that delays delivery of certain assets to the developer, or delays approval of the game, or changes its mind as to certain aspects of the license, requiring additional development work. Alternatively, the third party may be a console owner that delays granting concept approval or final manufacturing approval for a game. These third parties usually have a contractual relationship with the publisher, although the publisher is not always the dominant force in any such contract. Again, it is not appropriate for the developer to be responsible or liable for delays falling in this third category, unless those delays were originally caused by the developer's separate breach of contract, for instance as a result of bugs in the developer's code.

## **Conclusion**

The developer should ensure that the development contract properly addresses the different types of delay and their consequences. Specifically, the developer should never accept responsibility for delay caused by the publisher or by third parties with whom the developer has no contractual relationship.

In the case of the first category of delay (delay caused by the developer or the developer's contractors), the developer should always seek an appropriate cure period, and be mindful of the consequences of delay if they are treated as a material breach of the agreement (as to which, refer to Jim Charne's essay entitled "*Breach and Material Breach*" in the 2<sup>nd</sup> release of the Contract Walk-Through).

In the case of the second category of delay (delay caused by the publisher), the developer should ensure that the contractual delivery dates will be adjusted automatically if any such delay occurs. This may sound straightforward, but it is not as simple as it may appear. In order to address delay properly, the development contract must first expressly list all of the obligations of the publisher, including all delivery obligations and testing obligations, together with a fixed date by which these must be carried out. If there is no such date, then it is difficult to show that there has been any delay in the first place; in which case, it will be practically impossible for the developer to rely on that delay to adjust the contractual timetable. Ultimately, the development contract should have a clear schedule of

deliverables required from the publisher, together with the dates for delivery of those deliverables.

It is even harder to address the third category of delay (delay caused by a third party). Even if the publisher has accepted that delay caused by a third party will require adjustment of the delivery dates, how is third party delay calculated in the first place? After all, many third party licensors won't commit to delivering assets or approving concepts within a certain time, and console owners are not in the habit of specifying fixed time periods for approving concepts and games. The simple solution is to set out a schedule of assumptions for all deliveries and approvals required by third parties. For instance, the parties will agree a time period for console owner gold master testing. Neither the publisher nor the developer will be contractually committing to these time periods, as they will be outside the control of both parties; but at least they will provide the means to measure delay caused by a third party. As a result, should testing take longer for reasons outside the control of the parties (i.e. not because the developer's code has multiple bugs, or because the publisher failed to send the candidate to the platform owner on time), then the delivery timetable would be amended accordingly.

The developer should also ensure that the publisher is properly notified as soon as it becomes apparent that any delay will affect a delivery date.

Finally, the developer should also consider the impact of any "Force Majeure" clause, should a delay occur for reasons outside the reasonable control of either party.

## Marketing Requests

By Barry Seaton, © 2006

### **Introduction**

Marketing requests from publishers during development that have not adequately been addressed in the development agreement can have a significant impact on developers' quality of life (QoL). These inevitable requests for art, screen shots, trade show demos, marketing videos, interviews, studio visits and the like often occur "spur of the moment" and must be quickly fulfilled. The diversion of resources and distraction accompanying the tasks can put undue pressure on the development schedule and budget, negatively affecting QoL.

### **Sample Clause**

Here is an example of a real marketing request clause from a development agreement:

*"At its sole cost and expense Developer shall develop and deliver to Publisher, during the development cycle at Publisher's request, interactive demonstration versions of the Game (which shall be suitable for introduction at industry trade shows or for incorporation into sampler discs and other promotional compilations). Developer shall also provide Publisher with certain materials (e.g., high resolution character art, high resolution "screen shots", concept artwork, Game Assets, etc.) as Publisher may request in its sole and absolute discretion from time-to-time, to use in connection with the marketing and promotion of the Game. Developer agrees to be reasonably available to assist Publisher with the marketing and promotion of the Game including, but not limited to, making promotional appearances."*

### **Discussion**

Marketing requests from publishers are a fact of life and developers understand this. The requests are sometimes last minute because publishers are either attempting to capitalize on marketing opportunities with short windows or meet a manufacturer deadline. The requests are not maliciously made to torpedo successful development, reduce profits, or reduce developers' QoL. In the end, good marketing support from publishers is often critical to a game's success, and benefits both publishers and developers. So how does this necessary step in game publishing negatively affect developers' QoL?

