WARM Heuristic Evaluation
May 7, 2013

Executive Summary

This document outlines the general principles for interaction design called "heuristics" as mapped to highlighted areas of the WARM application as well as general commentary and analysis of the application.

Heuristics
For the 10 usability heuristics there are 23 items highlighted. Twenty of the highlighted items are in conflict with these general principles

Recommendations

These recommendations directly address areas itemized in the report known to cause issues including personalization, error handling and user navigation:

1. Hide all unnecessary navigation options based on role or task.
2. Normalize the display of navigation areas. Secondary navigation areas that are not in use should be empty, not disappear.
3. Populate portlets with user's data instead of allowing users to manually do it.
4. Pre-populate filters based on user's roles, and show results instead of a question when they hit the screen. They can always perform another filter.
5. Don't make users perform repetitive actions that always have the same value – ie: Populate for timesheets and "ready to submit" for PM tasks.
6. Use type-ahead instead of searching lists.
7. Make error messages specific to tasks being performed.
8. Disable any button that has no action associated when the user clicks on it.
9. Add a breadcrumb trail so users can back out of tasks.
10. Disable (or hide) any links that leads to a function that a user has no privileges to use.
11. Create wizards to lead users through tasks instead of relying on external documentation.
12. Add short and pointed instructional text within the application.
13. Add dynamically saved filters for last 5 filtered items.
14. Streamline documentation, make it task based so it is easier to consume and refer to.
15. Remove or fully explain "Power Filtering"

•

Heuristics
Below are the general principles for interaction design called "heuristics" as mapped to highlighted areas of the WARM application.

Visibility of system status
The system should always keep users informed about what is going on, through appropriate feedback within reasonable time.

[+] The notifications section of the homepage keeps track of all items a user is associated with.

[-] Error messages are disconnected from actions and are generic.

[-] Action Buttons always appear active.

Match between system and the real world
The system should speak the users' language, with words, phrases and concepts familiar to the user, rather than system-oriented terms. Follow real-world conventions, making information appear in a natural and logical order.

[+] ‘Projects’ filtered results display a nice overview complete with a schedule. It would be nice if this was the default.

[-] Specific knowledge is needed to perform actions. Actions are difficult to complete without external explanations and resources.

[-] Filters are not pre-populated with the user’s known content.

[-] When users perform actions such as filtering and there is a single result, the result is not selected by default on either the search results or general filter. This forces the user to perform a search, select the single option; click ‘ADD’ then click “FILTER’ to select a single entity.

[-] The user is forced to perform duplicate actions. As example the user must click a “Ready to submit” checkbox before clicking “Submit” or “Populate” to insert values in a timesheet

User control and freedom
Users often choose system functions by mistake and will need a clearly marked "emergency exit" to leave the unwanted state without having to go through an extended dialogue. Support undo and redo.

[-] Lots of navigation makes it prone to error

[-] If user clicks on items within the navigation the state of any changes the user has not saved is lost without any notification to the user.

[-] Lack of breadcrumb trail makes it hard to traverse back a step without remembering which widget you clicked on to get to where you are.

Consistency and standards
Users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform conventions.

[-] Save and submit buttons are inconsistent throughout the application

[-] Some filters can be saved. Advanced filters cannot.

[-] Disabled options and links appear active. Unavailable links let the user enter areas they do not have rights to before informing them as such.

Error prevention
Even better than good error messages is a careful design which prevents a problem from occurring in the first place. Either eliminate error-prone conditions or check for them and present users with a confirmation option before they commit to the action.

[-] All menu items are available at all times, even if the user does not have privileges.

[-] Power Filters let the user create filters that produce an error without the ability to go back and edit or continue with previous settings.

Recognition rather than recall
Minimize the user's memory load by making objects, actions, and options visible. The user should not have to remember information from one part of the dialogue to another. Instructions for use of the system should be visible or easily retrievable whenever appropriate.

[+] 'Projects' filtered results display a nice overview complete with a schedule.

[-] System requires guides and help for users to perform actions. This reduces the overall effectiveness of the application

[-] Messaging throughout the application would aid in task completion. Currently there is no direction oriented texts within the application

Flexibility and efficiency of use
Accelerators -- unseen by the novice user -- may often speed up the interaction for the expert user such that the system can cater to both inexperienced and experienced users. Allow users to tailor frequent actions.

[-] Filters are not pre-populated with the user's known content.

[-] There are no wizards or on screen help to lead the user through complicated tasks.

Aesthetic and minimalist design
Dialogues should not contain information which is irrelevant or rarely needed. Every extra unit of information in a dialogue competes with the relevant units of information and diminishes their relative visibility.

[-] All functions and navigation items are available throughout the application.

