



Quality of Life in the Game Industry: Challenges and Best Practices

April 20, 2004
www.igda.org/qol

© 2004 IGDA. All rights reserved.

Table of Contents

<u>SUMMARY & CONCLUSIONS</u>	5
PREAMBLE	5
CONCLUSIONS ON IMPROVING QUALITY OF LIFE	6
ABOUT THE QUALITY OF LIFE COMMITTEE	7
ABOUT THE IGDA	7
ABOUT THE AUTHORS	8
ACKNOWLEDGEMENTS	9
<u>INTRODUCTION</u>	10
PREAMBLE	10
A UNIVERSAL ISSUE	10
SUSTAINING QUALITY OF LIFE	11
DAMAGING QUALITY OF LIFE	12
THE QUALITY OF WORK	12
WORK AND LIFE: FRIENDS OR FOES?	13
WHEN THE WORST HAPPENS	13
<u>THE IGDA QUALITY OF LIFE SURVEY</u>	15
JUSTIFICATION	15
DEMOGRAPHIC PROFILE	15
THE FIRST STEPS	16
EXPERIENCE LEVELS	16
OVERALL, A HAPPY BUNCH...	16
LONG HOURS	18
INADEQUATE STAFFING	19
WORK ORGANIZATION PROBLEMS	19
GAMES AND COMMUNITY	20
CREDITS	20
<u>THE BALANCING ACT: A GAME CAREER VS. ‘A LIFE’</u>	21
FROM THE INTERVIEWS...	21
FROM THE SURVEY...	21
PREAMBLE	21
WHY ACHIEVING BALANCE IS HARD	22
CONSEQUENCES	26

BEST PRACTICES	28
<u>LONG HOURS AND EXTENDED CRUNCHES</u>	30
FROM THE INTERVIEWS AND THE MESSAGE BOARDS...	30
FROM THE SURVEY...	30
PREAMBLE	31
CAUSES	32
CONSEQUENCES	33
BEST PRACTICES	34
<u>HIGH TURNOVER AND JOB INSTABILITY</u>	42
FROM THE INTERVIEWS...	42
FROM THE SURVEY...	42
PREAMBLE	42
CAUSES	42
CONSEQUENCES	44
BEST PRACTICES	46
<u>WORK ORGANIZATION ISSUES</u>	48
FROM THE INTERVIEWS AND THE MESSAGE BOARDS...	48
FROM THE SURVEY...	48
PREAMBLE	48
CAUSES	49
CONSEQUENCES	51
BEST PRACTICES	52
<u>LESSONS FROM OTHER CREATIVE INDUSTRIES</u>	58
TRANSITIONING FROM OTHER INDUSTRIES	58
LESSONS FROM SOFTWARE ENGINEERING	60
OTHER INDUSTRIES' MODELS	63
<u>OWNERSHIP OF WORK</u>	64
PREAMBLE	64
WORK DONE FOR OTHERS DURING TIME OFF	64
AN EMPLOYEE'S ORIGINAL CREATION	64
ORIGINAL WORKS TO WHICH THE DEVELOPER OWNS IP RIGHTS	65

ON CONTRACTED “WORK-FOR-HIRE” PROJECTS	66
FOR MORE INFORMATION	66
<u>RELATED WORK FROM OTHER IGDA INITIATIVES</u>	<u>67</u>
BEST PRACTICE REPORTS	67
INTELLECTUAL PROPERTY RIGHTS COMMITTEE – BEST PRACTICES	68
CONTRACTUAL BEST PRACTICES	68
CREDIT STANDARDS COMMITTEE	68
WOMEN IN GAME DEVELOPMENT SIG	69
HUMAN RESOURCES SIG	69
IGDA CHAPTERS	69
<u>APPENDIX A - BIBLIOGRAPHY</u>	<u>70</u>
BOOKS	70
MAGAZINE AND NEWSPAPER ARTICLES	70
ONLINE DOCUMENTS	71
<u>APPENDIX B - INTERVIEW TRANSCRIPTS</u>	<u>72</u>
PREAMBLE	72
INTERVIEW #1: CTO OF A SMALL STUDIO (FRANCE)	72
INTERVIEW #2: PROGRAMMER (ORLANDO, FLORIDA)	73
INTERVIEW #3: PROGRAMMER (CANADA)	75
INTERVIEW #4: PROGRAMMER (BRAZIL)	77
INTERVIEW #5: QA TESTER FOR A LARGE PUBLISHER (UNITED STATES)	80
<u>APPENDIX C – QUESTIONNAIRE AND RAW DATA</u>	<u>83</u>
PREAMBLE	83
QUESTIONNAIRE	83
RAW DATA	90

SUMMARY & CONCLUSIONS

Preamble

The workplace is a stressful one everywhere, more so in the game industry than in most others. Long hours are endemic. Most projects fail in the marketplace. High-profile studio closures are announced every month. And while the work we do is stimulating and rewarding in its own right, there is no reason not to make our working conditions a lot better than they are today.

For young career-oriented people fresh out of school, our industry's shortcomings, with its endemic long hours and 95% marketplace failure rate, may not always seem obvious or crippling. But after a few years, all-nighters fuelled by coffee and pizza lose much of their appeal. Most people come to want significant relationships, a more balanced life, and sometimes, children as well. None of this is easy to achieve in the typical game company. As a result, many experienced developers fed up with the crunch cycle decide to leave the industry, taking with them a wealth of talent and experience that we can ill afford to lose.

Some within the industry consider long hours, high pressure and generally poor quality of life as normal, or even as signs of strength to be bragged about. We do not. No enlightened company should, either: one of the surprising findings of the IGDA Quality of Life Survey is that the majority of game developers now consider games to be only one of many valid career choices for them, not the be-all and end-all that industry folklore still holds dear. As projects get ever bigger and we have to compete with companies like Qualcomm, HP and Pixar for talent, we just can't afford to drive our best people away by offering comparatively mediocre quality of life conditions.

Contents of the White Paper

This white paper, prepared by a group of developers representing all game development professions and all types of companies, discusses the quality of life issues most common in our industry, their causes, and some of the solutions elaborated by enlightened individuals and corporations. Its core contains 6 major sections:

- **Introduction** sets the canvas by looking at the state of the contemporary workplace in general.
- **The IGDA Quality of Life Survey** analyzes the results of the survey we held in early 2004 and to which nearly a thousand developers responded.
- **The Balancing Act** describes the challenges that developers face when trying to balance a quality personal life with a fulfilling game development career. Topics discussed include the industry's high-risk economics, scheduling, peer pressures, lack of concern and support for the family unit in game companies, community resistance to controversial game content, family tensions and early departures toward rival industries.

- **Long Hours and Extended Crunches** discusses the most obvious drawback of a game development career, how we ignore the problem and chalk it up to passion or the natural order of things, and the losses of experience and productivity that “forced workaholism” inflict upon our companies. We also talk about best practices in scheduling, budgeting, project management methodology and human resource management that may help teams alleviate the problem.
- **High Turnover and Job Instability** talks about the job insecurity associated with hiring/layoff cycles and poor development contracts, and the lack of standard job descriptions that make it very difficult to move from one company to another without extensive re-training and ramping-up costs.
- **Work Organization** discusses the industry’s weaknesses when it comes to management training, haphazard change control policies, poor compensation schemes, credit allocation, and rivalries between developers and management, all of which can compound the long hours problem and damage team integrity.

The rest of the white paper contains supporting materials on related topics: lessons that we can learn from other creative industries, and related work from other IGDA committees.

The appendixes an extensive bibliography, transcripts of several interviews we conducted while preparing this paper and the text of the IGDA Quality of Life Survey.

Conclusions on Improving Quality of Life

By and large, game development is stimulating work, better than most alternatives. However, it is all too often performed in crippling conditions that make it hard to sustain quality of life and lead too many senior developers to leave the industry before they have had time to perform their best work.

It is our moral duty and in our best interest to do all that we can to alleviate the quality of life problems endemic to our industry. This white paper has identified several ways to do so:

- Family-friendly practices.
- A conscious effort to minimize overtime.
- Better communication between management and developers.
- Better contracts between individuals, studios, and publishers.
- Better planning and budgeting.
- Better project management practices.
- Better human resource management.

We understand that perfection is impossible to achieve. However, that is no reason not to try, especially since a reasonable effort will usually achieve a great deal of results. Go out and spread the word!

Educate the young kids just starting out. Tell them that working 36-hour shifts and installing tents around their cubicles so they never have to go home is unhealthy, and that even if they are able to handle it at 21, they won't find that sort of environment anywhere near as appealing at 30. Tell them that unless we break the cycle, they will find themselves in an untenable situation once they start wanting serious relationships and families.

Educate your managers. Tell them that driving their most effective senior employees out of the industry through impossible working conditions is stupid. Tell them that endemic crunches, slippage, cancellations and studio closures are NOT normal parts of developing games, but consequences of conscious policies that perpetuate error by depriving ourselves of our hard-won experience.

Educate your peers. Tell them that producers and managers are not always the enemy, and that a great deal can be accomplished through honest communication and respect.

Educate the publishers. Tell them that there is a whole lot more money to be made from predictable projects that lead to quality products than from haphazard, try-everything-and-see-what-sticks lemon-squeezing. Tell them that current industry practice may have worked in the days of \$50,000 budgets, and that we have survived it to this day, but that no one (not developers and certainly not publishers themselves) is in any position to sustain a 95% failure rate on the \$25 million projects that are just around the corner.

About the Quality of Life Committee

The IGDA Quality of Life Committee's mission is to make the game development experience smoother and more pleasant for everyone involved, therefore extending developers' careers and helping companies make better games in easier fashion.

We are always looking for volunteers willing to help out in our ongoing efforts. To learn more about us, visit our web site at <http://www.igda.org/qol> or write us at qol@igda.org.

About the IGDA

The International Game Developers Association is a non-profit membership organization that advocates globally on issues related to digital game creation. The IGDA's mission is to strengthen the international game development community and effect change to benefit that community. For more information on the IGDA, please visit www.igda.org or e-mail info@igda.org.

About the Authors

Scott Bonds

Scott is a producer at Electronic Arts. Before joining the game industry, he was a "serial entrepreneur" who founded three different companies and developed business software for clients such as Microsoft and Siemens.

Jamie Briant

The director of research and development at interactive fitness company Exertris, Jamie was also one of two senior programmers responsible for the *Tony Hawk Pro Skater 3* engine at Neversoft, and he led the team that brought *Resident Evil 2* to the Nintendo 64 platform. His first hit game, *Axis*, was published in 1993.

Dustin Clingman

Dustin is the president of Zeitgeist Games, which he co-founded with *Dungeons and Dragons'* co-creator Dave Arneson. As an engineer, he worked on several handheld games at Perpetual Motion Interactive. He also teaches at Full Sail and coordinates the IGDA's Orlando chapter.

Hank Howie

Hank is the president of Blue Fang Games, the studio responsible for the hits *Zoo Tycoon*, *Dinosaur Digs* and *Marine Mania*. He joined the industry in 1989, as producer of *Riders of Rohan*, and later managed Ziff-Davis' online gaming service, Interchange GameSpace.

François Dominic Laramée

The IGDA Quality of Life Committee's chair, François Dominic is a freelance designer and programmer with over 20 console, interactive TV, PC and board games to his credit. He also writes for television and standup comedians, and has published two game development books and over 100 articles.

Greg LoPiccolo

Greg is vice-president of product development at Harmonix Music Systems, the company responsible for the PS2 games *FreQuency*, *Amplitude* and *Karaoke Revolution*. Before joining Harmonix, he spent several years at Looking Glass studios, a stint which he ended as project lead on the acclaimed *Thief*.

Andy Luckey

Andy began working on games in the early 1990's. Having started as an artist at his family's animation studio in the 1970's, he is now the President of Greater Family, LLC, producing family oriented media. His credits include *Sesame Street*, *Teenage Mutant Ninja Turtles* and Disney/Virgin's *Aladdin*. A former youth pastor, he is a part-time member of the ministry.

Mike McShaffry

Mike spent seven years at Origin Systems, working on *Martian Dreams* and several *Ultima* titles, before leaving to start his own casual game studio, which was later acquired by a publisher. He is the author of *Game Coding Complete*.

Acknowledgements

The authors would like to thank the developers who took time to answer our survey and interview questions. This effort would not have the same credibility without you.

And special thanks to Liz Wakefield at the IGDA head office for implementing the web survey and providing us with the results.

INTRODUCTION

Preamble

Welcome to the IGDA's 2004 Quality of Life white paper. We hope that you will find it useful.

With its long hours, high pressure and inherent instability, the game industry makes it very difficult to lead a balanced life. This document aims to address the issue by discussing the consequences of the poor quality of life endemic to the game industry and identifying solutions that can help alleviate the problem.

A Universal Issue

While quality of life problems may be particularly strident in the game industry, we do not hold a monopoly on them by any stretch of the imagination.

Everyone is Stressed Out

- More than half of the European Union's 147 million workers report working at very high speeds and to tight deadlines [Docherty02].
- Two thirds of the respondents to *Workplace 2000*, a national survey of Canadians [Lowe00], said that their jobs were "somewhat" or "very" stressful. Among the most common complaints were heavy workloads, long hours, lack of involvement and recognition, rigid work structures that don't let individuals contribute to the fullest extent of their talents, and underemployment.
- A Health Quebec study [Galipeau03.1] reports that 25.5% of workers experienced job-related psychological distress of one kind or another in 1993. The young, often burdened by heavy student debt and of whom companies demand incredible amounts of effort and performance in exchange for no job security, are especially at risk. Industries like ours, where personal investment is so strong, compound the problem.

Everyone Works Too Much

- On average, Americans work 350 hours more per year than Europeans, the equivalent of nearly 9 full weeks. [Galipeau03.2] A coalition of universities and community organizations has created Take Back Your Time Day, on October 24th (precisely 9 weeks before the end of the year), in protest.
- A March 2003 survey of Quebec workers revealed that 68% of them would gladly trade away some of their income for an extra day off every week [Galipeau03.2]; in the United States, the proportion is 52% according to a similar survey held in August 2003 by the Center For a New American Dream, which further reveals

that 83% of Americans believe they don't have enough time to spend with their families.

- The average American worker only gets two weeks of unpaid vacation time a year, versus five or six paid weeks for his European counterpart. Since 1999, the European Union enforces a minimum of four weeks of paid vacation time that can't be traded in for extra money [Désiront03]. And yet, Désiront says, several European economies outperform the U.S., because productivity (measured as output per hour worked) is reinforced when workers are well rested.
- In North America, overtime is largely unregulated and, in the game industry, unpaid. Furthermore, legislation introduced in Congress in 2003 would make it easier for employers to declare more employees exempt from what rules do exist.
- According to HR professionals quoted in [Tremblay03.1], a compressed work week during the summer months, with employees getting Friday afternoons off, is by far the most popular and least expensive way to conciliate work and family; it also improves morale and makes hiring and personnel retention easier.

Sustaining Quality of Life

In 1997, a coalition of community organizations in Alberta, Canada, organized a Quality of Life commission [Lowe00] that identified 6 key contributing factors to high quality of life in the workplace and elsewhere:

- Meeting basic necessities
- Hope
- Self-determination
- Health and well-being
- Security
- Community

Hope and self-determination, specifically the chance to apply one's skills in meaningful ways to projects one cares about, the game industry provides in plenty. And while some other related industries may pay better, there is no doubt that developers are able to put bread on the table once they find regular employment.

However, long crunches, frequent periods of intensive stress, and a hit-driven industry in which high-profile project cancellations and studio closures happen almost on a weekly basis don't support much in the way of mental and physical health, nor a sense of security. As for community, game development itself is a remarkably friendly brotherhood, but the opinion in which the general public holds us all too often ranges from amused contempt ("When are you going to get a real job?") to outright hostility ("This ultra-violent worthless tripe you do should be outlawed, you crazy sociopath!")

Damaging Quality of Life

In [Docherty02], co-author Mari Kira lists five reasons why work can become "consuming" instead of "regenerating":

- Lack of resources: Some organizations are just chronically understaffed.
- Self-intensification: In other words, driven individuals taking on too much responsibility and pushing themselves beyond their limits.
- Excessive complexity: Jobs that require more skills or resources than the individual can bring to bear.
- Lack of regenerative processes: No time for learning or rest.
- Not enough complexity: Jobs that don't take advantage of the individual's skills are as draining as those that overwhelm them.

While lack of complexity is very unusual in the game industry, except maybe at the entry level, the other factors of consumption are the norm in our business, not the exception. There never are enough experienced professionals for all of the projects going, so "newbies" and junior developers are pushed into roles for which they are not prepared; exciting, yes, but increasingly stressful as the milestones begin to slip and Christmas approaches.

The Quality of Work

That being said, it is important to remember that one of the most important determining factors in quality of life at work is the quality of the work itself. In that respect, game development is far and away superior to most of the alternatives.

According to University of Sherbrooke psychologist Charles-Henri Amherdt [Galipeau03.2], over 80% of workers consider their jobs to be a source of pain and suffering, usually because the work provides insufficient intellectual stimulation – hardly a problem that we need to worry about!

Quality of Work and Diversification of the Game Development Community

On the other hand, game studios are competing for personnel with other knowledge industries, like commercial software development and movie special effects, which some people may find equally stimulating and less overly demanding. As the industry and its projects grow in scope, we can no longer afford to hire only those few people who can't imagine doing anything else; we must be competitive in the general marketplace. Failing to address this issue will only worsen the industry's chronic penury of experienced talent.

It should be further noted that talent drain is likely to increase as the game development population diversifies. Recent experiences in the Quebec medical community underline this point: young female doctors outnumber their male counterparts by a wide margin, and they simply refuse to work the extreme hours that older male doctors routinely accepted.

Work and Life: Friends or Foes?

As [Friedman98] notes, work and personal life are often treated as competing, when they are in fact complementary:

- A company that helps its employees achieve a stable balance reaps rewards of trust and loyalty;
- Which breed personal investment in the company's success;
- Which leads to better job performance;
- Which increases the odds of profits;
- Which create resources that can in turn be reinvested in helping sustain the balance.

When the Worst Happens

Many developers have absorbed (and benefited from) information from books, courses and seminars about the software development process, which asserts that rigorous application of good development practices can completely eliminate crunch times and associated employee burnout.

This flies in the face of the reality that many developers have experienced.

Perfect Control is Implausible...

To put the lessons of this white paper into context, it may come to pass that, even with the best intentions and practices in place, the team will need to work extra evenings and weekends to meet a milestone or ship a product. Indeed, the nature of game development schedules is such that it maximizes the likelihood of such an outcome:

- Game development is a creative endeavor, wherein the final outcome of a project is rarely fully known at the outset.
- It is often true that the ship date is impossible to delay without catastrophic consequences like cancellation or even bankruptcy.
- Significant chunks of a project's schedule may be outside the developer's control. Licensed assets may be late or never materialize. Third-party tools and libraries may be late, buggy, or both. Publishers may force late-stage design changes.