The biggest impact on QoL results from the marketing requests' stress on an already tight development schedule and budget. Time, work and costs to fulfill marketing requests are either not figured in or underestimated in preparing development schedules and budgets,

which rarely contain much cushion. When the satisfying of requests for screen shots or other game assets, and especially demonstration versions, diverts resources away from completing the game, that extra time and cost eats into the schedule and budget. It also steals focus from the main task at hand. The developer then has no alternative but to make its team work longer hours (late nights, weekends, etc.) at the developer's expense, resulting in the developer not being able to compensate its team adequately for the extra work. The additional work without adequate compensation, including the stress, exhaustion, lowered morale and distraction that often accompany it, can have a substantial negative impact on QoL.

The main problems with marketing requests clauses is that they are usually pretty vague, and they do not establish sufficient parameters for the division of labor and responsibility and the timing for such requests. This is somewhat understandable, given that publishers may not have a marketing plan in place at the time the contract is signed, and publishers wish to remain flexible to exploit opportunistic marketing avenues. However, publishers may be well served by not taking unfair advantage of these clauses.

According to the sample clause above, the developer is required at its own expense to create and deliver a demonstration version of the Game during development when the publisher requests. The demo requested might vary depending whether it's for exhibition at a trade show or incorporation into other software, and no demo specifications are identified. If the demo version has not been figured into the project budget and schedule, the demo's creation will be on the developer's own nickel and will divert resources necessary to maintain the schedule. These problems can become much greater if the publisher requires one type of demo when the developer is anticipating another, if the publisher requires an extremely elaborate demo, or if the publisher requests the demo at the late stages of the project.

Another problem with the sample clause is that it requires the developer to provide the publisher with certain art assets and other assets from the Game at the publisher's request from time-to-time in its sole and absolute discretion. It is very open-ended in that it does not specify what assets will be requested, when they will be requested, how often and for what purpose. Any requests for this material can be a distraction and will divert resources away from the project, but numerous or extensive requests can have a potentially large impact on the project's schedule and budget.

The requirement for the developer to be available for promotional appearances is qualified by a "reasonableness" standard, but still is undefined. Developers of big games are sometimes requested to do press tours outside their countries. Requiring key members of the development team to make promotional appearances or attend interviews or press tours during development, especially at the critical development stages, can result in depriving the development of critical resources, creating a leadership void, and the remainder of the team having to overcompensate for the absence of key personnel. The possible delay and

distraction caused these absences can put additional pressure on the developer and its team to deliver a quality game, while staying within schedule and budget.

This haphazard approach to obtaining developers' cooperation with publishers' marketing efforts does not serve either party well in the end. Putting additional pressure on developers and creating distractions can result in games being delayed, game quality suffering, additional costs, and even a breakdown in developer-publisher relations.

Alternatives to these general marketing request clauses are in the nature of more specific contractual provisions that attempt to separately identify the marketing requests that the publisher will make or is likely to make, and to define responsibilities for the work and cost for such requests and their timing. How this is done depends on the type of marketing request in question.

If the publisher wants a demonstration version of the game, a separate clause can be included setting forth the parties' agreement regarding the type of demo version, date for delivery, minimum specifications and features, and responsibility for cost. This agreement could alternatively be included as a separate milestone in the game's milestone schedule, design documents or milestone completion checklist. If the parties do not know or have agreement on these parameters at the time of contracting, the clause could be a simple agreement that a certain type of demonstration version will be provided at a given date with the features and costs to be negotiated by the parties in good faith at a later time.

Should the publisher not volunteer the information, the developer should ask the publisher during contract negotiations about its expectations for the developer to provide art, "screen shots" and other game assets, and for the developer to make promotional appearances. Most importantly, agreement at a sufficient level of detail will allow the parties to build the responsibilities, timing and cost into the project schedule and budget. Additionally, it will allow specific clauses to be drafted into the contract to cover the most critical marketing items if the parties wish to do so.

