



Usability Testing Plan

GNOME Shell Phase I

Version 1.0
10 February 2010

Máirín Duffy <duffy@gnome.org>



Test Plan at a Glance

Version History.....	3
Overview.....	3
About GNOME Shell.....	3
Test Objectives.....	3
Methodology.....	3
Potential Approaches.....	3
Example Task Creation.....	4
Hypotheses.....	5
Window Management.....	5
Application Launching.....	5
Notification Handling.....	6
User Focus / Immersion.....	6
Documents.....	6
Task Switching.....	7
Workspace Interaction.....	7

Version History

This is version 1.0 of this document.

0.1	25 January 2010	First initial cut of document – introductory information on the usability project and the first cut of the hypotheses to test.
0.2	4 February 2010	After review with Jon McCann and Jeremy Perry, there have been revisions to the introductory information, correcting some typos and mistakes, and the hypotheses have been modified and expanded.
0.3	8 February 2010	Revisions to many of the hypotheses and introductory material made after feedback from Owen Taylor, Jon McCann, and Jeremy Perry.
1.0	10 February 2010	Public posting.

Overview

ABOUT GNOME SHELL

GNOME Shell is the defining technology of the GNOME 3 desktop user experience. It provides core desktop interface functionality. For example, GNOME Shell allows users to switch between windows and launch applications. GNOME Shell takes advantage of the capabilities of modern graphics hardware and introduces innovative user interface concepts to provide a delightful and comfortable desktop experience.

Test Objectives

The usability tests proposed in this plan will be the first formal usability tests proposed for the GNOME Shell. The main goals of these tests are to:

- Evaluate many of the claims made in the GNOME Shell design process and specification.
- Provide a framework for feeding observational data into the Shell design process in order to inform and revise design decision making.

Methodology

POTENTIAL APPROACHES

The GNOME Shell is a computing environment, not a single application geared towards a well-defined set of user goals and tasks within a particular knowledge domain. This plan therefore proposes a usability testing methodology that is not application-centric. This approach is meant to test assumptions about user interaction with a desktop shell rather than identify deficiencies in any single application's ability to accommodate specific user goals.

For example, this plan will not cover holistic, scenario-based task workflows such as:

Your manager just invited you to an important meeting that conflicts with your annual scheduled physical exam. Write an email to your doctor's office to rebook your physical exam.

The above task could be useful for identifying usability issues during the usage of a set of software applications (in this case, an address book and email client) for a reasonably common real-life scenario. Our interest, however, is in identifying issues in basic desktop shell functionality such as window management, mouse handling, and task switching. The example test above could help to incidentally identify the kinds of issues we're interested in, but their effect could not be analyzed unless enough users happened to stumble upon the same issues to provide enough data – leaving the effectiveness of our test to chance.

GNOME Shell introduces a new different set of basic desktop interactions, such as window management and searching within files and other documents, that are not intimately connected to scenario-based user tasks ('send my physician an email.'). This introduces a challenge in testing methodology: an approach that simply compares user performance of scenario-based tasks across desktop environments naturally tends to favor the more familiar desktop environments. We do not need to prove GNOME Shell is new and unfamiliar to users. A scenario-based approach would tell us little more than that.

A longitudinal study of scenario-based tasks in GNOME Shell would eliminate some of the effects of the novelty of an unfamiliar desktop environment. Unfortunately, such a study is necessarily a long-term project and wouldn't satisfy the immediate needs of the project or the objectives outlined above. However, it is expected that such a study will be performed in addition and as a supplement to the one described here.

The approach we propose here involves crafting tasks to align to specific hypotheses in the GNOME Shell design, and testing the hypotheses' validity comparatively across desktops. The tasks involved are designed to account for the familiarity bias in order to avoid distorting the results (either positively or negatively.)

EXAMPLE TASK CREATION

Let's walk through the creation of example task in order to illustrate this proposed methodological approach. Here's an example of a hypothesis we would like to test:

A persistent window list is distracting to the user. A user will be less distracted if the window list is not always visible on the screen.

A potential task to test this hypothesis might be to ask the user to perform a task that requires concentration or would be adversely affected by a distraction. One idea is to ask the user to transcribe a set of handwritten notes, and introduce a distraction during the completion of the task. Ask some users to complete the task in a desktop environment that has a persistent window list, such as GNOME 2, and ask some users complete the task in GNOME Shell, which does not have an always-visible window list, and compare the results.

There are a few ways you could evaluate the user's performance in such a task:

- time to complete the transcription task
 - Total time from beginning of task to completion of task
 - Amount of time spent actually transcribing the document
 - Amount of time spent in the desktop environment not transcribing the document
- user's level of distraction as assessed by:
 - User self-rating using Likert scale-based questionnaire completed post-task

- Number of switches between keyboard and mousing device (transcription is primarily keyboard-based, so if the user is required to switch input modality to a mousing device, she is not working towards completion of the task)
- Task completion rate

The following outcomes would provide support for the example hypothesis in the GNOME Shell environment compared to GNOME 2:

- The user completes the task.
- The user takes less total time to complete the task.
- The user spends more time actually transcribing the document than otherwise.
- The user indicates a lower level of distraction in questionnaire results.
- The user switches between keyboard and mouse less.