These factors can put even the best of developers in do-or-die situations, in which case "do" generally trumps "die" no matter what the theory says.

... But Not Really Necessary

So why bother working on this at all, if the deck is stacked against us? For two reasons:

- There are development practices, described herein, that can both drastically reduce the likelihood of crunch time and other sources of quality of life deterioration, and minimize the negative impact of what can't be avoided.
- If management is aggressively pursuing positive quality of life policies, and is perceived by the staff as doing so, the staff is much more likely to a) cooperate with management to reduce risk and overtime, and b) take in stride whatever little crunch does materialize without becoming bitter and despondent.

Most developers are realistic in their expectations. If they feel that management has their best interests at heart, they will deal with any imperfect situations that may arise. Conversely, if developers believe that management has dealt with a project incompetently, or that it simply doesn't care about their fates, then the company is in trouble.

Some companies do in fact manage to completely avoid employee burnout. Good for them; they serve as an inspiration for the rest of us. But even if your company doesn't manage to achieve perfect quality of life, aggressively striving toward it will earn your team most of the advantages of perfection. In this area, there are no small gains.

THE IGDA QUALITY OF LIFE SURVEY

Justification

In order to support our research and orient our action for 2004-2005, we conducted a survey of game developers through the IGDA web site.

While this survey is not scientific, in the sense that the respondents were self-selected instead of being randomly chosen from among the entire population of IGDA members or game developers, the number of responses was so high (994) that we believe the results to be significant.

The questionnaire used in the survey can be found in Appendix C, and the raw results database will be made available from the IGDA web site via www.igda.org/qol/.

Demographic Profile

The survey paints a picture of the average developer that contains few surprises. The average respondent is:

- **Male.** Of those respondents who chose to answer the question about their gender, 92.9% are men and only 7.1% are women.
- **Young.** 33.8% of respondents are between 25 and 29, only 18.4% over 35.
- **In the early stages of his/her career.** 74.4% of respondents have been in the industry for 8 years or less, with 2-5 years being the most common response.
- **Less likely than the population as a whole to have children.** 76.9% of respondents (and 82.9% of female respondents) have no kids.

About a third (34.1%) of respondents are single and unattached, which is actually *lower* than, for example, in the Canadian population as a whole (42.4% in 2002 according to the Statistics Canada web site). About 44.5% are married or living with a partner, and 20.3% are single but in a serious relationship.

However, contrary to expectations, more people said that games were only one of many career options for them (34%) than said games were their only choice (32%). As the industry grows, the proportion of “non-fanatics” among game developers is likely to increase.

Also worth noting: in analyzing the answers to the survey, there is no clear, statistically significant difference between marital status and age groups, smaller sample sizes notwithstanding, except for the trivial cases (i.e., no one in the group with less than 2 years of experience has ever been with the same company for five years or more.) The same appears to be true when considering employment status (full-time, freelance, independent, etc.) Some differences emerged when comparing male and female respondents, but the small sample of female participants may have skewed the results.

The First Steps

It seems that the first year in the industry is difficult for a significant minority of developers. While most respondents said they had no trouble finding their first job (“easy” got twice as many votes as “moderately hard” and “hard” combined), 30% stated that they often felt they were paying their dues with grunt work during their first year and 18% said they even considered leaving the industry at some point during that first year.

The fact that only 25% of respondents had a mentor when they first joined the industry may explain at least part of this phenomenon, although the evidence that can be inferred from our numbers is slim (20% of developers without mentors considered quitting the industry, against 16.8% of those who had one).

Experience Levels

As is to be expected in an industry where the majority of workers are young, the average experience levels of developers are much lower than in other areas.

Experience of Rank-and-File Developers

56% of respondents said that their coworkers had, on average, 2 to 5 years of game development experience, while 24.6% said 5 to 10 years. **Only 3.4%, or roughly one in 30, said that their coworkers averaged 10 or more years of experience.**

Compared to other professional endeavours (e.g., law, accounting, engineering, business software), these numbers are very low.

Experience of Leads

Understandably, leads tend to have more game development experience than their charges: 52.3% of respondents said that people in positions of leadership at their studio had 5 to 10 years of experience.

However, 26.6% said that their leads had 2-5 years of experience, while only 9.8% answered “more than 10 years”. On the positive side, very few industries provide so many opportunities for advancement so early in a person’s career. On the other hand, the fact that **fewer than one lead developer in 10 has over ten years of experience** indicates that we lose a depressingly high proportion of our senior people to rival industries before they have had a chance to do their best work.

Overall, A Happy Bunch...

Most game developers work in this industry because they love to make games, and their responses reflect this fact:

- When asked to rate their overall satisfaction with their game development career, respondents **chose 7 or 8 about 46% of the time, and 9 or 10 about 16% of the time.** The proportion who rated their career as a 5 out of 10 or less, while significant (27.2%), is much lower. Interestingly, the level of satisfaction tends to

rise with seniority: 7.9 on average for developers with 10 years of experience or more, 7.0 for those with 5-10 years, 6.8 for those with 2-5 years, 6.0 for those with 1-2 years, and 5.6 for those with less than a year. Self-selection is undoubtedly at play here, as satisfied people are more likely to stick around for a long time.

- Not surprisingly, the average level of satisfaction is higher among developers who have projects lined up ahead of time (7.0 out of 10) and those working in companies that keep staff on salary during down time (6.6) than among developers living with a project-based hiring/layoff cycle (5.7 out of 10).
- The typical developer has stayed with the same company for 2 to 5 years (42.9%) or over 5 years (24.3%) at least once in their career.
- Over 55.9% of respondents said they knew where their next project would be coming from even before their current one was completed, while an additional 23.9% said that the company kept its employees on the payroll during downtime between projects. Fewer than one developer in six reported working in a company that hires and lays off by project.
- Respondents are satisfied with the level of challenge their jobs provide, with 83% saying the job is constantly or usually stimulating or that they are in no hurry to change even though they would be ready for new challenges. Only 12% of respondents feel overqualified for their current duties, while 2% feel overwhelmed.
- Finally, 63.1% of respondents said they had never been laid off from a game job (18.9% said they had lost a job when their studio had gone out of business) and 65.1% have never quit in the middle of a project.

... But not forever

However, while the most popular answer to the question: “How long do you plan to stay involved in the game industry?” was “for the rest of my career” (47.2% of respondents), **34.3% expect to leave the industry within 5 years**, and 51.2%, within 10 years. For the industry as a whole, such a high turnover rate is nothing short of catastrophic, and it goes a long way towards explaining our difficulty in ensuring that our projects run smoothly.

No clear tendency emerged from the answers to our question about whether developers had a definite career plan. All of the choices received roughly equivalent numbers of votes, with “No, we’ll just see what comes” finishing slightly ahead at 29.4%.

Outside Pressures

The answers to our survey hint that **the game industry may cause a significant amount of friction in developers’ relationships and families.**

When we asked our respondents to examine a set of assertions and pick all of those which their spouses might be likely to use to qualify their game development careers, the most popular answers were:

- “You work too much and don’t spend enough time with me and the children.” (61.5% of respondents)
- “You are always stressed out.” (43.5%)
- “You don’t make enough money.” (35.6%)

Active support from their partners is less common, with only 26.3% of respondents saying their spouses would comment: “You seem so happy, it’s great!” and 21.6%, “I wish I had a job like that.”

Internal Pressures

The developers themselves, when asked to name one thing they would change if they could, said they would earn more money (24.8%) and work shorter hours (23.7%). Other popular answers included increased job stability and more interesting projects. Only 3.7% of respondents said they wouldn’t change anything.

Finally, when asked to identify the principal source of stress for themselves and their coworkers, the respondents chose tight ship dates (36.8%), far ahead of bad relationships between management and developers (15.8%) and uncertainty regarding the next project (11.5%). Only 3.5% of respondents said “Everything is fine.”

Long Hours

Game developers spend a lot of time at work, sometimes by choice, sometimes because it’s the only way to ship the game on time and avoid a disaster, sometimes because it’s company policy. **Most of the time, it is due to outside pressures:** Only 11% of respondents said their companies would release a game only when it was ready, versus 47% who are under significant pressure to release at a certain date (usually Christmas) and another 38% who just can’t afford not to.

- **Almost three developers out of five report working 46 hours or more in a typical week** (38.1% say 46-55 hours, 19.7% say over 55). Most of them (58.8%) say their colleagues work about the same number of hours as they do.
- **Crunch time is omnipresent**, whether before every milestone (57.2%), during beta (20.7%) or on at least a monthly basis (16.7%). Only 2.4% of respondents report that their company never has any crunch at all.
- Crunches of all durations were reported, with the most frequent being 1-2 weeks (29%) and 2-4 weeks (23%). Over 18% of respondents reported having experienced crunches of two months or more.

- **During crunch, respondents work 65 to 80 hours a week (35.2%)**, with 55-65 hours also being frequent (30.4%). The average crunch work week exceeds 80 hours in 13% of responses.
- When asked to describe their company's policy regarding crunch, **a whopping 51.7% of respondents said "Management sees crunch as a normal part of doing business in the game industry,"** ahead of "Management sees crunch as a necessary evil and tries to minimize its impact" (38.9%). Only 2.3% of respondents said their companies actively implemented no-crunch policies, like the 40-hour work week.
- **Overtime is often uncompensated (46.8%)**, with the most common form of compensation being time off at the end of the project (19.4% give partial compensation, 3.2% count all hours), ahead of royalties and profit sharing (12.5%). Only 4.3% of respondents say their companies pay overtime in cash.

Inadequate Staffing

Game companies often don't have the human resources to support a smooth production process.

When we asked respondents to describe their company's staffing situation, the most common response was "We could use some more people or special skills in some areas once in a while," with 44% of the vote. This is the case just about everywhere in the private sector these days, and thus isn't a cause for concern.

However, fewer than 10% of respondents said their companies "have all the people that we need to make production smooth and painless." Compare this to the 31% who said they often had to work extra hours and/or improvise because they couldn't hire all of the skills they needed, not to mention the 12% who said they were chronically understaffed and production was always stressful.

Work Organization Problems

Work organization and project management are major problem areas according to our respondents.

- Only 13.5% of respondents said that their companies' pre-production schedules and staffing plans were "very accurate" or "sufficiently accurate and flexible to get by with only a minimal amount of crunch time". The most popular answers to this question were "Reasonable in most cases, but occasionally flawed, leading to tense periods" with 38.9% and "Wishful thinking that will only fit reality if no unforeseen problems arise" with 32.4%.
- **Almost as many people (11.7%) said their companies' schedule estimates were so optimistic they knew they'd be in crunch from Day 1 as said they were accurate (11.9%).**

- Feature creep wreaks havoc with schedules in 32% of companies, while another 49% routinely add features during production while attempting to minimize their impact. **Only 16% of respondents said their companies had formal change control policies.**

On the other hand, developers as a whole are happy with their working environment, which they characterize as “comfortable” (54%) and “effectively promoting teamwork” (35%). Issues raised include lack of privacy (34%) and noise (24%), which are often a consequence of open floor plans, and computers/networks that need upgrades (24%).

Games and Community

While few developers (14.5% overall, but 32.9% of the small female sample) say they are personally bothered by controversial content in games like Grand Theft Auto 3 and Postal, many (43.7%) resent the media coverage that makes game developers look bad.

Another 15.8% of developers said: “The content doesn't bother me personally, but it bothers my friends, family or community and that makes me uncomfortable.” This may represent an additional source of outside pressure for them.

Credits

Most respondents say they always get the in-game credits they deserve (53.7%) and/or that the credit allocation policy at their company is fair and balanced (22.8%).

The problems raised by respondents include management and publisher staff getting too much credit (33.8%), people losing their credits if they leave the company before the game is published (24%) and people getting credits for games on which they didn't work (21.9%).

THE BALANCING ACT: A GAME CAREER VS. 'A LIFE'

From the Interviews...

"I choose to spend a lot of time at work, and that can mess with personal life. A lot of times I'll come home from work and play a game. It can be hard for a girlfriend to deal with (my last 3 girlfriends have blamed the game industry as the reason our relationship didn't work out)." – Programmer, Canada.

"Most companies seem willing to work with someone who is ill or has other concerns, though. In other smaller companies, there was a great deal of flexibility. It seemed to be directly related to the individual's standing within the company." – Programmer, Orlando.

"When I left my company, they were aware that the testers just get tortured. It's a really hard, lousy job, you don't make any money, and to the rest of the company you are like bugs. [...] I lost my girlfriend of 7 years because I worked a Christmas and that was the straw that broke the camel's back." – Tester, USA.

"I feel very comfortable with what I do, because I love it. It was my childhood dream. [Before a milestone] I've worked 10, 12 hours everyday, very stressful indeed. But we have our good times also, like pizza at night and a lot of jokes!" – Programmer, Brazil.

From the Survey...

- 61.5% of respondents said their spouses would be likely to say they work too much and don't spend enough time with their families.
- 43.5% of respondents said their spouses complain that they are always stressed out.
- 43.7% of respondents resent the way the media cover controversial games like *Grand Theft Auto 3*.

Preamble

When looking for a new job, some people won't even consider going outside the game industry, because making games is their dream. Others, however, may notice that non-game companies like HP or Qualcomm offer on-site day care, generous health benefits, family-friendly policies and family get-togethers. This sort of side benefit is most likely to be attractive to senior staff, who are older and have formed their own families.

The computer game development industry has become notorious for overworking and 'burning-out' workers. Accordingly, workers tend to leave the industry for less stressful work in early to mid-career – sometimes by age 30. The worst consequence of this "brain drain" is that few seasoned veterans remain in the industry to develop technology and techniques utilizing their wealth of experience, and so we as an industry repeat the mistakes from which we haven't been able to learn.

In this section, we look into several of the causes of burnout and discuss how a member of the game community can balance a game career with “a life.”

Why Achieving Balance Is Hard

Long Hours

Obviously, it is difficult to balance life and career when the career demands a disproportionate share of the individual’s waking hours. In the game industry, long hours are a norm, and extremely long hours, far from uncommon.

Given the importance of this phenomenon, we will devote an entire section of this paper to the specific causes and effects of long hours.

High Risk, Short Shelf-Life Product

For game developers, never has the pressure to work hard and fast been stronger than it is today:

- Game budgets skyrocket, but fewer than 5% of development projects actually break even once they reach the marketplace.
- For many teams, missing the Christmas sales season means immediate bankruptcy – the product must ship, no matter what the costs.
- If the game fails to sell at a healthy pace during the 4-6 weeks following its release, retailers will quickly pull it from the shelves and replace it with something new. For a developer who has spent years working long hours and investing a great deal of himself or herself in a game, seeing it vanish from the market with barely a whimper is a depressing experience to say the least.

A movie has many opportunities to make money, including theatrical release, pay-per-view, DVD purchases and rental, premium cable (e.g. HBO), regular cable (e.g. Sci-Fi) and broadcast television, merchandising, novelizations, the Director’s Cut, etc. At every step, broad print and electronic media advertising support can be expected. Perhaps more importantly, the public’s desire for information and gossip about what’s going on in Hollywood provides opportunities for actors, directors, and (sometimes) screenwriters to appear on talk shows and in mass-market magazines to promote their latest project.

For a game, purchase opportunities are much more modest. A best-seller may get re-released on a “Classic” or budget line some time after it has gone out of print. In a very few cases, the publisher may sell ancillary rights to make action figures, a movie or a book based on the game franchise. None of this helps the run-of-the-mill project. And while a few game releases will be supported by TV ads, most will have to rely on reviews in the gaming press, who typically give more attention to the big-name publishers, and on impulse buys (which still account for a majority of game purchases). In any case, a player who wants to buy a specific game had better do it as fast as he

can, because it may very well have disappeared from the shelves the next time the player returns to the store where he first saw it.

Poor Processes

Making games is a risky enterprise. Many projects never make it to release. Reaching the shelves is no guarantee of quality. Quality is no guarantee of market acceptance. But why is game development so risky?

Any business venture, from making fitness equipment to selling ice cream, must answer these questions:

- Can the product be made?
- Will people buy it?
- Where and how will they buy it?

What is clear from the evidence is that we, as an industry, are fundamentally no good at answering these questions with any level of accuracy, because we have insufficient knowledge of what constitutes good practice, method and process. As a result, it is often impossible to predict the outcome of a project, to funnel development so that poor projects are eliminated early, and to measure the performance of individuals and teams (including developers, publishers, marketing etc).

Case in point: A large international food manufacturer has a 30 page booklet which it gives to all newly hired graduates. The book contains about fifty process diagrams, hierarchies, and models, which form the basis of how the company creates new products. In addition to R&D, a new product, such as a new ice-cream dessert, may require a new factory to be built. The costs are substantial. How many games are developed with such strict methods and processes? How many have any? Games were once a cottage industry, but now many games approach or exceed the costs one would expect of a large manufacturing operation. An examination of product development strategies from other industries would prove useful.

Do we believe that we are doing the best job that we can, and that games are just inherently high risk? Shigeru Miyamoto thinks not.

“I’m sure that each case has its own unique cause, but I know that often times when game designers and producers make their plans without a sufficient grasp of the technology and engineering necessary to make their game, they will often fail.”

[Miyamoto99]

Overly Ambitious Scheduling and Budgeting

Small development houses tend to live on cash-flow – which is an eloquent way of stating that many, if not most, live from one publisher advance to the next, often depending on a tenuous credit line to bridge the gaps and make payroll when an

advance comes in late. As a result, these companies perpetually find themselves at the mercy of the publisher in charge of their next project, because a couple of months without a contract will drive them to bankruptcy.

As a result, project planning often degenerates into something like finding a more or less appropriate answer to this question:

How little can we, the developer, ask in our proposal and how soon can we promise the product without the publisher thinking we're totally nuts?

In asking this hypothetical question the developer often knows full well that they won't be able to deliver on the schedule and for the price they quoted. Moreover, they bet that the publisher will give them more time and money later to finish the product instead of canceling the project outright – despite the fact that publishers of late have gotten wise and built severe penalties for such breaches into their development contracts.

Worse yet is a developer who takes the following hypothetical (but hardly unusual) position:

Hey, we know a lot about games, but we aren't big on planning. So let's 'guesstimate' how long and how much it will take. We'll keep it low enough to make sure we get the deal. Somehow it will work out.

This is a team member's nightmare come true. When it comes down to it the delivery, if it happens at all, will come from a combination of the following:

- The team is burdened with unconscionable amounts of overtime in order to meet badly conceived milestones.
- The work is secretly subcontracted outside the development house in hopes the publisher won't find out.
- The project will have to be renegotiated in midstream to extend deadlines (with penalties) or to scale back the scope of the game to fit with reality (likely with heavier penalties).

No matter what the eventual outcome of the deal, the team members will suffer because of the inability of management to either plan or negotiate reality.