Another alternative is for the contract to include a marketing assets responsibility matrix either as an exhibit to the contract or in the contract itself. The matrix can list the various items the publisher will require and assign responsibility and delivery dates for the items. It can also specify the party bearing the cost for the item. Setting forth this information in the contract being negotiated allows the parties to discuss these matters at the earliest stages, and negotiate a mutually satisfactory arrangement.

Realizing that it is often difficult for the publisher to anticipate their marketing needs with any specificity at the contract stage, a further approach used is the insertion of limiting language in the marketing request clauses to attempt to build some level of control over the marketing requests. For example, developers can request that the clauses reflect that marketing requests be "reasonable", be limited to a certain number of requests within a given period, or not be made during certain critical stages of development. Also, a

provision could be added to the clause that the publisher will be responsible for additional costs resulting from delays caused by marketing requests and that developer is not penalized for such delays. While these attempted limitations will meet with resistance from publishers, at a minimum it opens the door for discussion and possible resolution of the issues.

### **Conclusion**

Marketing requests during development are a fact of life for developers. But excessive or unanticipated requests for marketing assets or promotional appearances by key personnel can place increased stress on the development schedule and budget, thereby negatively impacting developer QoL.

Both publishers and developers are best served by discussing the publishers' marketing needs and expectations in as much detail as possible during contract negotiations, and building a realistic and reasonable marketing assets delivery plan into the project schedule, project budget and the development contract.

## **Outsourced Assets and Potential Effect on Developer’s Quality of Life**

By Patrick Sweeney, © 2006

### **Introduction**

One of the realities for a Developer is the ability to outsource certain portions of the development process. Whether you are outsourcing technical aspects of your title such as testing or programming functions or the more creative aspects like music or sound design, this ability can give the Developer the necessary tools to make the best game possible by spreading around core competencies and expertise among other parties.

However if not carefully managed, this same tool can give rise to management and contractual problems for the Developer. A poorly managed outsourcing relationship may result in a negative impact on the quality of life for that Developer and his team.

### **Where and How Outsourcing May Go Wrong**

Not surprisingly, a Developer’s ability to outsource parts of the development process is often closely monitored by the Publisher. Publishers want to make sure their money is well-spent on quality people and that their chosen developer doesn’t simply “pass the buck” to someone less qualified to do the job.

Accordingly, Publishers control Developer outsourcing by inserting themselves into the process.

Most Publisher agreements contain a provision that reads something like this:

*“Developer may not assign this Agreement, delegate, and/or sub-contract any of the rights or obligations hereunder without the prior written consent of Publisher.”*

This control exercised by the Publisher can result in a Developer’s inability to use the most cost-effective and/or the Developer’s most trusted contractors. Combine this with a Publisher’s ability to reject Developer milestones for a multitude of (generally subjective) reasons, and the Developer could be in breach of contract with his Publisher as well as multiple outsourcing contractors!

This situation is likely to arise when a Publisher rejects a milestone, thereby delaying payment to the Developer. However, payment to a Contractor (who may have nothing to do with the milestone rejection) is often not conditioned on your payment from the Publisher, so now the Contractor becomes a creditor.

Alternatively, a non-performing Contractor (either due to late delivery or sub-standard performance) may cause a milestone rejection by your Publisher. In this case, the Developer can't meet its obligation due to the Contractor's failure to perform. Unfortunately, the Publisher may not be sympathetic to your plight.

In either situation, there is an unhappy Publisher with a rejected milestone, a cash-strapped Developer behind schedule and an unpaid Contractor. The Developer is caught in the middle of this scenario and is ultimately responsible to the Publisher.

The dynamic between a Publisher's milestone rejection and your outsourcing relationships with your Contractor(s) will have a severe impact on the Developer's quality of life.

Consider the following implications:

- Developer expenditure of added internal resources to deal with these problems.
- Potentially diminished reputation and goodwill from an unpaid Contractor.
- Potentially diminished reputation from the Publisher.
- A development schedule that may be made tighter due to Contractor delays/Publisher rejections.
- Cost to remedy these deficiencies may result in cost over-runs, cutting into your margin as well as your ability to pay staff and overhead (overtime, etc.).
- Low morale among staff due to added pressures of resolving these circumstances.
- Potential breach of contract actions/remedies on all sides and their associated costs.

