
“Fit and Finish” Using a Bug Tracking System – Challenges and Recommendations

Yossi Avnon

Advertiser and Publisher Solutions Group
Microsoft, Inc.
1 Microsoft Way.
Redmond, WA 98052 USA
yossi.avnon@microsoft.com

Scott L. Boggan

Advertiser and Publisher Solutions Group
Microsoft, Inc.
1 Microsoft Way.
Redmond, WA 98052 USA
scott.boggan@microsoft.com

Abstract

This article shares practical lessons for using a bug management tool to manage user interface fit and finish process for a software product. It describes common challenges and provides recommendations for processes that will lead to enhanced product quality.

Keywords

Fit and finish, bug management, UI processes.

ACM Classification Keywords

H5.m. Information interfaces and presentation; H.5.2 User Interfaces (*Screen design, Standardization, Style guides*); D.2.5 Testing and Debugging

General Terms

See list of the limited ACM 16 terms in the instructions.

Introduction

In a perfect world we wouldn't need this article. The fit and finish of software products would always be at the highest level as developers and testers would always follow precise and detailed specification documents and UI (user interface) style guidelines. These documents would clearly describe all the necessary requirements and design guidelines would be very easy to retrieve. The different teams working on the product - whether

Copyright is held by the author/owner(s).
CHI 2010, April 10–15, 2010, Atlanta, Georgia, USA.
ACM 978-1-60558-930-5/10/04.

in different geographical locations, countries, or cultures - would always be in perfect synchronization. In such a world there would be little need to iterate on resolving UI bugs and the result would be high quality, bug-free products. As we all know this isn't reality yet.

This case study relates the experience of a UX (User Experience) team in their attempt to improve the fit and finish of a software product in the last cycle of development using a standard bug tracking tool. The bug management system is primarily designed for functional bugs. However, since functional bugs and fit and finish bugs are different, using a bug management tool for this task introduces multiple challenges:

- Fit and finish issues are often either visual or interaction-dependent.
- Functional bugs are binary: a certain function works or doesn't.
- Many fit and finish issues are about fine details

This article describes some common issues and provides recommendations that can be useful for other teams trying to achieve a high level of fit and finish. Adopting even some of the recommendations listed below, can potentially make the fit and finish process much more efficient and hopefully a bit more fun.

In the past, many UX teams went through the fit and finish process without a dedicated software tool to document and manage the UI bugs. This fit and finish process was often managed by having a UI design specialist demonstrating UI bugs directly to developers and using emails and other informal methods, without the aid of a formal tracking tool. It may feel as if such a

process is more efficient, in fact an ad hoc process can work well in a small company with fewer people and less structured processes. Some aspects of this simple engagement model are worth preserving, even in more rigorous and methodical environments.

Our product is developed by many different teams, several of whom are located remotely. These teams are using different technologies with different UI features and capabilities. The UI developers had different skill sets, varying levels of attention to detail. When the code from these various teams was finally integrated, portions of the UI had a slightly different level of conformance to the interface guidelines and specs. Some of the inconsistencies we discovered involved detailed interactions, padding and alignment for UI elements and visual attributes of UI and typographic elements. All of this created a somewhat ragged and unpredictable experience, which led to several design reviews and bug tracking. Like any other bug, the UX fit and finish bugs needed to go through a triage process with various stakeholders in order to evaluate their importance and their cost. All the stakeholders had visibility to the process through the bug management tool, and several iterations were often needed.

The fit and finish process began a few months before the release of the product, and design review sessions—either face to face, or over LiveMeeting—were scheduled with each of the various teams. After each session the UX team logged the bugs in the bug management tool. The product used was Microsoft Team Foundation Server (TFS), the lessons we've learned will apply to any standard bug management system. The challenges that the UX team faced and the recommendations are described below.

Log one bug for each defect

When filing a fit and finish bug, try to log a single bug for each discrete item. This will enable more flexible tracking and prioritization of individual bugs. The tradeoff, of course, is that logging and managing each bug separately requires more work upfront, but the benefits will worth it.