Competitive Pressures

Many studios suffer from competition, rivalry and divisiveness between professional categories, be it Artists vs. Programmers (a problem so endemic it is the topic of its own roundtable at the annual Game Developers Conference), or Producers vs. Everyone. Competition also exists between different companies, and sometimes between teams at

the same company. While competition has many positive benefits, we will limit the discussion here to the negative aspects relevant to this white paper.

- **Personality conflicts** within a team can negatively impact day-to-day life at work. Much work has been done to study different types of personalities and their interactions; for example, other IT industries use psychometric tests and lengthy interview processes to ensure compatibility, but this is rare in the games industry. There can also be a measurable difference between peoples' personality types when things are going well vs. when things are going badly. Games companies appear to hire a bunch of people and see who hangs out together. When good teams form it is seemingly by chance, not through any understanding of team psychology.
- **Failure modes.** Each vocation contributes to the project in different ways, and therefore when each fails, they fail in different ways. How they fail, when they fail, when the failure is discovered, and how the individuals responsible cope with failure and how others respond are all different. Little has been done to study the failure modes of game development teams (and how to recover from them.)
- **Responsibility dynamics.** Someone is always responsible, and therefore may have to make difficult decisions that are not understood by those who are not in the line of fire. The youth and inexperience of teams can exacerbate this problem.
- **Lack of understanding of each other's roles.** We asked developers to point out the specific failings of the people they thought were "bad" examples of other professions. A number of programmers and designers said that "bad" producers lack practical knowledge of how to run a project and harbor an unfounded conviction that they know how to design a game. Producers thought "bad" programmers were overambitious and unable to comprehend the importance of deadlines.
- **Unrealistic expectations.** "I don't care if we have 10% of the budget and manpower, I will make something better than the next *Half-Life*. Our engine must do XYZ. Our artwork will win this and that award." A single individual can delay a project by holding their work to unrealistic standards – typically by comparing it to the equivalent in a high-profile game that benefits from enormous resources. An honest appraisal of the game-play and financial objectives of their own game could give developers much greater satisfaction.

Controversial Game Content and Community Resistance

Many high-profile games feature interaction based on very high levels of violence, sometimes putting the player in the role of a drug dealer (*Grand Theft Auto*), an automotive mass murderer (*Carmaggedon*) or a psychotic disgruntled employee (*Postal*).

Whether this violence is justified by storyline purposes or not, the presence of the interactivity factor – the player makes a conscious decision to shoot at people – has made it harder for many people to accept the violence in games, even though it may not necessarily be higher than in relatively mainstream movies (*The Matrix*, *Terminator*) or even in classical literature (*The Illiad*).

Given the glut of product on the market and the low probability of a hit, some developers and publishers have tried to differentiate their products by pushing the envelope in the area of controversial content. Some of these attempts have been successful (*GTA*), others not (*BMX XXX*), but the publicity even the failures have attracted may be sufficient to sustain the trend.

Indeed, the very things that make games fun for some categories of players also make them questionable content for younger players – some of whom are the very children of game developers. This prompts the question: “When a game that an individual works on isn’t ESRB-rated “E” for everyone or “T” for teen, is it appropriate to have your own children play it – and to encourage other peoples children to play it as well?”

Consequences

Injury to the Family Unit

Overtime and stress have numerous consequences for the family unit, the most obvious of which is the inability of a husband or wife, father or mother, to be present when needed.

Of course, occasional absences can usually be dealt with, but when overtime is habitual and an individual is leaving home at 6 AM and returning in the wee hours of the morning, the deleterious impacts are unavoidable. One producer, who prefers to remain anonymous, described his schedule as follows:

- In the office at 6:30 AM
- Home at 1:00 PM for a nap
- Back at the studio at 3:30 PM
- Home at 1 AM to sleep until 6AM
- Repeat, six days a week

This makes for 14-15 hour work days, presuming a short commute, and 84-90 hour work weeks. Clearly this is an extreme case. But 10-hour days and 50-hour weeks are common in the industry – and woe to the “slacker” who finishes his or her work in 35 hours and goes home in a company where long hours are the norm.

No industry-specific divorce or burnout statistics exist. Also unidentified is the impact upon non-marriage relationships, boyfriend/girlfriend or same-sex couples, or the number of people who are unable to begin relationships because they are unavailable to meet people outside the workplace and unwilling to engage in (usually inadvisable) workplace romance. But it is easy to see that unpredictability of work schedules leads to

tension in relationships, marriages and child-parent interaction. Frankly, game development “widows” (and widowers) may well outnumber those estranged by televised sports...

Early Departure from Gaming to Related Industries

In many ways, the video game industry has become a romanticized career for young and aspiring programmers, artists, writers, and producers. This phenomenon is not unlike the draw to the motion picture industry, which has high public exposure and often high rates of pay – although the general public’s assumption that game careers provide high salaries, royalties and stock options are not always founded in truth.

But once the “honeymoon” is over, nearly every game developer begins to feel the stress of deadlines closing in, code that doesn’t quite execute the way it was supposed to, a publisher who seems increasingly dissatisfied with progress or (perhaps worse) demands huge amounts of additional (and unplanned) content or functionality in the product. At times like this, even the most dedicated developer can’t help but thinking:

Wouldn't a nice 9-5 job at a blue chip company with a pension plan and job security be great?

With too much work, little probability of achieving notoriety within the community through involvement in a hit project, and very little respect from non-players over the age of 30, the daily life of a game developer has little in common with this romantic notion. As a result, countless experienced developers leave gaming for less stressful, more socially acceptable and often more lucrative related industries, like movie special effects, corporate software and other media. The results of the IGDA Quality of Life Survey are telling in this respect: **only 3.4% of respondents said their coworkers averaged 10 or more years of game development experience, and only 9.8% of respondents said their leads averaged that much!**

The loss of talent and experience for the industry as a whole can only be described as crippling; time and time again, studios unable to hire sufficient numbers of seasoned professionals are forced to push junior employees into roles for which they aren't ready, with predictable results on schedules, crunches, and profitability.

Community Pressures

Bad publicity associated with game content being criticized in public forums may lead to the creators of interactive entertainment being criticized in their community by sheer virtue of the work that they do. While rarely sufficient to drive people away from the industry on its own, this factor may trigger a decision in an already dissatisfied person.

Conversely, the public spotlight placed upon the game industry can cause another form of stress – that known to those in motion picture and television studios for decades, the quasi-celebrity status which can bring inquires as broad as: “Can you get my kid a job in games” or as precise as: “Can you get me this toy related to the game you worked on?”

Best Practices

Conveying “Realism” to Third-Party Financiers

First, determine what a publisher or financier can and/or will pay for a product and under what terms. If the other party’s expectations are unrealistic, you are better off walking away from the deal and looking elsewhere.

Within that framework, your management and legal counsel must negotiate the best terms possible while keeping competitive. Contract negotiations deserve an entire white paper in their own right (and the IGDA has published two contract walkthroughs), but here are a few tips:

- Have a detailed and supportable budget and schedule – preferably with line items available. This can expose your payroll, but it will also make your case clear – and besides, the publisher or financier will often audit the details before signing anyway.
- Tout the skill of your management and team. Show that you have a track record and are a comparatively low-risk proposition.
- Assure that keeping your team and management happy will avail the financier of a production resource for future products, saving them the need to seek-out other houses.

There is no such thing as “a perfect deal.” If you get “pretty-good” most of the time, you are doing well.

Support of the Family Unit at the Employer Level

Quality of life, for an employee and his or her family, is not merely a matter of time and compensation. Other major influences include security and community.

- ***Health care.*** Access to comprehensive health care can contribute greatly to an employee’s sense of security and well-being, particularly an employee who is supporting a family. In countries with national health care systems, this may not be much of an issue for developers, but in countries without such a system (the United States being the most prominent example), access to employer-provided health care is a significant benefit. Although a good health plan can represent a significant expense for the employer, it will help reduce turnover, especially among older employees who get married and start families.
- ***Family get-togethers.*** By creating opportunities for the families of team members to meet and bond, the company can help create a community and a support network that can help fight loneliness during crunch times, especially for the families of employees who have recently moved from another city or country.

Sammy Studios held a day-time family party during the 2003 Christmas season – which impressed at least one job applicant a great deal!

- **Be tolerant when a child or spouse is ill or during important life events.** For example, the UK has statutory paternity leave, but this is meaningless if the company puts pressure on fathers not to take it.
- **Reasonable working hours.** Much of this document emphasizes the importance of good project management and its effect on quality of life. This is especially true for families. Be supportive of employees who devote time to their family. This is a good thing. Remember that just because one team member spends less time in the office doesn't mean that all the others will demand the same: different people have different priorities. In one example, a new team member joined and worked 10-6, while some of the team worked around the clock and slept on the couch in the corner. His efforts were much appreciated, by the team and the client, and added bonus content to the game. Teams work well when everything is communicated, not necessarily when everything is equal.
- **Day Care.** Larger employers should establish on-site day care. Smaller ones might think about where the nearest one is.

The Employee's Responsibility

Ultimately, responsibility for your quality of life rests with you.

- **Pick the right company.** Not every company has the same values; find one that fits yours. A company of recent graduates working out of their garage isn't likely to be terribly supportive of family issues.
- **Be up-front with your employer.** Don't be afraid to communicate your requirements and responsibilities to your current or prospective employer. For example, "I am a Dad/Mom/Single Parent and I have these responsibilities x, y and z". If you will be going to your kid's baseball games, be up front. You may not get the job – but then again, you would regret it if you did.
- **Know when to be flexible.** If a problem arises, know when to bend – and when to stand your ground. Negotiation is a two-way process; if one party makes all of the concessions, neither is likely to be satisfied with the end result.

LONG HOURS AND EXTENDED CRUNCHES

From the Interviews and the Message Boards...

“Most of [our projects] ended with a 2 or 3 month crunch period, the record being almost 1 year of crunch. Crunch time usually consisted of 1 to 4 hours of daily overtime, with weekends added at the end of the crunch period.” – CTO of a small studio in France

“Oh Geez. The longest crunch was an eight-month period that pretty well comprised the entirety of my involvement on the project. [...] From a quality of life perspective, they are taking advantage of people. Some people took the stance that if you didn't want to work the crazy hours, they would show you the door. The threat of losing a “cool” job was really held over everyone's head [...]” – Programmer, Florida.

“When we were getting ready to ship 11 SKUs (one game, different platforms and languages), we worked 8:30a to 12:00a for 3 weeks, no weekends. Testers work like dogs even without crunch: 12 hours a day, 6 or 7 days a week.” – Programmer, Canada.

“I worked really long hours [...] In the end you're working like 100 hours a week sometimes. Sleeping in the back of your desk or underneath your desk. Get up, take a shower, hit the gym, get back in your seat and start working again.” – Tester, USA.

“First of all, let me say that I have never seen any of the behaviors you talk about (threaten, bully, coerce, etc.) But the truth is, when getting close to a deadline, the team usually needs to kick into second gear. Usually, during these periods a core team does most of the crazy hours. It's always the same people, and they are people that accept this. For example, on my last project, during rush periods, we had about 50% of the programmers staying till 10-11 pm almost every night. About 30% did this two or three times a week. The other 20% did their normal hours, not more, not less. Management knows that not everyone can give 80 hour weeks, as people (especially seniors) are starting to have wives and kids. The trick is to build your team with a good mix of “crazy workers” and normal people. Like every thing in life, balance is the key.” – Maxime, Ubisoft Montreal.

“I've been working in games for almost a year and a half now. When the project reached deadlines, most of us spent an extra 2-4 hours a day, but this didn't last longer than a couple weeks. I have had to do only one all-nighter, and a couple work days ended at 3 am. I consider this getting off easy compared to my boss and the stories he tells of other companies...” – jasonbentley, USA.

From the Survey...

- Three developers out of five work 46 hours or more in a normal week.
- During crunch, 35.2% of respondents work 65 to 80 hours a week and 13% work over 80 hours a week.

- Only 2.4% of respondents say their companies never have crunches.
- 51.7% of respondents say that their management sees crunch as a normal part of doing business in the game industry. Only 2.3% say their companies actively pursue no-crunch policies.
- 11.7% of developers say their companies' schedule estimates are so wildly optimistic that they know they'll be in crunch from Day 1. Only 11.9% think they are "very" or "sufficiently" accurate and flexible to minimize crunch time.
- 34.3% of developers expect to leave the industry within 5 years.
- 51.2% of developers expect to leave the industry within 10 years.

Preamble

Long hours are the de facto standard at many game development companies today. A standard that often goes unchallenged because it has become part of our "hardcore" culture – but one that may cause more harm than good to our companies and our people.

This section of the paper is intended to point out the problems associated with extended overtime and to offer viable alternatives. The fact is that a number of game development companies make high-quality, top-selling games without severe or extended crunches, and are in fact doing it in something resembling a regular 40-hour work week. Long hours are **not** a necessary evil.

How We Ignore the Problem

To judge how developers view long hours, we have examined thirty relatively recent *Game Developer* magazine project Postmortems articles selected at random. The results were unsettling:

- Of those thirty, only five specifically cited arduous hours and overtime as both regrettable and something to be avoided in the future.
- Nine implied or mentioned that there had been serious overtime, but did not describe it as either regrettable or avoidable. One of those nine described people working twenty-four hours straight, sleeping six hours on site, and then working another twenty-four as "...fostering a hardcore work ethic..." Another mentioned working until dawn and sacrificing weekends, but not in a way that implied that it was wrong or even very consequential.
- The remaining sixteen Post Mortems made no mention of overtime at all. However, given the norms of the game industry, it is safe to say that a fair portion of the projects covered in them entailed some significant level of overtime as well - and it says a lot about our industry that the authors didn't think enough of overtime to raise it as an issue.

The fact that a majority (51.7%) of respondents to our survey said their companies' management view crunch time as a normal part of doing business shows a general

lack of concern about the consequences of the situation – not the least of which is the fact that 51.2% of survey participants expect to leave the industry within 10 years!

Causes

The interactive entertainment industry thrives on enthusiasm. The garage-hacker culture, where game development has its roots [Adams03], drives the decisions and attitudes of many people in the industry to this day—creating an environment which places a greater emphasis on hours worked and passion than on effectiveness and long-term productivity. In short, we tend to “work harder” more readily than we “work smarter.”

Though it is always difficult to determine specific causes, or to generalize over a large industry such as gaming, the main causes described by game industry participants and veterans seem to generally boil down to these:

- The game industry has traditionally been staffed primarily by young game enthusiasts with a surplus of enthusiasm and dedication, a deficit of real-world work experience and task management skills, and (usually) few binding commitments such as marriage or children. An inability to accurately estimate tasks and schedules, great enthusiasm for the job at hand, and lack of any real disincentive to work all the time: this is a sure recipe for extended, crazy hours.
- Once long hours and brutal crunch times are locked in as both the cultural norm and a “necessary ingredient” to ensure project completion at a given company, they don’t go away. As we will see later, long hours are not in fact any guarantor of increased productivity, but it is not hard for developers to convince themselves otherwise: when the pressure is on and the stakes are high, it can be comforting to be able to tell yourself that you’re doing everything possible to succeed, even if it is not really helping, and is more likely to be hurting productivity over the long haul.
- When people who learn to make games “the hard way” migrate to other companies or start their own, they bring with them the work practices and development methodologies they have acquired, thus propagating the myth.
- A large proportion of games under development have hard deadlines, such as the E3 pitch and the Christmas release, which can’t be missed without imperiling the project. A hard deadline, combined with an insufficient planning and scheduling process, invariably leads to evenings and weekends in the office.

Consequences

Loss of Experience

Long hours, especially during extended crunch periods, conflict with our ability to experience and enjoy life outside of work, including family life. There is significant anecdotal evidence that turnover within the industry is affected by this conflict (see interviews in the appendices for examples).

Worse: as game makers gain experience and become more valuable to the industry, they are also more likely to leave the industry for new careers that offer extra time for family and other interests. The impact of this loss has not yet been quantified, but if parallels in software production are any indication, it exacts a high price, leaving us with enthusiastic kids and a perception that you can get away with sloppy management if you make up for it with passion and enthusiasm. But studies show that experience is much more effective and much less expensive - we are being penny-wise and pound-foolish.

As the IT industry has matured it has discovered that programmers and software architects generate a greater number of bugs under conditions of high schedule and workload pressure, costing the company more than a balanced schedule would. The loss of highly experienced contributors greatly exacerbates the problem since experienced programmers introduce fewer bugs at every stage, produce more output per dollar of compensation, and choose lifecycle and design paths that result in a substantially less expensive project given any particular feature set. The difference between a team of people with an average experience on 2 years with 10% project turnover vs. an average of 10 years and 2% turnover is estimated to be 50% on the overall cost of the project - and the experienced team is 75% more likely to be able to stick to the original timeline.

Game developers themselves often cite working conditions, extended overtime as the critical factor among them, when explaining why they quit companies or leave the industry entirely. Is it any wonder?

The bottom line: Our companies are losing some of their very best people to a practice that is yielding less than optimal results. Recruiting costs continue to climb as studios swap experienced craftspeople back and forth. Recruiting talent is expensive. Training and acculturating new people is expensive. Ramping up new team members mid-project is expensive. Bad design and management decisions that would have been avoided by more experienced contributors are expensive.

Loss of Efficiency

In companies where extended overtime and crunches are the norm, people get burnt out. Once they are burnt out, whether it happens in mid-development or in critical milestone delivery stages, they can't possibly be giving their best.

Game-making is one of the most difficult technology-related creative endeavors in the world. Asking people to perform at their highest levels when they haven't had enough sleep, haven't had a weekend to themselves in weeks/months and are just plain burnt out on the project is clearly nonsensical. Even worse: team leaders and managers are also probably burnt out; their own judgment is compromised, and they are not able to give either the project or the employees their best.

Smart game developers do their best to prevent extended overtime. Fatigued engineers, for example, should not be allowed (let alone encouraged) to check in code at the end of a brutally long work day or series of work days: the probability is very high that this code will contain defects, far more of them than would be the case if the engineer were allowed (forced) to go home, get some sleep, and then have the opportunity to review his or her work the following day.

“You may honestly believe that mandated overtime is helping your staff get the work done. More likely, however, you are actually encountering slow progress, as your programmers are creating more defects and much of the work that was done late at night fails to stand up to the critical light of day.” – [Rothman00]

The fact is that overtime is most often the consequence of less than ideal planning, task estimation and scheduling. While there certainly are times when some level of overtime can be utilized and prove useful, anything beyond a week or two is in itself an indication that things have gone awry.

“When a project is perceived to be out of control, requiring developers to work more overtime is one of the most common things managers and team leads do to bring the project under control. But overtime is, in itself, a sign that a project is out of control.” - [McConnell96]

Best Practices

Diminished (Non-Existent) Overtime a Priority

First, in order to achieve reasonable hours for employees, the commitment to doing so must be one of the company's essential values. Otherwise, and especially if it hasn't been an essential value to date and one is trying to overcome an ingrained overtime culture, it will be too easy to backslide when confronted with a difficult situation.