### **Avoiding the Pitfalls**

Although no magic contract clause is going to keep these problems from arising, a careful approach when dealing with Contractors will preserve your internal quality of life and minimize your risks of using Contractors.

#### Know your outsourcing needs from the outset.

- Before you finalize your Publisher arrangements, know in advance what your outsourcing needs are likely to be. Where possible, obtain approvals from the Publisher before your publishing deal is finalized. Append the pre-approved Contractors in an exhibit to your agreement.
- Don't wait until your deal is in place with the Publisher to start shopping around for your outsourcing contractors. You are likely to have a "ballpark" idea of your budget for various areas. Knowing a Contractor's availability and price will save you time and resources later as well as give you a better certainty about the budget you're working under from the Publisher. No one wants to find out one year into development that their music costs are going to be triple what they thought or that their favorite provider can't meet your schedule. The alternative is to find a provider that fits your budget, but at that point, you get what you pay for and your game may suffer.

Build in flexibility with your Publisher where you can.

It's not always easy to have your Publisher meet you "half-way" when negotiating your contract. But most Publishers will recognize your need for some freedom when selecting a Contractor.

- Looking back at the Publisher's consent requirement for any assignment of rights, most Publishers will agree that their consent "will not be unreasonably withheld or delayed." This small addition may help your contractor choices get approved by the Publisher in a more efficient manner. Note: This additional qualifier works in addition to having "pre-approved" contractors (as noted above).
- Take all steps to avoid agreeing to "time is of the essence" provisions anywhere in your Publisher agreement. Although seemingly innocuous and self-evident, this provision has a legal meaning to make your delay (regardless of how brief and whatever the cause) a material breach of your Publisher agreement. Not having this language preserves your position that your one-week delay due to Contractor issues may not be material enough for breach and/or termination.
- Broaden your Force Majeure clause. Most Force Majeure clauses are designed to allow for suspended performance in the event of fire, riots, war, natural disasters, acts of God, etc. A common add-on to this language is "or any cause beyond the party's reasonable control." It won't make your outsourced projects come in on time, but if the delay is truly out of your control, this broader language may have you in a better position with your Publisher.

Manage the process relative to your obligations.

- Coordinate payments to match your Milestone Schedule. Be sure that the payment terms for your contractors are at least one month after the applicable Publisher milestone funds are due. Planning for Publisher delays, rejection and re-submission while still avoiding late payment to your Contractor is the ideal situation. You don't ever want to be in a position where you owe a contractor but the funds aren't due from the Publisher for another month. When negotiating your Contractor agreements, refer to your Publisher agreement to be sure they match up. An alternative to this is to provide that payment to the contractor will not be due until ten days after receipt of the corresponding Publisher milestone payment.
- Avoid situations that could put you in a hole elsewhere. Front-loaded Contractor agreements that pay at the outset simply won't work well if your Publisher deal is light on advances at the outset with a balloon payment at the end of development. You're going to end up tight on cash throughout development, causing you to cut corners on staff and overhead. Similarly, if all your Contractors are entitled to late

fees for non-payment, this will cut into your Publisher funds. Again, negotiate your Contractor agreements with an eye towards your Publisher obligations.

- Act like a Publisher when dealing with your Contractors. In a bit of role reversal, you are now the paying customer. You bear the risk of the Contractor's non-performance. Manage them accordingly. Check in early and often to assess progress. A sure-fire recipe for disaster is to out-source some area of development, assume it will be done properly and on-time only to find out at your deadline that the Contractor's submission will need another two weeks to integrate. Stay on top of them on a regular basis to make sure their efforts sync up with your Publisher-driven schedule.
- “Manage” your Publisher. Keep the Publisher in the loop with respect to all contractor progress. They've already approved the Contractor...now let them monitor progress through you. Weekly updates regarding all active outsourced projects will be appreciated. Also, if (when) something goes wrong, the Publisher won't be shocked and the consequences may be less severe. As you know, Publishers don't like surprises.