Hypotheses

WINDOW MANAGEMENT

1. A persistent window list is distracting to the user. A user will be less distracted if the window list is not always visible on the screen.
2. It is easier for a user to find the window they are looking for if recognizable visualizations of the windows are provided rather than just application icons.
3. Users have a greater feeling of control over their applications if they can view a large visual catalog of open windows rather than a textual or icon-based summary anchored to a panel.
4. A zoom animation to shift the user from full-size windows to an overview of open windows helps the user feel more comfortable with the transition from the full-screen desktop to the Activities overview.
5. A zoom animation to shift the user from full-size windows to an overview of open windows helps the user feel they are widening their “field-of-view” or stepping back to gain perspective on their work.
6. Users shifting from full-size windows to the window overview will spend a minimal amount of time in the window overview, and the majority of their time in the full-size windows.

APPLICATION LAUNCHING

7. When presented with an application search interface and a menu-based application browsing interface, users will try to launch an application via search first when the name and location of the target application is not recalled or known by the user.
8. Category-based application lists are more distracting than usage-based application lists.
9. If an application list that only shows a subset of the total applications available does not contain the type of target application the user seeks, the user will be able to locate it in an extended flat list containing all applications in a reasonable amount of time in comparison to locating the application in a full and category-based application menu listing.
10. Locating an application through the use of spatial memory and visual recognition is more comfortable than locating an application through the recall of where it is filed categorically.

11. If application launchers consistently open the most recent window of the already-running application rather than launch a new window of that application, users will have a better sense of control over their desktop.
12. If an application list that only shows a subset of the total applications available does not contain the type of target application the user seeks, the user will be able to find the search box and locate the application using search in a reasonable amount of time in comparison to locating the application in a full and category-based application menu listing.
13. That the search box located in the activities overview searches across applications, documents, and the file system will not surprise or confuse users when they evaluate the search results, and the functionality will be convenient to them.

NOTIFICATION HANDLING

14. Requiring the user to open an application in order to turn off a given notification is more distracting than interacting with only the notification itself in order to turn it off.
15. Notifications that provide the user with information that does not prompt them to act cause less stress than those that demand attention.
16. Incoming mail notifications for most users are too frequent to be useful by default – the interruption created by their frequency outweighs their utility to the user.
17. A short messaging tray aligned along the length of the bottom of the screen is less distracting than a rectangular bubble message spawned from an icon in the upper right corner of the screen.
18. Notifications that optionally allow user to respond to a message or make a decision without having to switch context will reduce context switches by the user and allow them to focus better on their current task.

USER FOCUS / IMMERSION

19. Users can achieve greater levels of immersion on tasks if they are performed in a less cluttered environment.
20. Users will achieve higher levels of immersion in environments where a greater percentage of screen real estate is dedicated to the task they are working on.
21. Users will achieve greater levels of immersion in environments where areas of the screen not directly related to their task are not as bright or noticeable.
22. Users will wander off-task more frequently in an environment that provides more distractions.
23. Users are better able to focus on a task if the non-primary desktop components supporting their task are less bright than those that are.
24. Users will feel more in control of their computer if there are less clickable spots on the screen that could launch applications or cause changes unrelated to the task they are focused on.

DOCUMENTS

25. Users are more likely to use a recent documents listing if it is visible to them more often in their desktop interactions.
26. That the search box in the activities overview searches both document names and

applications will not surprise or confuse users, and will be interpreted as convenient.

27. Users will feel more frustrated when attempting to locate a document by browsing a hierarchy of documents as compared to attempting to locate a document via search.

TASK SWITCHING

28. Not requiring a click to access the applications overview, in comparison to requiring a click to access it, will be perceived as more efficient by users.
29. Not requiring a click to access the applications overview will allow users to keep their gaze directed on their document while shifting from viewing their document to viewing all open windows.
30. The placement of the Activities hot corner in the upper left will cause users to be less prone to accidentally triggering it than if it were placed in the upper right corner.
31. Placement of the activities hot corner in the upper left of the screen will reduce confusion about it being similar to the Windows start menu.
32. In non-cloned multi-display configurations, the top menu bar is only needed on the primary display.
33. The hot corner's ability to open the Activities Overview with or without a click will not disorient the user.

WORKSPACE INTERACTION

34. It is easier to transfer applications between workspaces when visually dragging them rather than moving them via a textual command menu.
35. If desktop workspaces are not persistently visible, users who are not familiar with using them will be less likely to accidentally enter them and lose windows in them.
36. Users are much less likely to lose or forget windows placed on workspaces that are not in view on the desktop when they are visually represented in the applications overview.