Focus on Task Estimation and Scheduling

Developers need to be very good – great even – in the pre-planning and planning phase of their projects. This is perhaps the most critical time in a game's development, and where most games go awry. If teams are not painstakingly thorough here, they will not be successful at keeping to the schedule and thus keeping the team's hours down. (Further reading on effective planning techniques is highly recommended – see the paper's bibliography for references.)

The benefits of effective planning should be obvious:

- From a quality of life standpoint, people spend less time at work, and can devote time to family, significant others, friends and outside interests.
- With appropriate systems and infrastructure in place (an important caveat), having employees work relatively steady 40-hour weeks is demonstrably more efficient and productive, especially over an extended period of development time, such as that presented by a game development cycle. In the long run, a 40-hour work week, if implemented in conjunction with professional project development methodologies, will yield better, less buggy products delivered on time and within budget.

Understand the Scope of Your Game

Developers must have a clear grasp of the essential components of their game, and most notably the various components that critically affect the scope of the game. Developers must know what can be cut, if necessary, and what can't.

Schedule Ownership

An eminently capable person (probably the producer) must “own” the schedule. This person has to be a top-notch planner, or be supported so effectively that any shortcomings are vanquished.

Ownership here means accountability and responsibility, not dictatorship. This accountability must filter down to the leads and the team. The leads should be thoroughly in agreement with the schedule and committed to it. Involving the leads can only have positive results. It reduces or eliminates the divide between producer and team, it contributes to their growth, and it focuses them on the business realities.

Schedule Methodology

There are a few identifiable types of game projects, which can require markedly different approaches to project planning and scheduling. Here are two examples. Your project may not fit exactly into one of these admittedly general categories, but is probably sufficiently similar to one of them that you can draw some useful parallels from it.

Category 1: Well-established Genre and/or Design

A lot of games in development have designs that are fairly well understood from the beginning of the project, particularly games that are sequels, or are examples of well-established genres. These projects generally involve the production of a lot of content (levels, characters, speech, text, music, etc), and require a lot of tuning, but the scope and nature of the work is well understood from the outset. Such projects can greatly benefit from rigorous, detailed scheduling at the beginning of the development cycle. This is possible because the game design is not likely to change greatly during development, so detailed plans are likely to remain useful and relevant.

For example, in a port of a game from one platform to another all the game design, artwork, and programming has been done once already. There may still be large areas of risk; one of this paper's authors led the effort to fit two full CDs of *Resident Evil 2* onto a Nintendo 64 game cartridge. However these can be targeted at the beginning of the project. Software practices created for classic software application design are therefore applicable to these kinds of projects.

Category 2: New Genre, design, or technology

Another category of game project, requiring a different approach to planning, is one in which the design or other major component of development is not well understood before development commences. To acknowledge this lack of understanding does not necessarily constitute bad planning. To crudely paraphrase the military, "No new game design survives contact with actual players." Some examples of such projects might include:

- developing a brand new design or genre
- creating a hybrid that synthesizes pre-existing genres in a new way
- incorporating significant new technology (for instance a new engine, or console, or peripheral device)

In this development model, beginning with detailed design specifications is a waste of time, as they will quickly be rendered obsolete during the course of development. However, rigorous planning (and tracking progress according to the plan) is still essential to the project's success; it just takes a different form. Generally, this sort of project entails early development of a full-featured prototype, followed by rigorous testing and iterative refinement of the prototype, followed by production of all the shipping content and tuning of the final game, after the design and/or technology is well understood. In this case, you are planning and scheduling the process whereby you arrive at the final design, after which traditional detailed project scheduling for asset production and tuning can take over to good effect.

These are not the only approaches available, nor are they necessarily mutually exclusive. The crucial points to take away are:

- Understand what sort of project you are attempting, and make sure your planning/scheduling approach fits the project
- Once you have settled on a plan, execute it relentlessly and in great detail. You cannot be too prepared.

More detailed discussion of this topic is outside the scope of this paper, but there is plenty of excellent material available on this subject. Fortunately there is a flourishing and healthy discussion on the subject of game design (e.g. [Kreimeier 03]) and the reader is directed to the many useful references in this paper's bibliography.

Schedule Realistically

Much of the responsibility for scheduling resides with developers. Keeping in mind the “market” mentioned above created by undershooting schedules, it behooves the management of a development house to realistically project the time it will take for the team to do the job. A well presented and well thought-out schedule that assesses the team, its skill set, its skill level, and a supportable man-hour labor projection with conservative contingencies will prove a compelling argument.

On the publishing side, executives and the internal producer handling outside development houses have an obligation to project in advance the amount of funding and time a given product will need. It is irresponsible for a publisher, who is in most cases the larger, better funded and more experienced party, to take the position “we’ll get the lowest bid and then put heavy penalties for missing milestones.” A responsible publisher will, instead, provide sufficient information, in some cases a worksheet, for the developer to estimate cost and time. The publisher will then qualify that each bidding developer, presuming there are multiple bids, is both competent and realistic in its approach to the project.

One developer shared an historical account of a bid he submitted for US\$3.5M for a project. The publisher insisted they only had US\$1M to spend. The developer said it wasn’t possible and if they wanted a \$1M product they would have to redesign it bottom-up to meet the number. The project was awarded elsewhere, ultimately ran a reported, but unconfirmed, \$3.6M, came in a year late and flopped in the market. History tends to repeat itself.

Some Tips

When building the schedule, account for absolutely everything: attending shows and conferences, creating demos, vacations, projected sick time, etc. Then, add a “fudge factor”.

Collaboration is vital here, as it is throughout the game-making process. When individuals are starting to hit a wall while thinking through and estimating tasks, for example, talking it through with one or more of their colleagues can prove invaluable. First, it can break the tedium, but secondly (and even more importantly), it can help them to see the problem in a new way. To be efficient, companies and teams must have clear, regular and vibrant communication, unhampered by politics or personal agendas.

There should be no such thing as a five day task, and certainly no task longer than three days. (This excludes block items such as “Optimization” and “Bug Fixing”, which can even be considerably larger.) When an engineer, for example, labels a discrete task “5 days”, it should be a clear indication that he or she hasn’t thought enough about the feature or problem.

Also keep in mind how difficult accurate task estimation really is. There is no substitute for experience, here. All the more reason to get your overtime hours down, and hang onto those experienced engineers who know how to accurately estimate projects!

The schedule's owner must continually monitor progress, periodically reworking and revising the schedule to reflect circumstances, or just as a sanity check. It is not out of the question to rework the schedule several times over the course of the project – as long as there is rock-solid justification for it. Constantly test original assumptions.

Realistic Budgeting

As with scheduling, budgeting has to be realistic. Employees of any organization expect to be paid a competitive wage given their qualifications and geography, they want to receive normal benefits, and they appreciate, if not outright expect, part of the profits in one of the common mechanisms for distributing them (options, royalties, ESOP plans, profit sharing etc..).

In order to maintain a happy, creative workforce, the budget must meet the needs of the staff and those of the organization as a projected on-going entity. Ideally, this budget should be part of a greater business plan by the management team, but at the very least it must cover (comfortably) the needs of the project for which a bid is being prepared.

Likewise, publishers should realize that “squeezing” a developer for every last penny available may come back to haunt them: In a best case scenario the developer will be mildly resentful for not having been paid a fair price. In the worst case the publisher could end up having the developer go broke mid-way through the project and end up wrestling with a bankruptcy trustee to recover salvageable assets – before finding another development house to finish the product (usually at a premium).

Implement a Strict Change-Control Policy

The amount of resources and time available to complete a project is finite. To eradicate extended crunches, developers must live and breathe the notion that features and tasks cannot be added without a corresponding amount of features and tasks being removed, unless, of course, sufficient resources and/or time can be added to the project.

On a related note, there are invariably times where developers learn new things, or come to a different understanding about their game. When that happens, they'll want to have the ability to apply that new learning or understanding. “Windows of opportunity” during which such changes may be performed should be built into the schedule, but since by definition the impact of unknown discoveries is impossible to guess accurately at the outset, knowing how and where the scope of the game can be cut or crimped without fatally affecting the project gives developers a significant level of flexibility – flexibility that can be applied to make these adjustments, instead of just saying, “Yeah, we need to add that, too.”

Use Overtime Sparingly but Effectively

Having railed against overtime to this point, we must stress that, when used sparingly, overtime is actually a very useful tool.

“Too much overtime and schedule pressure can damage a development schedule, but a little overtime can increase the amount of work accomplished each week and improve motivation. An extra four to eight hours a week increases output by 10 to 20 percent or more.” - McConnell

It's fairly widely recognized that knowledge workers can absorb a temporary increase in their workload, for short periods of time. The recommendation is to constrain these short periods of overtime to discrete, non-contiguous weeks. Publicize these short crunches as far in advance as possible, so people/families can plan. The beauty of crunching in this manner is that the whole team can buy into discrete pushes: it demonstrates professionalism, confidence and proper planning, as opposed to a schedule veering out of control. As a result, they are able to be even more productive because they are well-rested.

Effective Human Resource Management

Management and managers, have a bad reputation in game companies. However, managers – the people actually responsible for the direction and performance of the staff – are one of the most, if not *the* most critical factor in any business' success. Unfortunately, while most game companies hire people to manage projects, the management of the people involved in the project is often neglected. It is easy to understand why:

- Most game shops were formed around a cohesive goal – nine times out of ten a game idea – but on a loose basis, not on business principles. That seminal idea brings people together and can hold them together – but only to a point.
- Assignment of management responsibilities is also a sensitive issue. After all, someone is being selected to have authority over others.

Why is good management critical? Because even good employees are human; they can falter. They can experience problems outside of work that can dramatically affect their work. They can lose motivation, become stressed out, lose their focus, become unsure of their own direction and that of the game – any number of issues. Good managers know how to deal with people issues effectively and help their staff through tough times. They get the most out of their people, and make each of them better.

It is generally accepted throughout industry that a very good manager can improve performance in an individual by at least 10%, and often times significantly more. Conversely, a bad one costs you at least that, and almost definitely more. A 10% swing may not seem like much, but over the course of an 18-month project involving 6 people, the difference between a good manager (yielding a 10% bonus in performance) and a bad one (costing 10%) will be just shy of *two person-years*. That figure is more than significant – it can make or break a game, and a company.

Picking the Right Managers

It bears repeating that just because a person is a great programmer or artist, it doesn't mean that he or she will be a great manager. In fact, the skill sets required by these positions have little in common.

Promoting a star developer to management when he or she doesn't have the right personality and experience to handle the job in one of the worst possible decisions a company can make: it takes a tremendous asset and turns it into a liability, thus negatively impacting all of the other people who now report to that person. Yes, egos are usually involved in manager selection. It's up to whoever is really in charge to make the right move in spite of personal ambitions, even when it might cause frictions among partners and founders – which is why the person making those decisions must be a great manager as well.

Managers and Overtime

Lastly, and perhaps most importantly when it comes to eliminating extended overtime, management is also about supervision, and effective management is essential to achieving the goal of regular hours. Management is required to make sure that people are pulling their weight, and getting their work done. The only way that regular work weeks can be effective is if everyone is contributing at their highest level during the work day: no slacking off for days or weeks and then working furiously to catch up.

This quid pro quo must be communicated, clearly and from the outset: people can have their weekends and regular workdays, but that means they've all got to be working, and working pretty diligently, during the time that they are supposed to be working. Everyone should be contributing eight hours of work per day, and/or forty hours of work per week. Otherwise, the system breaks down.

Does that mean developers can't play games, or take breaks, or have the fun that game companies are famous for? Of course not. Everyday interactions, lunchtime conversations regarding games, films, music, and just the general shared camaraderie are often some of the biggest benefits of working in gaming. No one's proposing to put a halt to multi-player gaming at night, for example. But that is not work, and that must be made clear. Game companies are also businesses, and any good business ensures that their people pull their weight. That's a big part (though not all) of what managers do.

Obligatory Rest & Relaxation – Maintaining Mental and Physical Health

Despite the best efforts of those planning any specific project, odds are good that there will be some long hours incurred at the end of the development cycle. Care should be taken to plan accordingly, so that development team members (and their families) get a chance to recuperate after crunch. Here are some ideas and guidelines.

Often, there is a period of limbo at the end of a project, when the gold master (GM)

candidate has been submitted to the publisher but has not yet been approved for manufacture. If the project contract provides for funding through gold master acceptance, and not just through submission, this period of time can serve to let most of the team rest offsite without impacting profitability. This is obviously a better scenario than being forced to burn one or more precious weeks from the beginning of the next development cycle. (The few members who need to remain on call to fix last-minute bugs can be liberated after acceptance.)

Awarding extra time off (or “comp time”) at the end of a project to a stressed, overworked team can reap big dividends in team morale and satisfaction. Generally speaking, it is not a good idea to attempt to compensate teams day-for-day or hour-for-hour, as this approach can contribute to unrealistic expectations on the part of the team. Usually, giving everybody roughly an extra week off per year spent on a project is a good rule of thumb.

For comp time to effectively contribute to team morale, it must be clearly explained, in advance. Teams are much more receptive to working their way through tough times if they can see a light at the end of the tunnel. A good time to lay out the comp time policy is at the onset of the shipping crunch. Discussing it earlier can be counterproductive, as the team isn't yet focused on what they are up against, and waiting too long can squander a lot of the benefit, as teams are less likely to feel good about comp time if they are already stressed and bitter from having pulled a lot of overtime

Project management should endeavour to keep tabs on each team member's yearly vacation schedule. Often, with a bit of encouragement, team members (particularly those with families) are able to schedule their yearly vacations well in advance, such that project impact can be minimized. However, it is important for management to communicate to the team that vacations should be taken, and for management to facilitate team members' vacation plans as much as possible. If no one is paying attention, it is easy to get into a situation whereby some team members (often valuable senior people) go from crisis to crisis and never get a chance to take time off, because it is not planned into the yearly schedule.

A more insidious situation can happen when very dedicated employees neglect to take time off, working long hours enthusiastically for lengthy periods – until their bodies crash into sickness or a fed-up spouse precipitates a crisis at the worst possible moment. Enlightened managers may have to make some employees (particularly younger ones) take care of themselves whether the employees feel the need to do so or not!

HIGH TURNOVER AND JOB INSTABILITY

From the Interviews...

“Yes, it happens [that employees leave in the middle of a project to get a better job.] But nowadays, everybody sticks to his position because of the lack of interesting job positions [in the French development industry].” – CTO, France.

“I feel very secure in my job. No problem whatsoever. If I felt insecure I’d leave and go somewhere else. [...] Right now we lost 2 people from tech group and now we need 2 people on top of the 5-6 positions that are already open. It’s a hiring blitz.” – Programmer, Canada.

From the Survey...

- The average level of career satisfaction reported by respondents who have future projects lined up before the current one is over is higher (7.0 out of 10) or who stay on salary during down time (6.6) is higher than for those who live by-project hiring/layoff cycles (5.6).

Preamble

The ever-present threat of project cancellations and layoffs, combined in some cases with no-competition clauses in employment contracts that prevent developers from leaving a bad job, creates an imbalance of power that can contribute heavily towards stress and loss of motivation.

From a studio's perspective, poor quality of life often leads to higher turnover, and with the game industry's ad hoc way of doing things (e.g., no standard job descriptions), high turnover means high costs and delays whenever a new employee needs to be integrated into the team.

Causes

Industry Economics

It is a well-known fact that a very small proportion of games published become successful in the marketplace. In 1999, fewer than 3% of PC games available on the market, and about 12% of console titles, sold more than 100,000 copies – a figure that is itself often far below the breakeven point [Laramée00].

Given the very large number of games published every year and the very limited shelf space allotted to interactive entertainment by retailers, games that don't become runaway hits disappear from store shelves in a matter of weeks. For developers, it is very difficult to feel satisfied when the project into which they have poured their hearts and souls for years finally gets released and almost immediately vanishes into anonymity.

Contract Practices

Employment contracts between studios and their employees are also far from being above reproach. No-competition clauses that prevent employees from leaving a job to work for another game company, as in the well-publicized Ubi Soft / Electronic Arts Montreal incident [Grammond03] during which Ubi Soft sued to enforce a one-year ban on its employees working anywhere else in North America, bring negative consequences for individuals, studios and for the industry as a whole [Laramée03]:

- They provide a powerful disincentive for strong performers to join the company imposing such restrictions in the first place.
- They incite unhappy employees to stay in unsuitable jobs much longer than they should, with predictable results on morale and performance.
- They force experienced developers out of the industry for extended periods – sometimes permanently, if they find life equally (or more) comfortable elsewhere.

While protecting one's intellectual property is an understandable goal, doing so through a no-compete clause is at best counterproductive – and when imposed on rank-and-file employees (i.e., anyone who has not just sold their company and accepted an employment contract as part of the sale), it is also illegitimate. And while no-compete clauses are often unenforceable in many jurisdictions, fighting them may require a court battle, which can be both daunting and costly.

No-compete agreements, enforceable or not, place both a former employee and their after-the-fact employer in an awkward position. This is because the former employer the no-compete agreement serves may choose to enforce it against the ex-employee as well as their new employer – who may or may not be aware of the agreement. The ability to enjoin an employee and employer as well as the ability to seek damages varies according to jurisdiction and circumstance. Moreover, it is possible that an overzealous employer who demanded a no-compete clause could end up in “reverse” legal trouble for trying to encumber a former employee’s livelihood. Again the laws of the venue determine the situation. As a rule it may be advisable to consider a contract that complies with the respective location of employment and is not overly restrictive toward the employee.

Hiring-Layoff Cycles

If project failure and bad jobs hurt quality of life, the threat of a layoff can be a much more powerful damaging force.

A 1996 survey of 1,014 companies, held by consultants Murray Axmith & Associates [Lowe00], reveals that while a layoff does succeed in lowering costs and increasing short-term earnings, morale, loyalty and job satisfaction declined steeply among those who remained. Long-term effects are hardly more inspiring: fewer than half of the respondents experienced a gain in overall productivity, and fewer than a third, an

improvement in customer service. For game studios, who frequently have only one customer (i.e., a publisher) at a time, this is a recipe for disaster.

Exhaustion

Overwork leads to ineffectiveness, errant behavior, and conflicts [Docherty02]. A tired company turns inward and concentrates on defenses meant to contain collective anxiety – in other words, rituals replace flexibility, and relationships between the team and everyone else (publisher, management, quality assurance, the rest of the company) deteriorate into open warfare.

Relying on constant peak performance, and especially on long hours, is absolutely counterproductive.

Lack of Standard Job Descriptions

In a Hollywood movie production [Patz97], everyone knows that the "best boy" is the second in command of the electrical crew and that he reports to the "gaffer," and everyone has a pretty good idea of what the first and second assistant camera persons are supposed to do all day.

Meanwhile, no two game studios give the exact same responsibilities to their producers, there is no strict border between the job descriptions of a junior, senior and lead programmer, and besides the (partial) exception of the design treatment, neither the game design documentation nor the production process are standardized to any significant extent. Publishers can also vary in the type and description of their employees who work and interface with developers.

