



**IGDA Business Committee**  
**- Best Practices in Quality Assurance/Testing -**

**Prepared by:**  
Eric Rees – freelance  
Laura Fryer – Xbox Advanced Technology Group

April 22, 2003  
[www.igda.org/biz](http://www.igda.org/biz)

## ***The IGDA Business Committee***

The IGDA Business Committee's mandate is to empower the development community with business knowledge and in the process allow developers to make better games.

The goals of the Business Committee are as follows:

- Enable developers to build stronger, more successful companies
- Provide knowledge and business support resources
- Increase the perception of game development as a credible business and raise the profile of game developers as viable companies
- Improve the publisher/developer relationship
- Improve the retailer/developer relationship

Additional information on the IGDA and the IGDA Business Committee can be found at

<http://www.igda.org/biz/>

<http://www.igda.org/committees/business.php>

[http://www.igda.org/committees/business\\_members.php](http://www.igda.org/committees/business_members.php)

## ***The Best Practices Roundtables & Reports***

The Best Practices Roundtables & Reports are one of the 2003 initiatives of the Business Committee of the IGDA. The end goal of these roundtables is to prepare a summary report on each topic for distribution to the game development community via the IGDA web site. In sharing this best-practice knowledge, we hope that developers will thereby be able to improve their human resources, schedule their projects more efficiently, work with their publishers to optimally market and promote their games, improve quality assurance testing on their games, and bring greater financial stability to their companies. The five topics covered in 2003 were:

- Best Practices in Human Resources
- Best Practices in Resource Management/Scheduling
- Best Practices in Promotion/Marketing
- Best Practices in QA/Testing
- Best Practices in Finance

Additional information on the Best Practices Roundtables can be found at:

[http://www.igda.org/biz/best\\_practices.php](http://www.igda.org/biz/best_practices.php)

## Overview

This roundtable on best practices in quality assurance/testing was held in two separate sessions at GDC 2003 (March 6, 2003 and March 8, 2003). The roundtable sessions were established to examine how developers can work with publishers most effectively to test their games (quality assurance, functionality, playability and compatibility).

Quality assurance/testing is the finding and reporting of bugs in video games and software. In the video game industry, QA has become essential for companies to compete for consumers. Testing is often considered one of the most critical components of the development cycle. It is the glue that holds the final product together. It is both time consuming and intensive. And there is nothing worse in the eyes of a consumer than having your favorite video game freeze/lockup after beating the 3<sup>rd</sup> level boss!

These best practices are mostly written from the publisher's perspective and tend to lean more towards the console testing, however all can apply to developers and publishers of console and PC games alike.

The best practices that were discussed are each organized around the following format:

- "Name" of the best practice
- Description of the best practice
- Pros
- Cons
- Other successful alternatives or variations

## Best Practices

### Automated Testing

*Description:* Automated testing is as simple as removing the "human factor" and letting the computer do the thinking (so to speak). This can be done with integrated debug tests, to much more intricate processes. The idea of these tests is to find bugs that are often very challenging or time intensive for human testers to find.

*Pros:* This sort of testing can save many man hours and can be more "efficient" in some cases.

*Cons:* It will cost more to ask a developer to write more lines of code into the game (or an external tool) than it does to pay a tester (in general) and there is always the chance there is a bug in the bug testing program. Reusability is another problem; you may not be able to transfer a testing program from one title (or platform) to another. And of course, there is always the "human factor" of testing that can never truly be replaced.

*Other successful alternatives or variations:* Nothing is infallible. Realistically, a moderate split of human and automated testing can rule out a wider range of possible bugs, rather than relying solely on one or the other. Giving the testers limited access to any automated tools can often help speed up the test cycle.

## **Specialized groups or individuals for testing standards (TRCs)**

*Description:* In all forms of software testing there are certain criteria or standards that must be met and this is especially true for console based games. Whether it is the font of the text or the load times, these are often put in place for a good reason. There are many, many different standards for each of the various platforms and it is often difficult to know them all.