#### Choose the right outsourcing partner.

Although not a “legal” solution, working with a solid, well-known, well-respected company for your outsourcing needs will solve or avoid many pitfalls. Whether it is a music composer or an outsourced port, knowing that your partner is as committed to your project as you are will go a long way. Availability, budget, reliability and results will all be slightly less stressful. If you don't have personal experience with a potential outsourcing company, ask around. Publishers, attorneys, agents, and your peers in the development community are likely to have a quality referral.

#### **Conclusion**

A Developer's quality of life is impacted when time, funds and resources are stretched to the limit. Deadlines and blown budgets often spiral out of control, impacting morale and product quality. Something will inevitably go wrong with your next project. Despite the value offered by quality outsourcing of assets, this is certainly an area where things can go awry. But if you proactively manage the process on all fronts, you might save yourself and your team some unnecessary angst and preserve your quality of life.

## Acceptance of Deliverables

By Thomas H. Buscaglia, © 2006

### Introduction

The acceptance of Deliverables by the Publisher triggers the payment of the Advances set out in the Milestone schedule. As each Deliverable is submitted and then approved, the correlative funding Advance is released. In this manner, the Publisher controls its risk by confirming that the project is on track before they continue funding development of the game. However, delays in the acceptance process can create serious problems for the Developer. Unreasonable delays result in longer development cycles that in turn result in more operational overhead and tighter schedules if not planned for in advance. These added pressures on the Developer can have a serious detrimental impact on both the quality of life of the development team and the financial stability of the studio.

### Sample Clause

Here is what a typical Acceptance provision in a Developer/Publisher contract might look like:

1. *Acceptance:*
  - a. *Acceptance by PUBLISHER*
    - i. *After DEVELOPER submits to PUBLISHER a Deliverable Item, PUBLISHER shall have ten (10) business days to examine and test such Deliverable Item to determine whether it is acceptable to PUBLISHER.*
    - ii. *PUBLISHER shall notify DEVELOPER of PUBLISHER's acceptance or rejection of such Deliverable item and, in the case of any rejection, shall provide DEVELOPER with a reasonably detailed list of specific elements of noncompliance with the delivery schedule in such Deliverable Item.*
    - iii. *No Deliverable Item shall be deemed accepted by PUBLISHER unless and until expressly accepted in writing by PUBLISHER, except as provided below. The approval of any game content by PUBLISHER is subject to its sole discretion. Payment by PUBLISHER of a Milestone according the payment schedule shall not, under any circumstances, be considered as acceptance by PUBLISHER of any Deliverable Item as may be included in the said*

*Milestone. If PUBLISHER fails to notify DEVELOPER of its decision within such ten (10) days period, then DEVELOPER may notify PUBLISHER of this fact and PUBLISHER shall have five (5) working days from receipt of the notice to accept or reject the Deliverable Item. Failure by PUBLISHER to respond within such five (5) working days period shall cause the Deliverable Item to be deemed accepted.*

- iv. *In the case of a rejection, DEVELOPER shall use diligent efforts to correct the deficiencies and shall resubmit the Deliverable Item, as correct, within fifteen (15) business days of PUBLISHER’s rejection.*
- v. *In the event that DEVELOPER fails to deliver an acceptable Deliverable Item with respect to the Game within such fifteen (15) day period following rejection, PUBLISHER may terminate the Agreement DEVELOPER subject to the termination provisions of this Agreement. The procedure as defined in paragraphs I. through v. above shall iterate until PUBLISHER either accepts the Deliverable Item or terminates the Agreement.*
- vi. *Notwithstanding the foregoing, PUBLISHER shall have no right to terminate the Agreement pursuant to this section if DEVELOPER’s delay with respect to delivery of Deliverable Items is caused (I) by PUBLISHER late delivery of items pursuant to the Agreement and/or (ii) by DEVELOPER due to Force Majeure as defined elsewhere in this Agreement.*
- vii. *Final Acceptance of any Deliverable will be the sole discretion of PUBLISHER.*

## **Discussion**

Timing and control are key areas of concern for Developers. Let’s consider the impact of delay first. If there are only five Milestones in a contract and the acceptance procedure results in only a two-week stall, the resulting impact on the development process is a month and a half of delay. That is either an additional month and a half of overhead, which will come right out of profits if there are any, or several months of 60 hour plus weeks for the team. Some delay in the acceptance process is expected and should be anticipated and included in the production schedule and budget. But unexpected extended delays in the acceptance process are often not taken into account. This is where the second consideration, control, comes into play.