While this ambiguity may be a good thing from a job quality and variety perspective, it can make it harder for a new employee to be sourced and subsequently integrated into a company.

Consequences

Lack of Cohesion

The "survivor syndrome," which affects employees that have avoided a wave of layoffs at their company, is well documented in management research. Its effects include loss of loyalty, loss of motivation, lower performance, distrust of management and other employees, and sometimes even depression and sabotage.

For an industry that depends on interpersonal communication for success (and uses this fact to justify its reliance on full-time on-site employment even in situations where a more informal working relationship would work just as well), the survivor syndrome can be devastating.

Employee Ramping-Up Costs

In a context of full-time employment/layoff cycles, the staffing process can become very costly. Recruiting either a new or replacement employee requires sourcing and sifting through applicants, scheduling and conducting interviews, and possibly paying a fee to a recruiter. In the case of the replacement, all of the interim time between the departure of the original employee and the start of the new one represents lost productivity. Once hired, current staff must be assigned to train the new hire, who may not be productive for several weeks, and will certainly not reach peak efficiency for several months at least, especially if she is a programmer working on proprietary technology.

Layoffs also come at a price that is not limited to severance pay: employees worried about an impending layoff tend to "make themselves indispensable" by stretching their work over longer periods, and sometimes even to devote more effort to finding their next job than to completing the current one.

Finally, given the industry's lack of standard job descriptions and management tools, it is very difficult for an employee to maintain all of the skills that may be required to find a new job. For example, much of the experience gained by working on a company's proprietary scripting system doesn't translate to another's – and it is quite difficult to find two companies with the exact same definition of a producer's responsibilities, never mind the actual processes involved. This problem is a source of stress for individuals and costs for studios.

Impact on Productivity and Slippage

Job instability exacts a high toll on our companies. Instead of enjoying the productivity gains that come with the cohesion and shared experiences of veteran teams, we are constantly starting over from scratch and suffer the consequences of "New Team Syndrome." [Collins02]

Every time we build a new team or substantially alter a veteran team, we must once again pay the price of building a new team culture from a set of individual personalities and interests. Until the team has invested a substantial amount of time together, communication is at the lowest efficiency it will ever be, estimates are the least accurate, and ability to manage the team is limited by an underdeveloped understanding of the new areas of accountability. The impact of New Team Syndrome upon the schedule and costs of projects is substantial.

Most studios don't have this money to burn. When an estimate is wrong and communication is inefficient, the costs hit our bottom lines and make it that much harder to succeed and survive. It is important to our ability to grow as industry to grow beyond the mercenary system that has got us here: no empire can survive long without the loyalty and full productivity of its own army's veteran soldiers. The studios that will have the greatest staying power in the years to come are the ones that invest in their people and whose people invest in them.

Best Practices

Employee Retention – Not With No-Compete Clauses

Obviously, employees who are treated well tend to stay in one place, while contractors who have good experiences with a company will make their special skills available to it again in the future. Ways to improve team spirit include:

- Running a solid, profitable company that has as one of its core values – if not its most important one – the well-being of its employees.
- Avoiding layoffs, by planning future projects ahead of time, building safety margins into contracts so that there is money to pay people during (hopefully limited) down time, etc.
- Communicating information about the company in a timely and straightforward manner so that everyone feels involved and well-informed.
- And of course, solid, well-planned projects.

Some companies try to defend themselves against employee departures by enforcing no-compete clauses. Not only is this practice illegal or legally impractical in some areas, it is also counterproductive [Laramée03].

- Fear is generally a poor motivator, particularly over the long haul, and employees who stay with a company under duress are unlikely to be good performers.
- The existence of the no-compete clause makes the studio unattractive to potential new hires, making it more difficult for the company to recruit quality employees.
- The industry as a whole suffers from a chronic penury of experienced developers; if a no-compete clause forces some of them away for a period of several months or years, we are only compounding the problem, and they may not come back.

Effective Developer-Publisher Contracts

In the Developer-Publisher relationship it is important that both parties recognize that legal issues can cause stress to both management and staff. While necessary legal protections are simply good business practice, care should be taken not to go overboard. The IGDA website provides numerous resources on this subject:

www.igda.org/biz

Standardizing Job Descriptions and Tools

As an industry, we stand to gain a great deal from standardizing the responsibilities of the various professional groups involved in our work, especially those whose jobs are

currently vague or unpredictable, like the producers, associate producers, executive producers, associate designers, lead designers, etc. Doing so would make it easier for employees to maintain sellable skills and move from job to job, and it would also make it easier for studios to recruit people on short notice and have them be effective from day 1.

For similar reasons, we stand to gain from developing standards for technology and tools. The IGDA's Artificial Intelligence Special Interest Group is developing such standards for their area of expertise; consolidation in the middleware market, which would allow programmers to become "Engine X" specialists and producers to have game-specific asset and project tracking tools at their disposal, would also be saluted.

WORK ORGANIZATION ISSUES

From the Interviews and the Message Boards...

“Even when we work long hours, it’s a good atmosphere. Everyone is friendly and in a good mood. You have to perpetuate your own happiness. You have to keep it going.” – Programmer, Canada.

“We always have 3 active projects and 2 prototypes in the pipe. There are always 3 teams on top of engine/tech teams. Tech group is a middleware provider to the production teams. Works great. We build all the tools and the engine – works a lot better than having one per project.” – Programmer, Canada.

“Some companies intentionally scheduled things so that people worked more hours. [...] There was a lack of pre-production and scheduling. [...] Inevitably, the project would be late. The projects were constantly scrapping everything and restarting from scratch in the middle.” – Programmer, Orlando.

“[At Angel Studios, now Rockstar San Diego, we have] made progress towards a mentorship program. At a mid to large company, there’s got to be somebody brilliant you want to talk to. Go grab lunch together every few weeks! [...] An R&D group can get people excited depending on the coolness of the demos that come out of it. Maybe this can be a rotating group or a reward for good work.” – KahnJekarl, Programmer, New Jersey.

“Our studio has an internal lunch room with a chef who cooks really great food. That probably adds to the morale a bit 😊 Sure beat’s Wendy’s!” – GhostGirl, QA Tester, Maryland.

From the Survey...

- Only 16% of respondents said their companies had formal change control policies.
- Respondents describe their companies’ pre-production staffing plans and schedules as wishful thinking (32.4%) or “reasonable in most cases, but occasionally flawed, leading to tense periods” (38.9%) far more often than as “very” or “sufficiently” accurate (13.5% total.)
- 46.8% of respondents say their overtime isn’t compensated at all. Only 4.3% of companies compensate overtime in cash.

Preamble

While the industry's economics are, to an extent, out of our control, the ways in which we run our companies are very much in our hands. Effective work organization, credit assignment, compensation structure and management training are but a handful of ways in which we can make life at work easier for all of us.

Causes

Lack of Management Training

More often than not, game development leads and producers are promoted from within: programmers, artists and designers who are very good at what they do but may not have received any formal training in management before they are thrust into roles of responsibility. As a result, they must learn by trial and error, sometimes through mentoring by other developers who have gone through the same ordeal, often on their own.

Indeed, few people combine the skills and personality types required to become an effective manager with those required by front-line game development. Thus, all too often do we see very good developers become unhappy leads, or professional managers/producers whose very useful skills are sneered at by non-developers who feel that the managers don't know what they're talking about because they have never programmed a game.

Whatever the case, lack of preparation and trust within the team makes it exceedingly hard to plan new projects (how do you know when to take someone's time estimates at face value and when to add comfortable safety margins?) and to resolve conflicts when they arise.

Credits

All too often do we hear complaints from developers who don't receive the on-screen credits they feel they deserve, either because they have left the company before the game was released, because publisher staff have flooded the credit list or for no identifiable reason at all.

Ineffective Compensation Schemes

Money is a poor motivator but a powerful de-motivator. Companies with ineffective compensation schemes experience quick (and often irreversible) declines in morale and productivity. Here are some examples.

- **Royalty-heavy pay.** For a game company, compensation heavily biased towards bonuses and royalty sharing can be tempting. For one thing, it helps keep base salaries low, thus allowing the company to complete projects on smaller budgets and therefore increasing the odds of obtaining contracts from publishers. For another, in the event of a hit, everyone involved can expect a handsome payoff. However, given the fact that only a small fraction of games ever recoup their advances, much less generate significant royalties, this is usually a fool's bargain.
- **Non-compensation of overtime hours.** Nearly all game developers in the U.S. are under "exempt" status, meaning that their overtime is uncompensated. Developers elsewhere must often work under similar conditions. This is

unhealthy. While it helps companies control costs, it also incites abuse; uncompensated overtime is free work, and free work is easy to sell at a profit. Non-compensation of overtime hours creates a climate of (all too often well-founded) distrust of management and disinterest in the company's welfare, at least in developers who have passed the idealistic phase that immediately follows their entry into the industry. Much more effective as a long-term team-building tool (and not as financially damaging as time-and-a-half) is compensation in extra time off once a milestone has been completed or a project ships; this is the strategy adopted at Digital Fiction, among others.

Exploitation of Non-Immigrant Visa Workers

For some time it has been easy to hire foreign workers on non-immigrant visas, in the USA most notably the H-1B visa for Europeans and the TN-1 for Canadians. All too often, these workers are paid considerably less than their local counterparts, thus yielding considerable savings to the employer – until the employees discover how much more their colleagues are earning.

Some employers may hope that a non-immigrant will see the bigger economic picture. Others may have more unsavoury motives; someone once suggested to one of this paper's authors that foreign workers know that they will be paid less, in exchange for the chance to get a foothold in the US, that they are a knowing accomplice in what is an illegal act, which thus goes unspoken and unwritten. It is, however, far more likely that they will feel taken advantage of. No-one likes to be duped. An informal poll of several British game developers working in the USA reveals that they never considered themselves "cheap foreign labor".

Permanent Work Permits

When a foreign national is the only person for the job, a company should consider the long term and plan for an immigrant visa if the employee wants one.

However, prospective employees must make sure that the employer promising assistance in obtaining such a visa is reliable and worthy of long-term employment, and then they should monitor the process closely all along the way. A "green card" application (in the U.S.A) can take up to 2 years to process, and the application is cancelled if the employee leaves the company for any reason. This naturally leaves the employee in a poor bargaining position if the situation turns sour. (Employers who abuse this situation will be stranded with employees who should quit, but stay and pollute the working environment as they count the days until they receive the green card.)

We have even witnessed examples of companies wilfully delaying the procedures required to provide an employee with a green card, even when one is specified in the employee's contract. In one case, the Lead Programmer of a project in beta discovered that the company had stalled his green card application a year earlier, and that instead of being nearly complete, it was sitting in a lawyer's desk, not even filed with the

government. Needless to say, the timing of this discovery was less than optimal for the project as a whole.

An individual who has the drive and imagination to leave their home nation is not likely to balk at moving across the street, or across the country, to find a new job that isn't started on such a negative foundation. They might also report their former employer to the State Department.

Floor Plans

More and more companies eliminate closed office space and replace it with open floor plans to favor teamwork and cut real estate expenses. However, noise carries much further in an open office than in one with enclosed spaces. The same is true, to a point, when open offices are compared with traditional high-tech cubicle farms.

Lack of Candid Communication between Teams and Management on Quality of Life Issues

Often, solvable quality of life problems can appear, or be allowed to persist, simply because good communication doesn't exist between the development staff and management. Many managers, even those with good intentions, and particularly those who are new or untrained, don't grasp that they need to actively solicit input from their teams about the best HR and development practices to adopt for that specific team. Not doing so can result in an avoidable mismatch between the policies that are in effect, and the policies that are most likely to keep the staff happy and productive.

Consequences

Lack of Personal Commitment to Accountability

When a climate of distrust exists between developers and management, when employees do not feel respected, or when conflicts fester between members of the team, it becomes very easy to blame others for failure instead of pulling together to achieve success.

Effective work organization, on the other hand, stimulates personal involvement in the company. Developers who truly feel "part of the team" and don't think that management views them as interchangeable are more likely to take the rough times in stride and keep moving.

Loss of Loyalty

Lack of fairness, whether real or perceived, destroys loyalty. Employees become more susceptible to "poaching" raids from competing companies. Articles lambasting the company appear on industry gossip web sites, making recruitment harder. Work takes a backseat to office politics. In the end, nobody wins.

Best Practices

Standardized Credits

Right now, credits are sometimes negotiated, often assigned at the discretion of the employer, and sometimes hard to associate with any actual work. As a result, game credits don't enjoy the level of credibility generally associated with those assigned in cinema or television.

Standardization of job titles and credit assignment would remove a major bone of contention between individuals, development studios and publishing houses.

- When a dispute occurs, a standard will bring clarity to the situation.
- The standard would also be very useful in resolving what happens when an employee leaves during the middle of a project. Some employers feel justified in removing the employee from the credit list out of spite or to provide greater visibility to the people who remain, no matter what the departing employee's contribution to the project might have been.
- The same applies when the employee leaves between the end of the project and its market release.
- Finally, one can argue that credits are a factual statement of who did what, rather than a discretionary "bonus". Consequently, leaving off an employee or ex-employee might be construed as an act of libel. Standards would certainly help protect everyone.

Some employers may feel that the threat of withholding credits may give employees second thoughts about leaving in the middle of a project. However, since everyone is aware of the system's shortcomings, this strategy isn't likely to have much impact: word of mouth plays an important role in our industry, and work will be recognized whether or not a person's name appears in the manual.

Until there are standards, employees should negotiate their credits as part of their contracts – and not be afraid of using word of mouth to receive the appreciation they are due if an employer does them wrong.

The IGDA has embarked on an attempt to standardize credits with the establishment of the Credit Standards Committee.

Separation of R&D and Product Development

The two principal sources of excessive work intensity, according to a Shani & Sena article included in [Docherty02], are feature creep and time to market. Nowhere is this more evident than in the Christmas-centric, bleeding-edge, change control averse game industry.

To make it easier to plan development and deliver according to schedule, game companies should consider:

- Creating a separate research and development team, who do not work on any specific product-related deadlines and therefore can freely experiment (and occasionally fail) without putting the entire company in jeopardy.
- Basing as much of their product development as possible on a "common company platform" of proven, reliable code. This platform can include a mixture of code delivered by the R&D team and external engines and libraries.

Support Different Career Paths

Effective companies recognize that not everyone has the same priorities. Some people want the responsibilities associated with management and leadership roles. Others prefer to push their own skills to the limit and assume a supporting role in the team. And the relative importance of career and other goals is also a fundamentally personal choice.

- Make the company's culture clear to potential employees during the interview process to ensure that only people who will fit in will actually accept job offers. Ensemble Studios and Big Huge Games, among others, push this to the limit by giving employees veto rights on any new hires. Whether or not your own team can afford to go this far, make sure that candidates know what your company values above all else (Short projects? Creation of original intellectual property? Hard work? Family-friendly games? Team spirit?)
- Accommodate different career paths for people of different goals. At Nortel, there are separate career ladders (with identical salary and benefit scales) for line managers and for technical specialists with unique skills but no leadership ambitions.
- During annual reviews, set explicit goals with the employee, making sure that these goals are compatible with the employee's own priorities. Overloading an employee who wants to spend more time with a young child or pursue graduate studies is counterproductive for all involved. So is denying a career-oriented person the opportunities for growth they crave.
- Define success in terms of goals (i.e., accomplish X by date Y) instead of processes (i.e., never leave before 9PM the week before a milestone). The difference is sometimes subtle, but work tends to expand to fill all time allotted to it, so enforcing long hours regardless of output will lead to diminished performance.

Leverage Remote Developers

Counter-Strike, arguably the most successful online multiplayer game of all time, was developed by a virtual team over the Internet. [Apgar98] evaluates the total number of telecommuters and home-based workers in the USA at 30 to 40 million people, a number that can only have grown since the Apgar study was conducted, given the increased availability of broadband Internet access.

And yet, many game studios insist that all of their collaborators work on the company's premises, all the time – even when flexible schedules mean that "face time" between team members is limited.

Allowing telecommuting or hiring remote contractors can have many advantages:

- **Cost reductions.** Thanks to its telecommuting program, IBM saved \$100 million a year in unneeded offices and overhead during the mid-90's. The savings included a 42% reduction in real estate expenses.
- **Higher productivity.** Telecommuters don't waste time in traffic and idle water cooler chat. In a study of one *well-managed* office, distractions related to the social norms and the general "atmosphere" of the place amounted to 70 minutes per day, per employee [Apgar98].
- **Easier recruiting** for companies located outside of the main industry centers.
- **No relocation fees** to pay.
- **Higher job satisfaction.**

The common argument in favor of on-site policies, that it increases team communication, is a valid one. However, much effective communication can easily be managed through email, instant messenger, conference calls, and a minimum of "face time". In larger companies, the less formal nature of email communication may actually increase the flow of information, as it flattens hierarchy, is easier on the shy, and doesn't require synchronizing the busy schedules of many people at the same time.

Effective Floor Plans

[Tremblay03.2] lists precautions that companies can take to maximize the cost reduction while keeping the additional distractions associated with an open office to a minimum:

- Using 120-degree separators instead of traditional square walls may allow 50% more people to occupy the same square footage.
- High-traffic areas (like the kitchen and the photocopier) must be isolated from the work areas.

- An open office requires additional enclosed meeting rooms for team exchanges and for confidential phone calls.
- It is sometimes useful to install noise-canceling devices, especially to dampen the impact of voices on other employees.

Customizing HR Policies to Achieve the Optimal Quality of Life for Your Team

One of the biggest challenges in implementing an optimal set of policies to raise the quality of life at your company is that many of the most effective policies are expensive in terms of resources, usually time or cash. It's all very well to recommend fantastic health-care benefits, no overtime, and weekly massages for all staff, but if you're trying to keep your burn rate down in order to survive until your next deal comes through, some of these options can seem beyond reach.

(It's also worth noting that many quality of life improvements have nothing to do with resource expenditure, and you have no excuse for not pursuing them!)

As a consequence of individual feedback, it can be very effective to tailor your HR policies to meet the specific needs of each employee. Given that different team members have different strengths and different needs, it is often possible to significantly raise a specific team members' quality of life with a "custom deal." Such deals might include:

- Working from home one or more days a week
- Non-standard hours (such as 4 10-hour days)
- Taking an leave of absence (for instance, after a project is complete)
- Special desk/chair/monitor/input device (for someone with specific physical needs or task requirements)
- Financial support for training or other schooling

It is not necessary to broadcast the specifics of each employee's particular arrangement to the team at large. In the same way that individual salaries are private, the particulars of individual arrangements can sometimes stay private, which minimizes the likelihood that other team members become discontent. It is necessary to walk a fine line; you don't want separate arrangements to turn into full-blown favoritism (or be perceived as such), but you want to retain enough policy flexibility to make each team member's life easier in ways that especially matter to him or her. If your resources are limited, it makes sense to apply them where they will do the most good. How do you find out where they will do the most good? Ask your staff!