Help users recognize, diagnose, and recover from errors
Error messages should be expressed in plain language (no codes), precisely indicate the problem, and constructively suggest a solution.

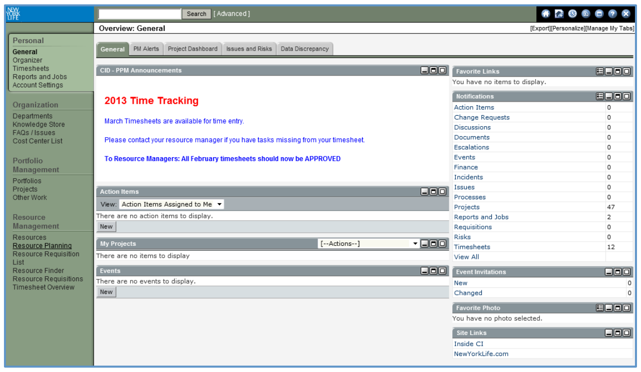
[-] Error messages are very general, providing no assistance recovering from errors.

Help and documentation
Even though it is better if the system can be used without documentation, it may be necessary to provide help and documentation. Any such information should be easy to search, focused on the user's task, list concrete steps to be carried out, and not be too large.

[-] Documentation while comprehensive is difficult to follow in places and assumes system and tool knowledge.

Commentary and Analysis

WARM has a tremendous amount of information and flexibility. It is this flexibility and availability of options that make the process and the interface confusing.



2013 Time Tracking

March Timesheets are available for time entry.

Please contact your resource manager if you have tasks missing from your timesheet.

To Resource Managers: All February timesheets should now be APPROVED

There are no action items to display.

There are no items to display

There are no events to display.

You have no items to display.

Action Items	0
Change Requests	0
Discussions	0
Documents	0
Escalations	0
Events	0
Finance	0
Incidents	0
Issues	0
Processes	0
Projects	47
Reports and Jobs	2
Requisitions	0
Risks	0
Timesheets	12
View All	

New	0
Changed	0

You have no photo selected.

To the right is the common landing page for users.

The WARM application uses a control board methodology where the majority of options are available to users at all times. While this adds a tremendous amount of flexibility it makes the interface busy and confusing.

Personalization

WARM has the ability to be personalized to meet the specific needs of users; most users however will never personalize it. What remains on the homepage is a strange assortment of portlets.

If the user does decide to personalize the homepage by adding their projects for example it is difficult to decipher how.

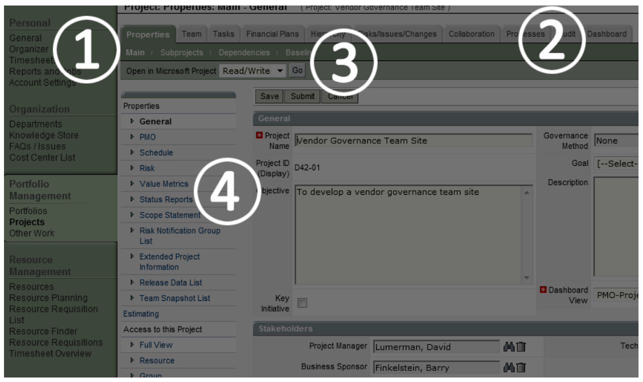
The various options are spread through the interface. To make the portlet appear or disappear, the user enters the “personalize” link in the upper right. To configure the columns, the user uses the “customize” drop down within the portlet area. To add items to the display, the user must go to each specific project and elect the project to be included from a link in the upper right of the project screen. This makes the likelihood of someone utilizing the feature very remote.

Tab sets are a common interface crutch used by WARM. Along with the tabs across the top there are tab cutouts on the left navigation. There is however a difference. The tabs on the top take the user to discrete views of content while the tab cutouts on the left just segment the functionality, providing no real value.

As the user delves deeper into the application additional functions and navigation appear.

Navigation Functionality

1. Left Navigation
   The left navigation is broken up into groupings. When an item is selected from a grouping the whole group set is selected. There is no additional functionality associated with this grouping. The organization of the information is also suspect. As examples, items such as FAQs and Knowledge Store are ill suited to an Organization grouping



General

Project Name: Vendor Governance Team Site

Project ID (Display): D42-01

Objective: To develop a vendor governance team site

Key Initiative: ☐

Governance Method: None

Goal: [--Select-

Description:

Dashboard View: PMO-Proj

Stakeholders

Project Manager: Lumerman, David

Business Sponsor: Finkelstein, Barry

2. Top Tabs
   The top tabs are the secondary navigation. This navigation is inconsistent, and appears dependent on the left navigation option selected if appearing at all. This secondary navigation appears in 4 of the 18 left navigation options and is not based on a specific left nav grouping.