The procedures used to determine the acceptance of each Deliverable by the Publisher are critical to trigger the payment obligation for the Advances. These procedures usually

include a time period for the Publisher to review and either accept or reject the submission. Of course, in negotiation the Developer should endeavor to make these time periods as short as possible. But what happens if the Publisher does not act within that time period? Since the Publisher usually drafts these agreements, you can expect the bias of these provisions to be entirely in the Publisher's favor. This means that usually the Acceptance must be in writing and if it is not given in writing, it does not happen. In many cases, no action on the part of the Publisher results in a situation where there is no rejection, no acceptance and no remedy for the Developer. The Developer and the game are in limbo.

The perception is that since publishers have money, they are competent efficient business operations, and many are. Unfortunately, some are not. And even if the one you have a deal with is not one of the latter, you do not want to be dependent on the Publisher taking an affirmative action, especially in writing, for Acceptance of your Deliverable. Sadly, if it is to the Publisher's advantage to stall the Advance payment, or even if they just do not have the time or the focus to deal with it, your Deliverable could end up laying around for a month or two before they get around to checking it out. Moreover, this scenario is seldom addressed in the breach provisions of the contract.

The Acceptance provisions should always include a procedure whereby the failure of the Publisher to act by either accepting or rejecting the Deliverable within the set time period is deemed Acceptance. This can occur either without a notice provision or with one followed by the automatic acceptance, as in the above Sample Clause. Of course, the Publisher still has to send the money. But at least at that point they are contractually obliged to do so.

In addition, often the Acceptance section of the agreement will contain a provision that states that the payment of an Advance is not Acceptance. This looks like it would work to a Developer's advantage by allowing a Publisher to advance funds even though the Deliverable is not yet accepted. But though this may at first blush sound good to the Developer, it isn't. Ultimately, it puts control of the Developer's destiny even more in the hands of the Publisher than it already is. In effect, the Developer is at a tremendous disadvantage by allowing the development of the game to proceed while the Developer is in a continuing state of breach due to their failure to deliver an acceptable Deliverable. Any time after that the Publisher can terminate the contract for cause (this issue was discussed in Contract Walk-Through Release 2). So, it is best to have the Advance equate to Acceptance.

Similarly, often a Publisher will accept a deliverable even though it is not consistent with the description set out in the Milestone Schedule. This usually occurs for several very good reasons including keeping the game development on track and allowing for creative iteration of the game. But when this does happen, the Deliverable schedule should be amended in writing so that the Deliverables actually accepted are reflected in the amended delivery schedule and the revised deliverables are incorporated into the subsequent deliverable descriptions. This way, the contract reflects what is actually occurring, not what was the "best guess" at the start of the project months or even years earlier. Again,

this avoids the Developer being in a constant state of breach and subject to termination for cause at the whim on the Publisher.

An additional difficulty in the approval process occurs when there is a third party licensee in control of the project content. Often the final approval of the milestone is dependent on the IP owner that licensed that IP to your Publisher. If the terms of that license provide this third party with absolute control of the content, the Developer must make sure that additional delays in the approval process are taken into account in the terms of the approval provisions. It should also be taken into account in the budget.

### **Conclusion**

The Developer needs to consider the Acceptance procedures set out in the proposed Developer/Publisher contract carefully. Anticipate and consider all of the possible scenarios that could occur in order to account for and appropriately address adverse potential situations in advance. Remember, Publishers will not offer to protect the Developer's interests. That's not their job. It's yours. So, it is up to the individual negotiating the contract on behalf of the Developer to make sure that the contract, including the Acceptance procedure, always takes into account the Developer's best interests.