Developing Lines of Communication between Management and Staff to Improve Quality of Life

Each development team and project is unique in terms of its strength, weaknesses, and needs. If you are trying to maximize quality of life for your team, it is essential to know what their quality of life priorities are.

For instance, if your team is predominately young and single, they may not be overly concerned about comprehensive health care, but very interested in flexible scheduling, such that they can come in late, take time off, etc. Conversely, if your team has lots of married folk with families, health care coverage may be a major consideration.

It is often possible to come up with flexible, creative solutions to HR problems, just by soliciting information from your team, and working with them to craft individualized solutions. Some strategies to pursue:

- Encourage your leads and department heads to solicit information from their staff about problems and grievances they may have, and to bring such information to your attention. Often, employees can be too intimidated to raise legitimate issues directly with senior management.
- If you have time and can establish a rapport, try to connect with team members directly to get a sense of what their issues are.
- A project postmortem is a very useful way to give your team a forum to express criticisms of the development process and overall team quality of life. For this to be effective, the conclusions need to be summarized and presented to the team, as well as a plan of action to respond to the conclusions.
- Keep your team informed about the broad goals of the project and company to as great a degree as is possible. If the team members know what the plan is, they are much more likely to synchronize their personal priorities with it.

Warning! One consequence of soliciting candid information from your staff is that you will end up on the receiving end of criticism that you don't especially feel like hearing.

- Perhaps management deficiencies will get pointed out that you're not prepared to address. Don't get defensive! It's almost certain that you could be better at your job. As uncomfortable as it can be, honest criticism is a gift, and should be received as such.
- Due to the vagaries of human nature, some team members may come forward with unfounded grievances or unreasonable/impossible demands. It is essential to draw a distinction between establishing communications with your team, and feeling obliged to meet any demand that any given person might have.

Generally, if you are truly listening, and doing your best to come up with solutions to the problems that are presented to you, things will turn out OK.

LESSONS FROM OTHER CREATIVE INDUSTRIES

Transitioning From Other Industries

Background

Despite its foibles, the video game industry holds a certain romantic appeal for those in other fields of work. Some of this has to do with the sheer “wild west” aspect of new technology. This attraction is far less common in film, television and traditional computer software in which fields the technology and industry norms have become rather fixed.

Strong industry growth has spurred the need to transition talented people from other industries into our own. As we face new challenges, we need to acquire skills that are more common in other fields – skills related to human resource management, software engineering, finance, and organizational behavior are needed to take the industry to the next level of prosperity. Game studios are just beginning to understand how to achieve this transition smoothly.

Challenges and Rewards

Transitioning talent from other industries brings quick access to needed skills and idea cross-pollination. However, integration requires effective cultural and skills assimilation. The companies that become really good at it will benefit from a larger pool of talent to recruit from – a powerful advantage in an industry chronically short of senior staff, especially during growth periods. Better yet: these companies will be able to take advantage of the best practices developed through decades of trial-and-error by other industries.

The move to bigger budgets and the bigger companies needed to manage them may have a deleterious effect on the garage-hacker culture that has fuelled the birth of our industry, but since the trend is likely irreversible, our goal must be to find out what quality of life benefits can we get in the bargain and how much of our original culture is worth preserving.

In short: our industry as a whole will likely enjoy a strong boost to our quality of life from the addition of people from other industries, for the following reasons:

- They bring expectations of reasonable working conditions that accommodate family life, community, and life outside of work.
- They bring efficiencies and practices that, as long as they are adapted in a manner that respects and accounts for our culture and way of thinking about things, will provide huge economic advantages.

- They connect us with the broader markets beyond our traditional audience. As we increase diversity in all areas of our industry, we will gain the kind of perspective that will enable us to build even better products with wider appeal.

By carefully preserving our innovative culture while taking advantage of the new ideas and skills our new teammates offer us, we can get the best of both worlds.

Transitioning New Talent

A hire from another industry can bring a wealth of experience and new skills to a game studio. But in order to make the most of your new addition, it is important that your firm has in place additional support to aid the transition process. Your new team member will no doubt be eager to apply his skills and help his new team, but he'll be better equipped to offer his assistance if you invest enough effort to bring him up to speed on your company's culture – and do so early enough.

In the past, many studios have been unable to absorb new talent because they have underestimated the importance of helping their new team members through this “cultural learning curve”. An under-planned attempt to add a team member who brings with him a different outlook and set of assumptions about business life can be painful for the team and the new hire. The challenge is to preserve the good parts of your studio's culture while introducing new ideas that will move it forward.

(It can be tempting at times to cut to the chase and try drastic new ideas, but you do so at your own peril. A better way is to take the time to bring everyone into parity and then make gradual improvements with a longer term goal in mind.)

Focusing on the cultural integration of your new hires is probably the single most important thing you can do to make his transitions smooth for his sake and yours. The better you are able to educate him about your team's expectations and working habits, the sooner he will be equipped to apply his skills in a manner that is both innovative and likely to be well received by your team. Get the acculturation right and the skills payoff will soon follow. Here are some ideas about what your studio can do to smooth the transition:

- **Mentoring.** Assign the new team member a veteran as a mentor. Having someone who is proactive in helping get the new hire up to speed is even more important for someone making a big transition.
- **Disseminate company culture.** Talk to your new hire about how work gets done, how decisions are made, how responsibility is assigned, how accountability works, how to advance, and what to expect. These things may seem obvious to you, but someone coming from another industry may have very different expectations about the ‘normal’ way of getting things done.

- **Integrate the whole person.** Involve your new hire in the social aspects of your team as much as you can. It will help him gain an intuitive understanding of the culture in your studio.
- **Homework.** Assign him some reading homework about the industry. An understanding and appreciation for the history of the industry and how things have gotten to be the way they are today can be a big help.
- **Be patient.** You may not enjoy the advantages of your new hire overnight, but as soon as he's settled you will have something most studios don't—a talented individual with a package of new best-practices and new skills who also speaks your team's language.

Lessons from Software Engineering

There's no denying that software development is usually the key element driving a game's schedule – including extended overtime and slippage. Oftentimes, problems arise because the game must not only conform to specification like any other software product (i.e., do what the design document says, with no apparent bugs), it must also be fun. It is not enough for the game to adhere perfectly to the original design, and for the art to meet all standards, and for the engine to be reliable and robust: the fun factor introduces a degree of difficulty to the production of games that far exceeds that of any other software project of comparable size.

Having said that, there are significant lessons that game developers can learn from the discipline of “standard” software engineering. Over the past 10-15 years, we have seen considerable amounts of research into development methodologies such as Waterfall, Spiral, Prototyping, Design-to-Schedule, pair programming, and extreme programming.

Here is a brief description of the better-known methodologies.

Pure Waterfall

Undoubtedly the oldest of standard methodologies, it serves as the base line for other methods. In a Pure Waterfall model, a project progresses through an orderly – some would say rigid – sequence of steps from concept through testing, and is very document driven.

This model is appropriate for well understood but complex projects. It works well if you have a technically weak or inexperienced staff, as it provides considerable structure to minimize wasted effort. Unfortunately, it maximizes quality over time and cost, and it's very difficult to “go back” and correct mistakes that were made in any prior phase, which clearly makes it a poor choice for game developers.

Code and Fix

This method is just like it sounds. There's very little (if any) upfront design, estimating or scheduling. You just jump in and start coding.

This model can be useful for small projects that will most likely be thrown away: proof of concepts, demos, etc. While you save big-time on project overhead, there are obvious downsides: there's no means of assessing progress, assessing quality or identifying risks. If you were to discover halfway through your project that your underlying design is flawed, you'd have no choice but to throw it out and start over.

Spiral

This method breaks the software project up into smaller sub-projects, each covering a major area of risk. When all areas of risk (broadly defined) have been addressed, the spiral approach concludes as a waterfall methodology would.

A big advantage of this model is that as costs and investment increase, the risk decreases, since risk is progressively lessened by definition. The downside of this approach is that it's extremely complicated, and requires very experienced management to implement and track.

Modified Waterfalls

There are several of these methods, but in general, they boil down to this: while the pure waterfall method is clearly flawed, all of the elements inherent to that method, i.e., concept, requirements gathering, design, architecture, code, debug are necessary at some point. The problem with pure waterfall is that these elements are discrete and sequential; overlapping these phases, constraining the amount of documentation required, and allowing for more regression will solve it. You can also, in effect combine waterfall with a spiral methodology for highly risky projects, or those that contain discrete elements of high risk.

Evolutionary Prototyping

Just like it sounds, evolutionary prototyping has the team developing the system concept as they move through the project, usually starting with the most visible systems. Then you refine and repeat, until the "prototype" is good enough to be released as the final product.

This methodology is most useful when the project's requirements change rapidly. However, though it produces signs of steady progress almost immediately, this technique has the major drawback that it is hard, nigh impossible, to determine exactly how long it will take to get to project completion.

Design to Schedule

The product is developed in successive stages, the main caveat being that you don't necessarily know at the outset that you'll ever make it to the last stage.

This methodology ensures that you'll always have a product to release by a certain date. There are many games and many developers who employ this methodology, or versions of it, some with great success. If you manage the scope of your product carefully, you will have prioritized your most critical features up front, and you have a great chance of hitting your dates. Then the question becomes: is it good enough?

Of course, if you don't get through all of the stages, you'll undoubtedly have wasted some amount of time specifying, architecting and designing features that don't end up being implemented, and you could have spent that time completing or refining other features.

X-Treme Programming

This is one of the newer, hotter, and fairly controversial programming methodologies being used today. In essence, it could also be called team programming. A team of developers (usually two) write code jointly, most often on the same machine at the same time. They discuss what they are doing, de-bugging and modifying on the fly.

The upside of this approach is that the code is generally more robust and bug-free than would otherwise be the case, and that the team as a whole is less likely to lose valuable expertise because at least two people are intimately familiar with every piece of code. The downside? The process is slower. You basically have two people doing what up until now was the purview of one. In theory, the method saves time in the long run, as the code is less buggy and better thought out.

Extreme programming has much to recommend it, but so far has been identified as most useful for large, iterative projects, such as a proprietary corporate applications, as opposed to standalone software solutions such as games.

Purposes of the Methodology

Whichever method a given studio decides to adopt requires several features to be of any value:

- A disciplined, organized approach to estimation and scheduling.
- Detailed and accurate tracking and measurement of the project's progress throughout its life cycle.
- A change-control policy and fallback plans in case the project spins out of control. If it becomes necessary for the team to work excessive hours ("excessive" being a relative term depending on the company's culture, but generally meaning "over 40-50 hours a week") for more than 2-3 weeks at a time to meet a deadline, then it is more than likely that the project is out-of-control. At this point, something needs to give: the ship date needs to be revised, the schedule/feature set needs to be adjusted, or (if feasible) additional resources need to be added.
- Solid managers that can motivate and lead the team to a successful conclusion.

Lacking those necessary elements, any software project, and that includes any electronic game project, is truly fighting an uphill battle.

Other Industries' Models

Every industry has its own funding and development model(s):

- Movie studios may commission a script, which may or may not lead to pre-production, which may or may not lead to a shoot. Some movies are financed by independent production companies, others by television networks, or even by the producers' own personal funds.
- Television series have a similar model, with the pilot episode replacing or supplementing pre-production. Only a minority of pilots will get "picked up" for a full series.
- The book publishing industry's model of advances and royalties most closely resembles our own. However, only a very small percentage of published writers earn enough to live off their craft.
- Music, theater, and the fine arts have financing models that are often completely unique to them.

While importing some aspects of film and TV financing into the game industry (e.g., separate envelopes for extensive pre-production and pilots before signing a production contract) would undoubtedly prove beneficial, defining how to do so is outside of this paper's scope.

OWNERSHIP OF WORK

Preamble

Generally speaking, an employee's work done for an employer, that relates to that employer's business, is done on their premises, with their tools, and is within the scope of the employee's job is the property of the employer.

However, what of the side projects that many developers engage in at some point in their careers? If you have an idea that doesn't fit into your company's current project, and develop it in your own time, is it still yours? If you create a demo while between jobs, then bring the code to work and integrate it into a commercial game, do you still own rights to your creation? Questions regarding the individual's right to his or her own work and the employer's right to ownership of ideas not strictly produced as part of a person's employment can sometimes become a significant source of stress and doubt.

This section attempts to clarify the intellectual property issues that can arise in the normal course of an individual's business relationship with a company. Make sure that your employment agreements specify ownership rights explicitly, and that they conform to the laws of the land.

Work Done For Others During Time Off

Another aspect of no-compete clauses has to do with the work an employee does outside of working hours that may or may not be within the scope of their employment. It is not unusual to have employment agreements restrict or prohibit employee activity outside of the workplace that relates to the work they do for their primary employer – and in extreme cases all work done outside. Such a position could take the form of a restriction or a statement that any outside work is owned by the primary employer.

Ambiguity, however, can come into play when an employee does not have a succinct job description that defines the scope of their work. Likewise, if the business of the employer is not narrowly limited, it may be difficult to define what activity competes with the business of the employer and what does not. For example, if a game animator working for "Game Co." freelances at night and on weekends creating characters and storyboards for "Toon Co.," which characters and boards are intended as preproduction elements of an animated television program – does that activity compete with the business of "Game Co.?" The answer may not be obvious. If Game Co. intends to exploit the employee's work outside of games or the owner of the animated program intends to cross-utilize the freelance work, perhaps in games, then "competition" becomes murky.

An Employee's Original Creation

What if the employee creates a work on his or her own time, in their home, utilizing their own materials?

For example, let's say this work is not a video game but a graphic novel (comic book). Lacking clear definition of the employer's business area and the scope of the employee's engagement, the intellectual property rights may become entangled. The situation becomes even more problematic if employees are allowed by the employer to use its facilities and tools (work stations, software etc..) for "their own projects." Who then owns the work? The answer can be very difficult to ascertain and is dependent upon the statute and case law of the jurisdiction(s) in which the work occurs. Few things are more stressful in the creative environment of the game studio than a feeling that "the boss 'stole' my idea." Therefore employers should be careful to make the lines of ownership clear and, where possible, give some leeway toward allowing workers, particularly artists and designers, to have creative outlets outside of the game studio.

Developer Principals

Unless otherwise provided for in an employment agreement, the work of a Corporate Officer, LLC Managing Member or L.P. General Partner that relates to the entity's business is owned by their employer, inasmuch as those individuals owe a fiduciary duty to their organization and its equity holders – this is the case in US and Canadian law.

Independent Contractors

An "independent contractor" can be an entity of any size, from a single individual to a large corporation with hundreds of employees. Generally speaking, an independent contractor agreement (either written or oral) and the jurisdiction's statutes and case law, dictate who owns what. It is in the best interest of both the contractor and the client to define ownership and competitive restrictions in an express written agreement before work starts.

The "Moral Rights of Authors"

Some countries subscribe to a concept called the "moral rights of authors" or "artists" regarding intellectual property. For more information review the IGDA's Intellectual Property Rights Committee White Paper, available on www.igda.org.

Original Works to Which the Developer Owns IP Rights

A developer who creates an original game property, then proceeds to fund it and put it into production, generally takes a lot of risk. Sometimes the savings and assets of the developer's principals are on the line, other times venture capital is involved, and in all cases the developer risks insolvency (or investors taking control of his company) if "the original property" doesn't succeed.

The pressure is enormous – the kind of thing that leads to substance abuse, marital problems, strained friendships and the like.

On the other hand, very large profits can result from a property that "takes off". It is the risk-reward trade-off of free enterprise. However, this risk is not always understandable

(or a point of empathy) for “worker bee” members of a team who may not hold equity in the company or royalty rights on the property. Sensitivity on both the side of the employer and employee to the other’s position can mean the difference between a thriving, happy environment and “a working hell.”

On Contracted “Work-For-Hire” Projects

These projects take two forms.

- The first is when a publisher engages a developer to create a game derived from an existing intellectual property – say a movie or TV show. In this scenario, the developer usually does their work on a “work-for-hire” basis, assigning all rights to the publisher, warranting that they owned the rights to assign, and agreeing to assist the publisher in perfecting any ambiguity of ownership that may arise later. Generally the publisher will also require the developer, its employees and contractors to sign agreements – the form of which may or may not be dictated by the publisher.
- The second form of “work-for-hire” agreements is when a developer creates an original property and then transfers the rights to a publisher in exchange for a development-publishing agreement.

Both of these types of agreements may provide royalties based on the commercial success of a product. Contracts are sometimes rather strongly worded and can be offensive, particularly to the less affluent, and often under-represented, developer. Again, sensitivity on the part of both parties is key to keeping a pleasant atmosphere and working relationship.

For More Information

For more information on related topics, please read the IGDA’s Intellectual Property Rights White Paper, which can be downloaded from http://www.igda.org/biz/ipr_paper.php. Other materials published by the IGDA’s Business Committee (<http://www.igda.org/biz>) may also be of assistance.

RELATED WORK FROM OTHER IGDA INITIATIVES

Best Practice Reports

The Business Committee has published a number of reports on best practices in various areas of game development, including marketing, contract negotiations, etc. Most of them compile the results of GDC roundtable discussions and/or surveys of practicing professionals. Of specific interest to quality of life are the *Human Resources Best Practices*, *(General) Resources Best Practices*, and *Human Resources Benchmark*, all of which can be downloaded for free from www.igda.org/biz.

Here are some of the strategies recommended in these documents:

- When creating a schedule, some tasks (and therefore some elements of a game's intended content) must be tagged as easily removable. This way, if changes must be made to the project in midstream, it will be practical to implement cuts instead of forcing overtime.
- The lead programmer, lead artist and lead designer should not be assigned specific tasks when creating the project schedule. Use them to plug holes, assist on tasks that slip, and act as buffers for the team in regards to external needs (for example, last-minute requests from marketing).
- Schedule buffers to protect the schedule and the team in case of slippage. Assume no more than 11 months of effective work a year and buffer each task by 15%-50% depending on complexity and predictability.
- Schedule no work on Fridays, using them to catch up on slips.
- Build a system to identify discrepancies between an individual's task duration estimates and the time they require in practice. Use the trends to create more realistic schedules in the future.
- Share information such as contract and financial data with employees, so that they understand management's decisions and their ramifications.
- Design bonuses and other forms of variable compensation in such a way that people easily understand the payout thresholds.
- Use 360 degree performance feedback. This system, which solicits evaluations from peers, bosses and direct reports, has been increasingly embraced as the best way to collect performance data. Gone are the days of working hard to impress only one person; now, the opinions of everyone an individual interacts with on a daily basis matter.

- When asked about the things that make companies great places to work, responses concentrated not on financial rewards, but rather on people-oriented factors like management consistency, exciting projects, commitment to people, a listening environment, and feeling valued.