   The tabs are also inconsistent within an individual top navigation option, not appearing when selected initially but appearing when a filter is performed. This type of inconsistency makes it more difficult for a user to decipher how an interface works.

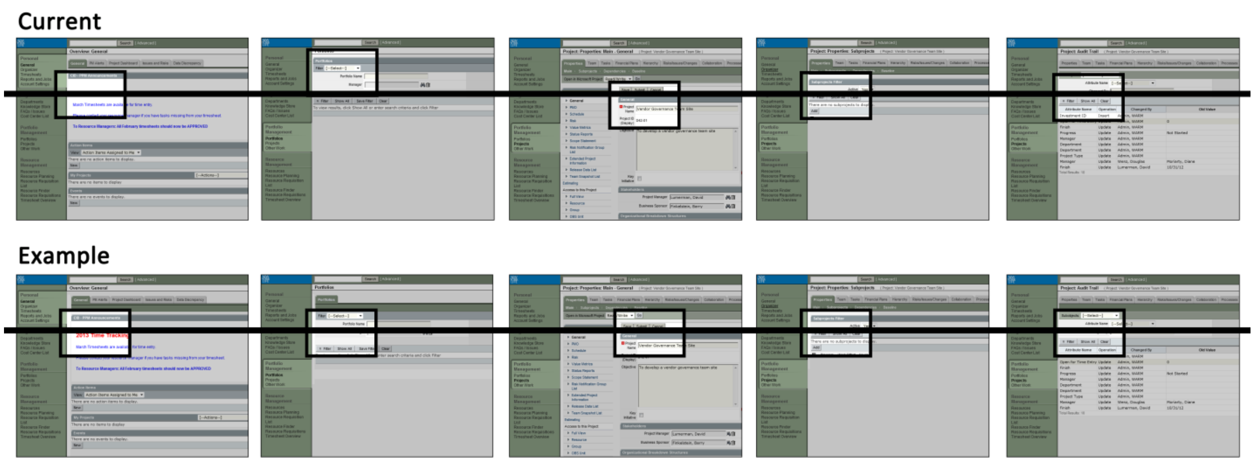
3. Sub Tabs Options
   The sub tab options are dependent on the tab that you currently have selected. This sub area works like the top tabs, appearing and disappearing depending on the option selected. As mentioned with the top tabs, this makes it more difficult for the user to decipher what is going on, forcing your perspective to change to the interface adjustments. It would be better to have the empty bar than one that appear and disappears at random.

4. Secondary Left Navigation
   The next set of hierarchal navigation is the secondary left navigation. This navigation is dependent on the sub tab selected. This is a curious implementation given the majority of sub tab options display different cuts of the information based on the tab. This is not however the case for the Projects>Properties>Main sub tab option.

This is a tremendous amount of navigation, which while unfolding, adds more and more possible options to the display. What it does not add however is an easy way to traverse back up the tree, to show in a single location where you currently are. Since all the navigational options are still available it is possible to find the originating link, click on it and return to previous areas but this is not ideal, it is time consuming and exacts a mental strain on the user.

The viewing of all navigational options at all times is not ideal, it adds confusion to the user and makes the interface more difficult to use. The application choice to collapse the tab and sub tab options area adds significantly to the complexity of the interface.

In the examples shown below you can see the visual differences in changing vs. an unchanging interface for the majority of screens as the user's eyes are forced to reinterpret each screen to find the beginning of the content area. The "current" visuals include 5 common screens as they are today. The "Example" below it normalizes the tabs, tab options as always displayed.



One would suspect the abundance of navigation is a common complaint to CA. Evidence of this is the ability to hide the left navigation. But while this is a useful feature, it is inconveniently disconnected from the point of action, meaning the button is way across the page from the navigation.

A combination of placing the button in proximity to the navigation and adding some application intelligence to the hide the left navigation would simplify the interface and prevent selecting left navigation options by accident.

Application intelligence

Application intelligence is the ability of the system to perform the most common actions for the user and present to the user the most appropriate interface to take action in a way that reduces the time needed to complete actions.

Listed are some examples where enhanced application intelligence would improve the application.

1. Providing know lists
   The WARM application follows the methodology that the user must enter a query to receive a set of results to take action on.

   The system however has key pieces of information. The system knows the user. The system knows the user's "stuff" and potentially knows what the user has seen before. With these key pieces of information the system should be smart enough take the user to the appropriate pre-populated lists.

   For example, each Project Manager is listed as the manager to a set of projects. Knowing this information the list of projects should be pre-populated when entering the projects tab by default. This methodology should be followed wherever possible. This is the same for the home page portlet. Pre-populate instead of searching.