## Contributors

### Tom Buscaglia

Tom Buscaglia, [The Game Attorney](#), practices law in Miami, Florida. Tom is dedicated to the computer and video game industry, assisting developers in all aspects of their legal and business needs and has been representing game developers from around the world since 1991. Tom is a member of the Board of Directors, as well as the Coordinator of the [South Florida Chapter](#), of the IGDA. He has published numerous articles to assist those wishing to start their own game development studios and recently launched [GameDevKit.com](#) to further help start up game developers. Tom is a perennial presenter at the annual Game Developers Conference and the Indie Games Com and speaks at numerous other game related conferences. He also speaks regularly on the use of games for beneficial applications and is the Executive Director of the Interactive Entertainment Institute, the presenters of the [G.A.M.E.S. Synergy Summit](#), and of [Games Florida](#). As FaTe[F8S] Tom plays online on a regular basis as the Supreme Warlord of [FaTe's Minions](#). Tom has a gamer's appreciation and understanding of games and the game industry.

### Jim Charne, chair

Jim Charne is a California, New York, and New Jersey lawyer who has provided legal representation for clients in all phases of interactive software entertainment since the mid-1980's. Jim has been chair of the Legal and Business tutorial at GDC since 1998, and writes "Famous Last Words," a monthly column on games contracting issues found at [www.igda.org/columns/lastwords](http://www.igda.org/columns/lastwords). He entered the industry in 1983 as a producer for Activision, and served as VP Legal & Business of console developer Absolute Entertainment. Jim is a member of IGDA, AIAS, G.A.N.G., AIMP (Association of Independent Music Publishers), SAG (Screen Actors Guild) and NARAS (the Recording Academy), is on the board of advisors of G.A.N.G., and served as the first President of the Academy of Interactive Arts and Sciences (AIAS) from 1998 to early 2001. Jim can be reached by email at [charne@usa.net](mailto:charne@usa.net) or on the web at [www.charnelaw.com](http://www.charnelaw.com).

### David Rosenbaum

David Rosenbaum has a wide-ranging practice representing entertainment clients. David regularly handles transactions involving development, financing, publication and distribution of interactive entertainment properties and related technology licensing; mergers and acquisitions in the interactive industry; licensing and merchandising of brands, character, comic book, entertainment and sports properties; production, marketing and distribution of motion pictures and television productions; and book publishing and licensing. David has negotiated agreements with leading video game hardware manufacturers and publishers, film studios, TV networks, sports leagues and book

publishers. David lectures frequently on legal and business issues in developing and publishing video games, including at events sponsored by the Game Developers Conference, E3, Australian Game Developers Conference and the California Lawyers for the Arts. David's practice is international in scope, having been retained by UK, Canadian, Australian and New Zealand companies to represent their interests in the US and representing US-based companies around the world.

### **Vincent Scheurer**

Vincent Scheurer is a London-based lawyer specializing in video game contracts. He has worked in the games industry since 1997, when he joined a UK games publisher as sole in house counsel. After spending five years at law firm Osborne Clarke, Vincent set up Sarassin LLP, a legal consultancy in the video games industry, in 2004. Sarassin LLP advises on all types of commercial agreements and related issues in the video games industry. Sarassin is also the Company Secretary of TIGA, the UK game development studios' trade association.

### **Barry Seaton**

Barry Seaton is a principal of Barry C. Seaton, P.C., located in Irvine, California. The firm counsels developers, publishers and other game companies and individuals in the video game, entertainment, Internet and other media industries, including major Hollywood studios. Mr. Seaton has been an intellectual property and transactions attorney for 19 years, including serving as Chief In-House Counsel for Crave Entertainment Group, and in the video game/licensing department at Fox Entertainment.

### **Patrick Sweeney**

Patrick Sweeney is a Partner at Haight, Brown and Bonesteel LLP, where he heads the firm's video game and multimedia practice, which focuses on the representation of game developers, publishers, licensors and other industry-specific clients. In addition, Mr. Sweeney has written various articles for game-related publications and is an adjunct Professor at both Whittier Law School and Southwestern Law School, teaching Video Game Law, the first US law school course solely dedicated to the legal issues of the video game industry. Please contact Mr. Sweeney with any comments/feedback to this article at [psweeney@hbblaw.com](mailto:psweeney@hbblaw.com).