Intellectual Property Rights Committee – Best Practices

The fundamental way in which quality of life can be effected by intellectual property rights is the simple fact that ownership, whole or in part, of valuable intellectual property is a source of revenue. The key focus of the Intellectual Property Committee is Intellectual Property (IP) matters that relate to the independent developer. The following are key areas from their 94 page White Paper [IGDA-IP03] that relate to quality of life in a specific sense:

- **Copyright** (pg 7) can enhance the quality of life of a work's creator inasmuch as it allows the creator of a work certain legal rights to exclusively exploit their protected work in the marketplace. Often these rights are transferred either by virtue of the work having been a "work-for-hire" or through a later license or assignment of the created work.
- A **patent** (pg 8) is the "exclusive right granted to an inventor to control production and distribution of an invention." As with a copyright it can cover a work made as a "work-for-hire" or can be licensed or assigned.
- **Moral Rights of Authors** (pg 43): Some countries, though not the U.S., recognize the right of a creator to protect their reputation and interest by providing certain rights by virtue of creation. This is a complex area of law.
- **IP Rights Agreements** (pg 49): By use of agreements, in accord with applicable law, creators can reserve, license or assign rights in their IP.
- **Insuring Against IP Risks** (pg 70): Insurance can provide financial protection against claims brought by or against a creator or assignee for infringement of IP rights.

Contractual Best Practices

The IGDA's Contract Walkthrough [IGDA-BIZ03] focuses on contractual issues that come to light in the day-to-day operations of independent game developers. The key passage pertaining to quality of life issues is the one on the moral rights of creators (pg. 32).

Credit Standards Committee

The IGDA has also recently established a committee whose mission is to study, document and propose voluntary game industry crediting practices that properly recognizes and credits those responsible for the creation of games. The Credit Standards Committee can be reached at <http://www.igda.org/committees/credits.php>

Women in Game Development SIG

This special interest group has established a mentoring program, community opportunities, and various other means of easing the way for women looking to enter the industry. They can be reached at <http://www.igda.org/women>

Human Resources SIG

Human resource managers are on the front line regarding the consequences of poor quality of life. Thus, the Human Resources SIG's work is in some ways closely related to our own. Visit them at <http://www.igda.org/hr/>

IGDA Chapters

Local IGDA chapters can be found in over 30 countries around the world. Meetings provide excellent ways to connect with fellow developers, unwind, and exchange ideas. Look for the chapter closest to you at <http://www.igda.org/chapters>

APPENDIX A - BIBLIOGRAPHY

Books

- [Adams03] Ernest Adams, *Break into the Game Industry*, McGraw-Hill, 2003.
- [Bolman97] Lee G. Bolman and Terrence E. Deal, *Reframing Organizations: Artistry, Choice and Leadership*, John Wiley & Sons, Inc., 1997.
- [Collins02] James C. Collins and Jerry I. Porras, *Built to Last: Successful Habits of Visionary Companies*, Harper Collins, 2002.
- [DeMarco82] Tom DeMarco, *Controlling Software Projects*, Yourdon Press, 1982.
- [DeMarco99] Tom DeMarco and Timothy Lister, *Peopleware: Productive Projects and Teams*, Dorset House Publishing Co., 1999.
- [DeMarco01] Tom DeMarco, *Slack: Getting Past Burnout, Busywork, and the Myth of Total Efficiency*, Broadway Books, 2001.
- [Docherty02] Peter Docherty *et al*, *Creating Sustainable Work Systems*, Routledge, 2002.
- [Jones94] Capers Jones, *Assessment and Control of Software Risks*, Prentice Hall, 1994.
- [Leipzig02] Jeremy Leipzig, *Work Issues in Software Engineering*, North Carolina State University, December 6, 2002.
- [Lowe00] Graham S. Lowe, *The Quality of Work*, Oxford University Press, 2000.
- [Maguire94] Steve Maguire, *Debugging the Development Process*, Microsoft Press, 1994.
- [McConnell96], Steve McConnell, *Rapid Development: Taming Wild Software Schedules*, Microsoft Press, 1996.
- [Metzger96] Phil Metzger and John Boddie, *Managing a Programming Project*, Prentice Hall, 1996.
- [Patz97] Deborah S. Patz, *Surviving Production: The Art of Production Management for Film and Television*, Michael Wiese Productions, 1997.

Magazine and Newspaper Articles

- [Apgar98] Mahlon Apgar IV, "The Alternative Workplace," *Harvard Business Review*, May-June 1998.

[Désiront03] André Désiront, "Nous partons de moins en moins en vacances" (We take fewer and fewer vacations), *La Presse*, Montréal, August 20th, 2003.

[Friedman98] Stewart P. Friedman *et al*, "Work and Life," *Harvard Business Review*, November-December 1998.

[Galipeau03.1] Sylvia Galipeau, "Le Burnout à 20 ans" (Burnout at 20), *La Presse*, Montréal, 2003.

[Galipeau03.2] Sylvia Galipeau, "STOP!" *La Presse*, Montréal, October 18th, 2003.

[Grammond03] Stéphanie Grammond, "Bataille judiciaire entre Ubi Soft et son concurrent Electronic Arts" (Court Battle between Ubi Soft and its competitor Electronic Arts), *La Presse*, Montréal, September 24th, 2003.

[Kimmel93] Michael S. Kimmel, "What Do Men Want?" *Harvard Business Review*, November-December 1993.

[Laramée03] François Dominic Laramée, "No-Competes: Bad for Business," *Game Developer*, December 2003.

[Rothman 00], Johanna Rothman, "Taking the Crunch Out of Crunch Time," *Software Development Magazine*, March 2000.

[Tremblay03.1] Jacinthe Tremblay, "L'horaire d'été, un concept qui s'étend" (Summer schedules gaining in popularity), *La Presse*, Montréal, 2003.

[Tremblay03.2] Jacinthe Tremblay, "Les murs tombent!" (The Walls are Coming Down!), *La Presse*, Montréal, 2003.

Online Documents

[IGDA-IP03] IGDA Intellectual Property Rights Committee, Toby Saulnier and Tom Barbalet co-chairs, *Intellectual Property Rights and the Video Game Industry (A White Paper)*, October 10, 2003, available at: http://www.igda.org/biz/ipr_paper.php

[IGDA-BIZ03] IGDA Attorneys' Panel of the Business Committee, James I. Charne, chair, *IGDA Contract Walk-Through – Second Release*, September 25, 2003, available at: http://www.igda.org/biz/contract_walkthrough.php

[Kreimeier 03] Bernd Kreimeier, *Game Design Methods: A 2003 Survey*, http://www.gamasutra.com/features/20030303/kreimeier_01.shtml

[Miyamoto99] Shigeru Miyamoto, *GDC 1999 Keynote Address (24m27s)*, available on www.gamasutra.com

APPENDIX B - INTERVIEW TRANSCRIPTS

Preamble

In addition to the Quality of Life Survey that we ran through the IGDA web site, we have interviewed several current and past game developers to know what they thought about the industry, their standing in it, and in the case of those who chose to leave the reasons behind their decision.

This section contains transcripts of these interviews.

Interview #1: CTO of a small studio (France)

What is your background?

I have worked as Lead Architect providing core technology & tools for more than 20 projects, and I'm currently working as CTO for a new studio I co-founded.

How many of your projects have had a crunch period?

Most of them ended with a 2 or 3 month crunch period (the record being almost 1 year of crunch). Crunch time usually consisted in 1 to 4 hours of daily overtime, with weekends added at the end of the crunch period.

Do you know anyone that has left the industry after being in for 2 or more years?

Yes, and I know of several people seriously thinking about it.

Why did they leave?

Mostly because they were fired, and did not manage to find an appropriate job in the French industry.

Why were they fired?

Most of them because their companies cancelled projects or closed their doors.

What difference does experience make in each of the roles?

I usually try to get the more experienced people for a job. Experience makes a real difference in terms of estimation and productivity.

What is the process for recruiting someone to a game production?

The recruiting process is fairly standard: ads, then interview. Then I usually try to get the more from the 3 or 6 month test period during which it's really easy to get rid of an employee.

Do you mean you try to get the new people to work hard during that time to get the most productivity out of them while they are still highly motivated?

It's due to the French legislation: when you hire someone, it's easier to fire him during the first 3 to 6 months. So you have to accurately evaluate the performance of your new employee during this period of time.

Have you ever seen someone leave a game production midstream? Why did they leave?

Yes, it happens. They usually leave to get a better job. But nowadays, everybody sticks to his position because of the lack of interesting job positions.

What effect does it have on a production to replace a member in the middle of a project?

It depends on the role of the person, but the impact is usually high.

Have you seen a team member fired from a game production mid-project? Why was he fired?

It is very rare.

Have you seen a project canceled entirely, mid-project?

It happened several times during my work at my last company. This is why I decided to start my own company: we're going to try to handle the project with enough professionalism to get rid of the issues you mentioned in your questions.

Interview #2: Programmer (Orlando, Florida)

What is your background?

I am a Programmer with a Masters degree in Computer Science. I have 6 years of experience in the game industry with an addition 4 years of simulation related software engineering.

How many game projects have you been involved with?

I have worked on a total of four games on PS1, PS2 and, PC.

Do you know anyone that has left the industry after being in for 2 or more years?

Yes.

How many of your projects have had a crunch period?

All of them!

How long was the crunch?

Oh Geez. The longest crunch was an eight-month period that pretty well comprised the entirety of my involvement on the project.

How would it look if you didn't work that many hours because you had a kid, or a sick relative?

At some companies, it would have been a major problem and someone who left early frequently would have been let go. Most companies seem willing to work with someone who is ill or had other concerns though. In other smaller companies, there was a great deal of flexibility. It seemed to be directly related to the individual's standing within the company.

How did you feel at the end of the crunch about starting your next project?

Burnt out and not ready to start the next game. I was unmotivated and disgruntled. I had a poor outlook on life and was frequently negative about things. In general, I felt really poorly.

What is the average experience level of the people you have worked with?

The average experience level was usually 2+ years of experience for the lead programmer, but most others had around a year of experience.

What sort of difference does experience play?

It makes a huge difference. For example, with a more experienced person, you will know that the work will be completed and require limited maintenance. Newer people require a lot of coaching because they sometimes don't understand the code base and get stuck. Some of the issues were related to just knowing how to work through a problem. More experienced people know how to find resources and work through problems a lot better.

How likely is the estimate of an experienced contributor to be right?

I think they are more likely to be closer. More experience yields more knowledge. People who have less experience estimate a lot less time to complete work because they don't realize the complexities and nuances of software development.

What happens when an estimate is wrong?

You work more. Typically, you have to make up the time. Sometimes deadlines could be extended, but in general, if you were behind schedule, you had to keep working until you got back on schedule. The scope of the game was not altered at all to reflect a slip in the schedule.

Have you ever seen someone leave a game production midstream? If so, how often?

Oh yeah, definitely. It happened at least one time per project.

What effect does it have on a production to replace a member in the middle of a project?

Usually the person who leaves is unhappy. That is not surprisingly the attitude of the whole team. It affects the chemistry of the team. The "feeling out" process that normally occurs is complicated because it happens in a pressured situation.

Have you seen a team member fired from a game production mid-project?

Yes.

Have you seen a project canceled entirely, mid-project?

Yes.

When one project completes, how do you get your next one?

I was always just assigned to the next team or whatever the company had laid out for me. They handled the reallocation of the programmers.

Do you feel comfortable that you know where your next gig is coming from? How do you feel about that?

At one company I definitely was concerned about where the next project was coming from. It made work feel unstable and worried me.

Is this something you feel you would like to do for the next 10 years?

No. I enjoyed the work in the game industry, but I didn't enjoy the hassle and stress.

Some companies intentionally scheduled things so that people worked more hours. That problem seemed like it could be fixed by hiring some more people. This is especially true when working on evergreen project that wouldn't have the bottom line cut by adding one more developer.

There was a lack of pre-production and scheduling. People didn't have a solid design going into production. A lot of redesign happened mid stream that caused delays in the schedule. Inevitably, the project would be late. The projects were constantly scrapping everything and restarting from scratch in the middle. Sometimes this was due to publisher influence, but either way they had a big rush to get the game finished. Things like that make it hard to want to work in this industry. Why should I put the effort into it? It's not like I am working extra to put in extra features, I am working because someone is making poor managerial decisions that cause me to miss out on time with family. That's why my answer is no.

Any parting comments?

People think that they can bring in a programmer for a low salary because they want to get experience working on games and just work them to death. I don't agree with that process. People have to maintain profitability, but you can't pay people such a low wage and expect them to be happy and productive. Some companies are clearing houses to bring in cheap programmers. Some students who are just graduating will work for almost nothing. It's not fair to the students or the more experienced programmers. Salaries are generally much lower in the game industry than in other industries. When the owners all have nice houses and expensive sports cars and the engineers are making lower wages, something just doesn't work.

From a quality of life perspective, they are taking advantage of people. Some people took the stance that if you didn't want to work the crazy hours, they would show you the door. The threat of losing a "cool" job was really held over everyone's head and used to play against people to get them to work more or whatever they wanted you to do.

These are quality of life issues, but most of the problems relate back to poorly trained and untalented managers. Give employees better hours and pay and you will see better productivity and better end products.

Interview #3: Programmer (Canada)

What is your background?

I'm a game programmer with 4 years of experience in the industry. I spent 1.5 years at my first company and the rest at my current one. I started in game development with a coop [internship]. I had wanted to get into game production for a long time when I started. I grew up playing Atari and reading game magazines. 2nd year of university I was working at a non-game company on a 3d engine and I decided to take my skills and try to get into games.

How many of your projects have had a crunch period?

All of them have had a crunch time. At my current company each person goes through a crunch about once per year. Otherwise it's not too bad...fairly 9 to 5.

How long was the crunch?

When we are getting ready to ship 11 SKUs (one game, different platforms and languages) we work 8:30a to 12a for 3 weeks, no weekends. Testers work like dogs even without crunch—12 hours a day 6 or 7 days a week.

How would it look if you didn't work that many hours because you had a kid, or a sick relative?

I've never had a schedule imposed on me. Even the nights that we're there late, we're just there in case something goes wrong.

Do you think long hours and exhaustion play any role in people leaving the game industry for other lines of work?

No doubt. It can easily cause a burnout. I had 136 hours of overtime a year ago and I was about to flip. Still feel tired from 2 months ago. I need a solid week of vacation. This holiday season is going to do me some good. I can totally understand that people would leave the industry. I've seen it where I work. People moved from crunch to crunch a couple times and then they leave the company. Some leave the industry altogether. Others have just downgraded from console to cell phone games that has a saner schedule. GBA team pumps a game out ever 6 months. They never seem overly stressed. Kinda weird how all the PS@ and GameCube projects are crazy busy but these guys are not there all that late. GBA has the content already tried and true whereas the consoles have to innovate the game and push the hardware.

What impact does turnover have upon a game production?

At my current company there has been a pretty high turnover rate even considering we are 150 people. 90% of the old staff with over 10 years of experience is gone. 3 are left. 9 or 10 left to pursue other things for a number of reasons. Some felt that there weren't up with the technology and they didn't want to retrain, so they decided to leave the industry. A lot of them had trouble finding work because their skills are outdated. 1 or 2 went into construction. Don't want to see a computer again.

Things have gotten better in the last 4-5 years. Crazy better. I hear horror stories from 5-10 years ago when there used to be continual, never ending crunch mode. When I first started I heard on internet that there would be long hours. When I started working there were the odd long hours, but one can stay up with it.

I choose to spend a lot of time at work, and that can mess with personal life. A lot of times I'll come home from work and play a game. It can be hard for a girlfriend to deal with (my last 3 girlfriends have blamed the game industry as the reason our relationship didn't work out). A lot of the havoc the game industry wrecked on my personal life happened while I was trying to get into the industry. I needed to make a lot of demos and they took a lot of time. The hardest part was getting in.

What impact to experienced developers have vs. inexperienced developers?

The experienced people could help us avoid repeating a mistake they made 6 years ago. We make them over and over. Hackers can't cut it anymore because they shoot themselves in the foot. You cannot afford to just code on the fly—the complexity is far too great to code and fix. So for coders, experience may not be as important as a solid education. But losing experienced

management yes, it's a big loss. It would be good to have someone who knows how things have progressed. The burnouts are programmers. Artists tend to stick it out--their tools evolve but not as fast.

How stable do you consider your job? Are most jobs in the industry? How does that affect your work?

I feel very secure in my job. No problem whatsoever. If I felt insecure I'd leave and go somewhere else. Everyone at my company is assured their job. If anything, there is a demand for new people. Projects require more and more people on tighter schedules. Right now we lost 2 people from tech group and now we need 2 people on top of 5-6 positions that are already open. It's a hiring blitz.

We always have 3 active projects and 2 prototypes in the pipe. There are always 3 teams on top of engine/tech teams. Tech group is a middleware provider to the production teams. Works great. We build all the tools and the engine--works a lot better than having one per project. If you are in the same company that can be a big waste. Some companies are like that--per project, even though they have a lot of the same teams. You have a bunch of games, all developed in the same studio, yet they used completely different tech. A bit of a waste. But hey, it's their money they're wasting.

Any parting comments?

I'm quite happy. I'm fairly simple. I'm independent. Get up, go to work, go play squash at lunch, come home at 5p, go out or play a game. Social club at work like happy hours, free beer at company on Friday at the local bistro. Excellent. Whenever we stay late the producers will get some beer after 10p. Even when we work long hours, it's a good atmosphere. Everyone is friendly and a good mood. You have to perpetuate your own happiness. You have to keep it going.

A lot of the game companies in Britain work 6 days a week. And producers work insane hours--we are taking bets when the one of our producer is going to drop dead. You could watch him degrade from Christmas to fall. He started happy and progressed to depressed. He took 3 weeks off and he's fresh. People get paid peanuts in Montreal. I make a decent salary, but that's because I started in Ontario and I could negotiate based on my previous salary. A lot of the guys I work with don't make that much. Games is a big salary hit... You make a lot less. I wouldn't be making games if I were in it for the money.

Interview #4: Programmer (Brazil)

What is your background?

I'm a game programmer. I've made some prototypes and demos on the past, until I was hired to make a game about a year ago.

How many game productions have you been involved in?

2 released PC games. I was also involved the development of a prototype for the XBox in the last couple of months.

Do you know anyone that has left the industry after being in for 2 or more years?

I don't know yet. My current company is a new company, all my co-workers joined the company with me or after me, and I've been there for only 2 years. Until now, everyone is really involved and do it because it's a passion.

How many of your projects have had a crunch period?

The only one that we did on time, without the craziness of working many, many hours to finish everything was the prototype. Maybe because we're learning how to make a schedule, or maybe because it was not a full game (where everything must be perfect). The 2 other projects had tight schedules (9 months), but I think we've made a really good job delivering it right on time, and with a significant improvement on the genre's games standards.

How long was the crunch?