Thoroughly document each bug

When logging several bugs, it's hard to ensure that each bug includes all of the details, accurately describes the steps needed to reproduce the issue, and includes screenshots of the intended design and as-built design. Bugs that didn't have the full details almost universally failed to be properly fixed during the first iteration, or were resubmitted to the UX team for further details. The developers that were assigned to fix these bugs were not necessarily the ones attending the review sessions, and any bug without full details are too easily subject to subjective interpretation and misunderstandings.

Include the visuals!

Many fit and finish bugs try to address a variety of seemingly minor visual issues (colors, fonts, alignment, spacing, and so forth). Without an image of the current implementation as well as the intended design, developers are unable to determine exactly what is wrong and how to correct the issue. Functional bugs can oftentimes be more easily communicated, but visuals are critical for fit and finish bugs.

Refer to UI guidelines

For the purpose of consistent UI, a reference system that has detailed guidelines is very important. One of the challenges with guidelines is structuring the

information so that you can efficiently refer to it in a bug. In real life situations a single fit and finish item can refer to multiple guidelines. Don't expect that the developer assigned to a bug will read through long documents to find the guideline that you are referring to. At the same time, it's inefficient to copy an entire guideline in each bug (can cause versioning issues).

Fix broken communication

When you have reported a fit and finish bug that was fixed incorrectly, there is a good chance that you haven't explained yourself correctly. Try to understand the source of the miscommunication and address this issue in a way that is relevant for the working style of your organization and the personalities of the team members with whom you are working.

Pick up the phone or meet in person

The personal touch is often more efficient in getting things done. If you see that a bug is not being properly resolved or there are communication issues, the best approach is to pick up the phone or meet with your team members to discuss the issue face-to-face.

Avoid asking for fixes outside the bug tool

If you have discussed an item and it's not captured in your bug management system, most likely it will not be fixed the way you want it. Once you are committed to your bug system, make sure all of the items you want to be fixed and validated are there.

Detail complex interactions

The fit and finish is not just of the static visual, the way the system behaves and responds for mouse hover, clicks, selections, and execution of commands is critical in creating an engaging and professional experience. In

many cases—especially when working with rich Internet applications a static screenshot will not clarify the intended design. Reporting a bug on an interaction is more difficult than a static visual so be prepared to pinpoint the design using a prototype or even a simple click-through.

Nail the basics

Placing and aligning controls on pages in a complex product with multiple frameworks and different technologies is tedious, time-consuming and frustrating. Carefully prioritize the issues you report, beginning with persistent UI elements that appear throughout the application (such as the alignment of header items). Users may not notice minor layout problems in pages that are infrequently used, but nail the basics.

Don't forget about embedded UI text

Aside from typographic mistakes, problems with text and wording are often not perceived as fit and finish items. Seeming casual updates to text might have downstream impacts that create problems. Carefully consider all the dependencies (for example, help files, tutorials, and web content) before modifying UI text and carefully coordinate these changes with the group responsible for the text. Don't let the fact that text is not code deceive you!

Be strategic!

Covering all areas in a single release can be too much: you might not have the necessary resources, or may still be defining design guidelines. If necessary, take an agile approach and focus in the most important areas, and get to the rest of the product in the next release. Focusing your efforts can often force you to refine your

design and develop more expertise in the core scenarios. Define a strategy to follow across releases.

Communicate the UX team priority

Other teams are sometimes unable to accurately assess the impact a particular fit and finish issue has upon the user experience. Make sure that the UX team is in sync on the priority of fit and finish bugs, and communicate the team's priority during bug triage.

Send ongoing progress reports

Fixing many small fit and finish things can end up requiring a large investment of time. Be sure to keep the various stakeholders informed about the overall progress. This is an opportunity to present a bigger picture of the importance of fit and finish on the overall product quality.

Conclusion

Current bug management systems are general purpose systems optimized for logging and managing functional bugs throughout the software development lifecycle. These form-based systems require most visual design bugs to be included as attachments.

A bug system that enables annotations and other direct interactions with a working product would be a big help. Imagine a bug system that included a visual component that would allow users to quickly navigate to visual problems, view and modify annotations right on the screen, and cycle through a sequence of screens or other interactive elements. This would make the process of tracking fit and finish issues more efficient, more engaging, and lead to a better user experience. What UX practitioner could ask for more?