*Pros:* Having a group or individual specialized in the “art” of standards is not only a huge asset but also a rarity. There are so many different standards it would be easier to memorize a dictionary ☺. If there is a set group or individual that is very versed in this, it can lead to less time being spent, or in reducing the chances of having to resubmit games for cert.

*Cons:* Quality assurance is about two things, resources and time. While having a group or individuals set aside to do this sort of testing can save time, it also takes away from resources, leaving other potential testing needs unmet.

*Other successful alternatives or variations:* Because of the complexity of the console standards and depending on the developer’s familiarity with the standards, it can become a cumbersome task. Having small teams of people (who may or may not work as a team) who are familiar a set of the standards or perhaps even one particular console’s set of standards can balance out the necessity for standards and bugs. Also organizing groups or individuals can specialize or be trained for other things, such as genera, sound or AI (test buckets); essentially leading to a ‘QA of the QA’.

## **Involving quality assurance in the development process**

*Description:* Many people get into quality assurance with hopes of it being a stepping stone into other areas of the game industry. In almost all cases they are avid gamers with an eye for detail. They know what they like and what they want in the games they play. Having the development team ask the quality assurance department for creative input can open a number of new doors.

*Pros:* Quality assurance testers are probably the pinnacle of “game players”. You would be hard pressed to find anyone who plays games more (and gets paid for it). Many are students, programmers or artists aspiring for something more. Getting a fresh perspective and untainted opinions of a project can help the developers keep their focus or change it for the better.

*Cons:* While it is important to get quality assurances feedback at the first playable build, it can be easy for a tester to become distracted from the quality assurance process. It isn’t hard for a tester to get caught up in the excitement of development.

*Other successful alternatives or variations:* Allowing only the lead tester(s) or a selected group to place creative input can still add value, without overwhelming the developer.

## **Outsourcing the Quality Assurance department**

*Description:* Quality Assurance is one of the last lines of defense in a product. Without QA testers, products would ship with all sorts of problems and bugs. As more and more developers come up, it is hard for publishers and developers alike to test all the titles in development. While quality assurance is absolutely necessary, it is also an overhead cost that smaller developers often choose to outsource.

*Pros:* Allows the publisher/developer to carry less headcount (and consequently less overhead cost) and not have to hire and release people all the time. This also tends to deal with the seasonal nature of testing (i.e. the down time between projects).

*Cons:* It can cost a developer or publisher more money than bringing the resource in house. Outsourced testers may not be familiar with console or publisher standards, which can tend to take more time.

*Other successful alternatives or variations:* Having a quality assurance of the quality assurance department can lead to higher quality product. By using an internal QA department as well as the developers (or vice-versa) it allows a seasoned staff to do the core testing. Some companies will also do a “co-testing” environment, where they will share their resources as well as the cost.

## **Testing at a particular milestone point**

*Description:* The question “when to test” often comes up. Should a game be tested as soon as it is interactive? Should it be tested when it is at a fully playable state? There is no one right answer for this question, but there are definitely some things to consider:

*Pros:* Testing at the first possible playable build can help the developer find or re-define their focus. It can also help the developer catch bugs early in their code that could otherwise turn up later, and potentially save time. Testing at a more advanced state can give the tester a more defined understanding of what the developer is trying to accomplish and therefore be more proactive in the testing cycle.

*Cons:* Testing too early can overwhelm the developer and lead to problems before they even start. Testing at a later time could often mean missed bugs that have been caught and prevented earlier in the development cycle.

*Other successful alternatives or variations:* Allowing only the lead tester(s) to test at an early stage can help, and keeps the balance of testing and development and the opportunity to develop a test plan for when the entire test team is assembled.

## Writing the Test Plan

*Description:* Not too different from the question of when to be testing is the question of who should be writing the testing plans and when this should start. Should these plans be written by the developer or the publisher? Should a draft be written at the first playable build, or even earlier? Again, there is no one ‘correct’ answer; just factors that may help you decide what is right for your environment:

*Pros:* Having the developer write the test plan can help keep the integrity of the original product and can help ensure that a more thorough test plan is written. Having the publisher write the test plan can help free up developers to work in their respective disciplines. Writing the test plan early in the development stage can also help the developer discover any problems before they appear, while waiting until later in the development cycle can make for a more thorough test plan.