We have milestones that take always 5 weeks. The crunch is always on the last week of the milestone. As our milestones are always scheduled for Monday morning, the weekend is used a lot to test, fix and add new features. Sometimes we worked all day and night, delivering it on the morning Monday and going home to have lunch and sleep on the rest of the day. The most affected area in these situations is the application programming department. On the last days, before the week, I've worked 10, 12 hours everyday, very stressful indeed. But we have our good times also, like pizza at night and a lot of jokes. :)

How would it look if you didn't work that many hours because you had a kid, or a sick relative?

That's a very good question, makes me think about it. It's a bad situation, because sometimes all the company's work can be affected by something like that. And we had in the past similar situations, but I think that it's not so bad because we always have someone else who can help out. But it's a pressure, without doubt. I feel myself pushed to work really hard, but I think we manage the situation "the best we can", or at least close to that.

How did you feel at the end of the crunch about starting your next project?

When one thing like that happens, I feel relief, a lot... Something like "well, I did my best, everything I could... I'm almost a zombie, but I did it, and it was for a good cause." The first thing is to rest some time, put things back on order, and the natural next step is to think what went wrong and right, and you start to want to make it better, start all over again, or make something completely different. It's nice to work on what we like.

What is the average experience level of the people you work with on a production?

When I started in the industry, in my current company, I had some co-workers that had 1, 2, even 3 years in the industry, but without a released product (except one, but it was a simple, small 2D game). The company spent the last years creating the technology to create games and some prototypes that never came to life. Other than that, all my co-workers were starting, and right now, all the recruited new people have no previous knowledge of the industry, we have to teach them what we know. We have few people in Brazil that have the knowledge, because we have few and recent companies. We're just beginning what in other countries is 2 decades of history. But I think, at least in my current company, that we're making a very good job until now, and growing ;)

What difference does experience make in each of the roles? What difference does 2 years vs. 8 years make for a developer, for example?

Right now I see that 5 years of experience vs. 2 makes a difference indeed... Don't know anyone with 8 years of experience, but from what I know, I can expect that it's almost the same case. The experienced developer knows the problems before they happen, he knows the shortcuts and previous situations where something did work and not. It's a plus, but of course we can have a 10 year developer being a bad developer, but the general case is: it makes difference, not so great difference, but it makes. Mainly when you think about team management and so on.

How much more or less likely is the estimate of an experienced contributor to be right?

I can guess 80% right. With 2 or 3 projects [under his belt], the developer can do a good estimate. But it requires organization and practice, of course.

What impact does it have on the project when an estimate is wrong? How is that handled?

Don't know... The idea is never to underestimate the project, so the estimate never can be wrong... But if it happens, the solution is to talk with the publisher, try to push the schedule or start cutting features. There's no other way. Hiding it from the publisher is the worst mistake you can make. Hiring some people can fix the problem, but it can be tragic too, since new people must be trained, which takes time (of experienced developers).

What is the process for recruiting someone to a game production?

I think it's the same for other industries. We always receive resumes, we put something in the newspaper, make interviews, and so on.

Have you ever seen someone leave a game production midstream?

Never saw it, but I think the only way for something like that happens is a disagreement with someone in the company or a better position on another company.

What effect does it have on a production to replace a member in the middle of a project?

It's difficult, but we try to manage it, since we try to not be over-specialized on one specific area. It never had happened with us, but we're trying to get a better position on a situation like that, making documentation for a new employee and improving our tools.

Have you seen a team member fired from a game production mid-project?

No.

Have you seen a project canceled entirely, mid-project?

No.

When one project completes, how do you get your next one?

We make research and prototypes, so we never stop. We have one person that is always searching for new projects, and one agent that is in touch with several companies too. When we finish a prototype, we visit publishers to show what we've got and try to get a contract for that project or another. Also, we go to conferences like E3 and GDC.

Do you feel comfortable that you know where your next gig is coming from? How do you feel about that? Is this something you feel you would like to do for the next 10 years?

I feel very comfortable with what I do, because I love it. It was my childhood dream. Though it was very difficult to start in this industry, I'm very happy to be able to be a part of it. And I'm in the beginning [of my career], learning a lot every day. Right now I have several new challenges, and it's very nice to be here. I can feel that the company is growing and we have some nice possibilities for the future. Feels great :)

Interview #5: QA tester for a large publisher (United States)

What is your background?

I have been a QA tester for 2 years with the same publisher and then I left the industry.

Why did you get out?

Too much work, not enough reward. You have to love games so much that you are willing to completely make that your life. I think there needs to be some balance and I didn't have any balance while I was there. It was all about production and getting the game out.

I worked really long hours. When a project starts you are almost working regular hours, around 40 hours a week. But as in wraps up it gets the point where in the end your working like 100 hours a week sometimes. Sleeping in back of your desk or underneath your desk. Get up, take a shower, hit the gym, get back in your seat and start working again. Even if you do go home you are there for only a few hours and then you are back again. It's tough--you're working serious hours.

How many of your projects had crunch periods?

They all crunch in the end. I've worked mostly on new games. Patches were easy, but any time there is a new game there is a lot of work to be done. I've worked mostly on PC games. PS2 games are a little easier--there's less to find.

How much of the time were you in crunch mode?

3-4 months of the development timeline has QA working. You get as much time off between cycles as you want. Just come back and get on another project when you are ready to come back, but it's still not enough. You are not getting paid and you don't have any friends anyway to hang out with because you disappear for 4 months at a time.

The hours just keeps building. 40 to 50. 50 to 55. 55 to 60. The last few weeks I was hitting 80 to 100 hour weeks.

Why did you work such long hours? Why not just go home after 40?

I think that they would actually get more out of you if you didn't work that much, but you know, they need you there. Sometimes you aren't doing anything; you are just waiting for the new build to come in. The programmers are working frantically to fix what you sent them and they send it back to you. You get to find out what broke when they fixed it. See if they fixed. Waiting for couriers. Everyone is waiting for everyone else.

If you said: "I have to go home, my wife is expecting me, see you later" what would happen?

You couldn't do that. You could maybe do that once. But you would feel bad. It's such a team environment. It turns into group hypnosis where everyone is working hard together. Not just the testers but the producers and the programmers. Everyone is crunching to get it out. If you don't do it, you will get a bad review. That's just the job.

How did you feel at the end of the crunch?

Every time I say I'm not going to do another one. And then I come back for another one. I did that with 4 games. It's exciting. But especially because I was allowed to choose my projects. When you see the first build of a game you are really interested in, and especially if you have a little say about how things are going to be, it's really cool, it's exciting. But then it just gets really repetitive and it gets to be too much.

How does exhaustion contribute to people's decision to leave the industry?

I think people are mentally and physically exhausted after a tough project. They need to either hire more people so that they can have people focused 40-50 hours a week really working hard rather than all the downtime that you are there waiting for a build is just a waste. I think a lot of people become alcoholics, they lose their families, they miss their kids because they are just at work all the time. I lost my girlfriend of 7 years because I worked a Christmas and that was the straw that broke the camel's back. It's exhausting and it's hard on everybody. You have to choose between being in the game industry and having any kind of life outside of work. When you are in film, at least on the set you have as social dynamic where even though there is pressure to finish everyone is moving around, it is more social. But at least where I've worked it's more isolated. They do things Vegas style--they never turn the lights on. It's dark all the time--you never know what time it is. If you need to write something you flip an a little reading light and write it. It's very [hard] staring at the screen for hours and hours and hours. You might turn to someone to ask a question and you take breaks to play a group game of Quake or something, but generally it's very isolating. You work alone most of the time. It's hard.

What is the average experience of the people you worked with?

0 years. Entry level people. My supes have been there 3-5 years. I didn't know anyone other than the producers and designs. Everyone else gets turnover. I've worked with testers that have been there as long as 3 years. They would be the lead testers.

What difference does experience make?

By the time you have done one project in test, you are a pro. Every game is so different, you have to be creative. If you are a good gamer, you can be a tester. I'm not sure that experience affects test too much. But for higher levels, you can't be a lead or an AP without a lot of experience.

Have you seen people leave midstream?

Yes. Too much work. It became more work for everyone else. Then you bring someone new in who isn't familiar with the game and the bugs and it would slow things down.

Ever seen a project cancelled?

No. By the time it gets to test it usually gets done.

Did you feel comfortable that you would have work after you completed a project considering you worked as a contractor?

By the time it's over I could just talk to the testing manager and if you are good you can go. I felt comfortable that I knew where my next gig was coming from. It's a never ending line of work.

Anything to add?

When I left my company they were aware that the testers just get tortured. It's a really hard, [lousy] job, you don't make any money, and to the rest of the company you are like bugs. To everyone else you are lowest of the low. No respect. They were toying with having more shifts with testing. And they were going to add cots to a room so people don't have to sleep on the floor. They are aware of what it's like--I don't think they expect people to stay with testing very long. The ones that do go on to become a lead or a producer. I don't know if being a producer is that much better--the people on the production side were just as worked and frazzled, but at least they are making a little more money. They can drive a little nicer car, come home to a decent house. I know they are definitely aware. I don't think they care, they are making too much money.

What is the solution?

I think if you have a really good producer who...there's nothing you can do really. For the testers they could have more shifts. Everyone would need to get together sometime to communicate, but you can see what's going on in the database. You can read about [it]. Just having multiple shifts would be good for everyone. As far as the poor programmers, I don't know. You can only have so many programmers, I think. Those guys never went home. We would be bitching about how many hours we were there, but those guys from the beginning of the project to the end just were always working, so I don't know. It seems lame to just say hire more people. Or instead of releasing at a particular time just increase the timeline--but it doesn't work that way, you have to get the game out in 15 months. I don't know what the solution is. I hope you guys figure out some good ones.

APPENDIX C – QUESTIONNAIRE AND RAW DATA

Preamble

The IGDA Quality of Life Survey was held in January and February of 2004, on the IGDA's web site. It was advertised through the IGDA's newsletter and web site, and also through messages posted on several popular development web sites, including Gamasutra.com, Gignews.com and Gamedev.net

All of the questions were multiple-choice. We received a total of 994 replies.

Questionnaire

Demographic Questions

How long have you been working in the game industry?

- Still a student or trying to get a first paying job.
- Less than 2 years.
- 2-5 years.
- 5-8 years.
- 8-12 years.
- Over 12 years.

How did you get your first game development job?

- Started my own company.
- Through a friend.
- Through a job fair.
- Through a job ad.
- Through an internship or work-study program.
- Other.

Which of the following best characterizes your current employment situation?

- Full-time, permanent employee of a game development studio.
- Part-time employee of a game development studio.
- Contractor working on a single project.
- Freelancer working on multiple projects.
- Independent developer working on games in my spare time.
- Studio manager or owner.
- Unemployed and looking for work.
- Student.

Which of the following best characterizes the game company you are currently working for?

- Large studio usually working on 4 or more projects at the same time.

- Mid-sized studio usually working on 2-3 projects at the same time.
- Small studio with a single project team.
- Independent team working part-time.
- I am a freelancer or contractor.
- Other: _____

What is the average experience level of developers at your company?

- Less than a year.
- One to two years.
- Two to five years.
- Five to ten years.
- Over ten years.

What is the average experience level of project leads (producer, lead programmer, lead artist, lead designer) at your company?

- Less than a year.
- One to two years.
- Two to five years.
- Five to ten years.
- Over ten years.

What is your age?

- 19 or younger
- 20-24
- 25-29
- 30-34
- 35-39
- 40-49
- 50 or older

What is your gender?

- Male
- Female

What is your marital status?

- Single, unattached.
- Single, in a serious relationship
- Married or living with a partner

Do you have any children?

- Yes
- No

General Satisfaction Questions

On a scale of 0 to 10, with 0 being the lowest and 10 being the highest, how satisfied are you with your game development career in general?

If you were to ask your current (or most recent) spouse, boyfriend or girlfriend, what would they say about your game development career? Check all that apply.

- You work too much and don't spend enough time with me and/or the kids.
- You are always stressed out.
- You don't make enough money.
- You seem so happy; it's great!
- When are you going to get a real job?
- I don't like the content of the games you work on.
- I wish I had a job like that.

How long do you plan on staying involved in the game industry?

- I will probably look for a job in another field within 2 years.
- I will probably stay in games for another 2-5 years, then leave.
- I will probably stay in games for another 5-10 years, then leave.
- I will probably stay in games for my entire career.

If you could change one thing about your game development career, what would it be?

- I would work on more interesting projects.
- I would earn more money.
- I would take on greater responsibilities.
- I would work shorter hours.
- I would get more job stability.
- I would not change anything.

In your opinion, what is the leading cause of stress for you and your coworkers?

- Tight ship dates.
- Bad relationships between management and developers.
- Bad relationships between coworkers.
- Bad relationships between the company and publishers.
- Company is in financial trouble.
- We don't know where the next project will be coming from.
- Everything is fine.
- Other: _____

Job Stability Questions

Which of the following best describes the transition between projects at your current company?

- We usually have new work lined up before a project is completed.

- We usually have significant down time between projects, but we keep our employees on the payroll during these periods.
- We hire on a project basis and lay off staffers once the project is completed.

Have you ever been laid off from a game development job? If so, why?

- No, never.
- Yes, when my project was cancelled in midstream but the company stayed in business.
- Yes, when the company went out of business or my local studio was closed.
- Yes, at the end of a project that shipped.

Have you ever quit on a project in midstream? If so, why?

- No, never.
- Yes, because I felt that the project was going to fail.
- Yes, because of conflicts with management or coworkers.
- Yes, because I was too exhausted to continue.
- Yes, because I found a better job elsewhere.
- Other: _____

During your game development career, what is the longest period of time you have ever spent with the same company?

- Less than a year.
- Between one and two years.
- Between two and five years.
- More than five years.

Work Load Questions

How many hours do you work in a regular week?

- Fewer than 35 hours.
- 35-45 hours.
- 46-55 hours.
- More than 55 hours.

Is the length of your workweek representative of the average at your studio?

- I tend to work fewer hours than my colleagues.
- I tend to work about the same number of hours as my colleagues.
- I tend to work more hours than my colleagues.

How does your company compensate overtime? Check all that apply.

- We pay overtime at the employee's usual hourly rate.
- We pay overtime at a premium rate.
- We count all hours and compensate overtime with an equal amount of time off.
- We compensate with some time off at the end of a project, but we don't count all hours.

- We pay milestone bonuses.
- We compensate overtime with royalties or profit sharing.
- We do not compensate overtime.
- Other: _____

How often does your company have "crunch times" during which most members of a team work longer hours than usual?

- Never.
- Only during final beta testing.
- Before every milestone.
- Monthly or more.

How long do crunch times typically last at your current company?

- Less than a week.
- Between one and two weeks.
- Between two weeks and a month.
- Between one and two months.
- Over two months.

How many hours do you typically work during a crunch week?

- Fewer than 35.
- 35-45.
- 46-55.
- 55-65.
- 65-80.
- More than 80.

Which of the following assertions best describe your company's policy regarding crunches? Check all that apply.

- Everyone gets involved, no matter what.
- All developers work long hours during crunch, but management doesn't.
- An employee doesn't have to participate if their own assigned work is completed.
- It is possible for an employee to refuse crunch time for personal or family reasons.
- Management sees crunch time as a normal part of doing business in the game industry.
- Management sees crunch time as a necessary evil and tries to minimize its impact on employees.
- Management actively applies a no-crunch policy, like the 40-hour workweek.
- Other: _____

What is your company's usual policy regarding release dates?

- We ship when the game is ready, no matter how long it takes. Quality comes first.

- We are under significant pressure to release for Christmas or at another fixed date, but we can survive if we slip by a couple of months.
- We absolutely, positively must ship by a certain fixed date or we will get killed in the marketplace.

Which of the following best describes the staffing situation at your company?

- We have all the people that we need to make production smooth and painless.
- We could use some more people or special skills in some areas once in a while.
- We often have to work extra hours, learn on the fly and/or improvise because we can't hire people with all of the skills that we need.
- We are chronically understaffed and production is always stressful.

Work Organization Questions

Do you feel that the schedules and staffing requirements that your company prepares in pre-production are:

- Very accurate and lead to easy production cycles.
- Sufficiently accurate and flexible to get by with only a minimal amount of crunch time.
- Reasonable in most cases, but occasionally flawed, leading to tense periods.
- Wishful thinking that will only fit reality if no unforeseen problems arise.
- So optimistic that we know we'll be in crunch from Day 1.

How does your company control changes to the game design during production?

- We never change anything to the game once production begins.
- We have a formal change control policy that minimizes changes.
- We often add features when someone on the team comes up with a good idea or sees something great in a competing product, but we're careful not to impact the schedule too much.
- Feature creep is a big problem for us, and it messes up our schedules big time.

Job Quality Questions

How would you characterize your working environment? Check all that apply.

- Noisy.
- Overcrowded.
- Not enough privacy.
- Effectively promotes teamwork.
- Too corporate.
- Comfortable.
- Computers and/or network need upgrades.
- Other: _____

How would you rate the level of challenge you experience at your current job?

- I am overwhelmed with complexity. The job is too hard.
- I am constantly challenged and I love it.

- My job is usually interesting, with small, manageable amounts of drudgery.
- I have mastered my job and would be ready to take on a new challenge, but I am in no hurry to change.
- I am overqualified for the work I do and I am bored or frustrated most of the time.

Varia

Which of the following best describes your experiences when you first joined the industry? Check all that apply.

- I had a mentor and it helped ease my way into the business a great deal.
- I had a mentor and it helped a little.
- I didn't have a mentor; I made my own way.
- I desperately wanted to develop games; it was the only choice for me.
- Developing games was only one career option among many.
- Getting my first game development job was easy; it took 3 months or less.
- Getting my first game development job was moderately hard; it took 3 to 6 months.
- Getting my first game development job was hard; it took over 6 months of effort.
- During my first year, I often felt I was "paying my dues" with grunt work instead of being challenged to my full potential.
- I felt part of the industry's "big picture" right away.
- At some point during my first year, I considered leaving the industry.
- Other: _____

How do you feel about the controversial content in games like *Grand Theft Auto 3* or *Postal*?

- It doesn't bother me at all.
- The only thing that bothers me about it is the media coverage that makes game developers look bad.
- The content doesn't bother me personally, but it bothers my friends, family or community and that makes me uncomfortable.
- I don't like it and would never work on a project like that.

Which of the following assertions best describe your company's policy regarding credits? Check all that apply.

- I always get the credits that my work deserves.
- If you leave the company before the project is released, you're probably not going to get a credit, no matter how much work you did.
- I feel that my work isn't properly credited.
- There are often people who get credits in games on which they didn't work.
- The credit allocation policy makes no sense to me.
- The credit allocation policy is fair and balanced.
- Management and publisher staff get too much credit compared to developers.

Do you have a clear plan for your career?

- Yes, and my company and my boss support my development actively.
- Yes, but I have to pursue it in a clandestine manner because the company would rather have me stay at my current level.
- No, I'll just see what comes.
- It doesn't matter, because it is hard to make any plans in this industry.

Raw Data

The raw results, split into various categories, are available as a separate set of Excel spreadsheets from the IGDA web site at www.igda.org/qol.