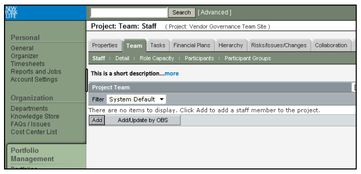
2. Intelligent options
   Providing intelligent options based on a user's role, preference or a combination of both simplifies a complicated navigational structure. Sections such as "Personal" and "Organization" are infrequently used but take 50% of the left navigation space.

3. Just in time communication
   The WARM system relies on help, documents and FAQs for users to traverse the application. Providing descriptive help text on the screen adds to a user's comprehension and jumpstarts them into the process.

   Embedding short descriptions with either links to or expanded texts to help the user start and complete tasks is less mental stress than searching through quick start guides.

   Taken to the next logical progression, based on a person's usage and familiarity the texts can disappear over time as processes become rote.

General Description

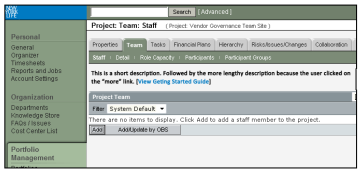


Project: Team: Staff (Project: Vendor Governance Team Site)

This is a short description...more

There are no items to display. Click Add to add a staff member to the project.

More Details

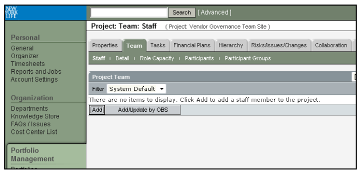


Project: Team: Staff (Project: Vendor Governance Team Site)

This is a short description. Followed by the more lengthy description because the user clicked on the "more" link. [View Geting Started Guide]

There are no items to display. Click Add to add a staff member to the project.

Habitual User



Project: Team: Staff (Project: Vendor Governance Team Site)

There are no items to display. Click Add to add a staff member to the project.

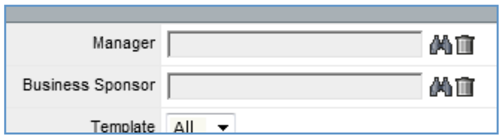
Interface Irregularities

Throughout the interface there are conventions that are non-standard with current interfaces. Many of these conventions require the user to go through multiple steps to complete simple actions. Below are examples.

1. Browse Functionality
   The browse function allows the user to select a value to perform searches. The function is robust and flexible allowing the user to perform searches by many criteria or search the list of options to make a selection.

   The more common use case however is for a user to know exactly the value required. The current interface bars the user from direct entry, forcing the use of the advanced filter.

   Many interfaces with similar challenges opt for a type-ahead function where there is a recursive search based on the characters a user types. The user never leaves the screen and the action is completed efficiently.



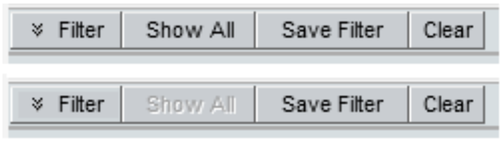
Manager	
Business Sponsor	
Template	All

2. Button State
   Throughout the interface the action buttons always appear active, leaving it to the user to decipher when they are appropriate to use. If an inactive button is chosen the interface provides no feedback.

   The desired action is to disable buttons that are not currently available.

   In the illustration to the right the current scenario is that something entered into a search field, but the filter has not been initiated.

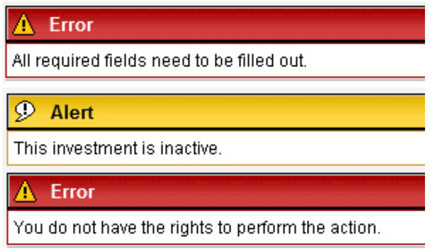
   The top image is how the buttons appear currently, the bottom image displays a more appropriate interface display, disabling buttons based on the application state.



Error Messaging

1. Messaging and Action
   Error messages are disconnected from the point of entry. This requires the user to search the interface for the position of the error.

   The messages themselves are also vague. While this makes for easy application development, it puts a burden on the user to locate the error and decipher the reason and efficiently correct the error.



Error
All required fields need to be filled out.
Alert
This investment is inactive.
Error
You do not have the rights to perform the action.

   In some instances Errors are substituted for Alerts. Errors are application states prevent the user from moving forward in a process. Alerts are instances where the user can perform action but the system would prefer alternate information.

2. Alerts
   Alerts are displayed similarly to actions, but do not add information on how to correct them.

3. Preventing errors
   Many errors can be avoided by not providing options that directly lead to errors. In the instance above (bottom), the path of “Other Work > Properties > Financial” produced an error that could have been avoided by disabling the option to an area the user does not have access to.

Filters
Throughout the application the user can save filters and mark them as the default filter, which potentially is a timesaving feature. Filters are muddled however because they cannot be transferred to other areas with similar criteria. Filters are further muddled because of “Power Filters” which can contain basic logic. These filters cannot be saved.