*Cons:* Having the development team write the test plan can overwhelm the developer and prolong the development time. Having the publisher write the test plan may completely miss the mark on what to test for. Writing the test plan early in the development stage can lead to making too vague a test plan, while waiting until later in the developer cycle can make for a test plan that is overly complex.

*Other successful alternatives or variations:* It’s not a bad idea to have the developer start a preliminary draft of a test plan and then pass it off to the publisher to write the final stages of the test plan. Also, handing over the design document can help the publisher write a more thorough test plan. Ideally a test plan should be a living document. The test plan should be started as early as possible and be continually updated/revised as the development cycle moves forward.

## List of participants

(Note: more people attended the sessions than are listed below, but not all filled out the sign-in sheet)

<i>Name</i>	<i>Title</i>	<i>Company</i>
Matthew Kangas	Software Test Engineer	Volt
Ardy Kriaz	Dir. Of Production Support	Incredible Technologies
Barry Caudill	Quality Assurance Manager	Firaxis Games
Benjamin Smith	Localization Project Supervisor	Electronic Arts
Chris Wilsion	Quality Assurance	Vivendi Universal Games
Craig Parrotte	Technical Support Supervisor	Vivendi Universal Games
Dave Losapio	Quality Control manager	Red Storm Entertainment
Hugh Falk	Executive Producer	Midway
John Walsenke	Quality Assurance	Immersion
Josh Druckman	CEO/Lead Designer	Dark Matter Entertainment
Kathy Schoback	Director of External Development	Sega
Kevin Holme	Software Test Engineer	Microsoft

Kevin O'Garman	Designer	Cosmic Orgins
Prodipto Roy	Software Test Engineer	Microsoft
Randy Lynch	Production Certification Manager	Vivendi Universal Games
Sean Jenkin	Lead Software Designer Evaluator	Microsoft
Sheri Pocolujko	Production Support Administrator	Incredible Technologies
Tim Knappebej	CTO	20/20 Labs
Tony Bourne	Marketing Director	Wise Monkey
Aaron Leiby	Developer	Legend Entertainment
Armen Casarjian	Game Designer	Realize Games
Benjamin Smith	Localization Project Supervisor	Electronic Arts
Charles Xavier	Director of Localization	Polarity Post
Chuck McFadden	Senior QA Lead	Lucas Arts
Dan Urquhart	Game Designer	Spielo
Daniel Sussman	Producer	Harmonix Music Systems
Dave Losapio	Quality Control Manager	Red Storm Entertainment
Heather Sowards	Audio Director	Microsoft
Hugh Grimley	Lead QA	IO Interactive
John Doyle	Director of Techology	Wizards of the Coast
John Harey	QA Lead/Tuner	Pseudo Interactive
Juan Santiago	Producer	
Keith Matejka	Quality Assurance Supervisor	Konami
Kristen Kennedy	Quality Assurance Manager	Rockstar Vienna
Larry Mellon	Engineer Lead	EA/Maxis
Michael Kelbaugh	Director of Product Testing	Nintendo
Mike Baldwin	Developer	Legend Entertainment
Mike Fox	Senior Programmer	Legend Entertainment
Prodipto Roy	Software Test Engineer	Microsoft
Reter Ehardt	Quality Assurance Manager	Rockstar Vienna
Ross Kudwill	CTO	
Steve Rossi	Developer	Cisco
Steven Clayton	Producer	
Stuart Hay	Director of Quality Assurance	Vivendi Universal
Tim Knappenberger	CTO	20/20 Labs
Matthew Kangas	Software Test Engineer	Volt
Ardy Kriaz	Dir. Of Production Support	Incredible Technologies

## **About the IGDA**

The International Game Developers Association is the independent, non-profit association established by game developers to foster the creation of a worldwide game development community. The IGDA's mission is to build a community of game developers that leverages the expertise of our members for the betterment of the industry and the development of the art form.

Visit [www.igda.org](http://www.igda.org) for more information.