

# KRAD - Error Message (& other validations) Architecture

## Purpose

There are two main purposes for this requirement:

**First:** If there are multiple errors or other validation messages on a page after submitting a form, provide users with an "advance organizer" summary, letting them know how many there are. The human brain needs to build a sense of the overall "landscape" first, before diving down into details. If the design doesn't enable this, users feel less oriented to the task ahead of them (the task of correcting the problems) and project that the task will be bigger, more overwhelming.

Make it easy to navigate from the error summary to the individual errors to fix them. Research shows that leaving users to locate each individual error, without providing them with this type of a "macro" view, makes them feel more lost in the "micro" specifics - and less certain they have located and fixed all errors, before re-submitting.

This summary view and ease of navigating is needed for all users, and is needed in particular for non-sighted users who may be using screen-readers and keyboard-only access.

**Second:** Place the validation error indication and the error text (applies to warnings and informational messages also) as close as possible to the field that is in error -- to make it easier for all users to locate and then fix the fields in question. This proximity and visual signalling is helpful for all users, and is needed in particular for low-vision users who may be using screen magnification.

## Detailed Description

See sections below and these slide decks:

- [Review in KAI on 3-1-2012 and in "Feedback Friday" session on 3-9-2012 \(default behaviors for Validation messaging\)](#)
- [Review in KAI on 3-29-2012 \(options for Validation messaging\)](#)

## Usage Scenarios

### Usage Scenario One:

- User submits complex form (after field-level client-side validation, if any, is done & passes client-side validation).
- Application finds errors (and/or other validation conditions .... warnings or informational messages) in multiple fields on the page, across sections of fields.
- Application returns best practice error&validation message handling =
  - At the top of the page there is a page-level summary of the validation messages (so that it is early in the screen-reading order) -- the summary includes the # of errors and other validation messages in each section that has errors, not the error text itself. Focus is placed here when the application returns messages on more than 1 field.
  - User can jump directly to any of the sections that has messages, because each section in the summary is marked up as a link.
    - If there is more than 1 field with messages in the section, the focus(link) moves to the section's summary, and from there the user can jump to any of the fields with messages, because each field in the section-level summary is marked up as a link. Section-level summaries are enabled by the architecture and having them is the recommended default, but it is an application decision whether to include only a page-level summary in addition to the field-level messages.
    - If the application does not include section-level summaries, when jumping to the section-level from the link in the page-level summary, the focus(link) moves to the section-level heading. (Research this versus placing in the first field that has an error in that group.)
  - Every summary of errors (page level and section level) starts with a heading.
  - The user can get back to the page-level summary using standard shortcuts for moving to top of page. (We could create a custom global shortcut to get back to the page-level summary - but this will be determined by a different requirement having to do with [keyboard support](#).)




## Mocks and Diagrams

Note: Consider the below to be "low-fidelity" mockups - these are not intended to specify the precise visual styling, relative placement, wording or branding. Those details will come later, in specifications. These are intended to convey the task flow and UI architectural constructs to support, only.

## Example Validation Message treatments added to Titles and Headers:

Titles and headers include the validation message icon that is associated with the highest severity message included in the set. In addition, text is appended to the title or header text, indicating how many messages are associated with that set (page, section, etc.). See example below:

*Example 2 - Information in Titles and Headers:*


-  **Page Title - x# errors and yz # other messages in form submission**
-  **Page Title - y# warnings & z# information messages in form submission**
-  **Page Title - z# information messages in form submission**





If there are all 3 types of messages, the number of errors is identified, and the aggregate number of other messages is identified.

## Example Page-Level Validation Message Summary:

Each of the four lines below are links, to the specific section in question.

**Form Start**

 **20 errors in form submission**

-  The '**Personal**' section has 4 errors
-  The '**Office**' section has 6 errors
-  The '**Overseas**' section has 6 errors
-  The '**Emergency Details**' section has 4 errors

With the heading text-variable filled in:

```
<h2 id="pageSummaryHeading" class="errorSummaryHeading">14 errors in form submission</h2>
```

## Example section-level Validation Messaging:

- The error state indication is placed near the field – see the icon and field visual treatment - the border outline (the precise visual treatment and placement will be determined.) This proximity helps low-vision users (using screen magnification) & all users to locate the targeted fields.
- Error text is placed in a section-level summary above the group's fields AND is placed in a field-level error tooltip. The tooltip appears contiguous with the field when the user hovers over (mouse) or selects the field.

Emergency Details - 4 errors

- ! Address: Value is not allowed
- ! Zip Code: Too many typos
- ! City: The value you entered was not recognized
- ! Maiden Name: Too many typos

This information will not be shared with third parties

<p><b>First Name</b></p> <p>Lorem ipsum dolor sit amet</p> <input style="width: 90%; margin-top: 10px;" type="text" value="Enter text here..."/>	<p><b>Last Name</b></p> <p>Lorem ipsum dolor sit amet Lorem ipsum dolor sit amet Lorem ipsum dolor sit amet Lorem ipsum dolor sit amet Lorem ipsum dolor sit amet</p> <input style="width: 90%; margin-top: 10px;" type="text" value="Enter text here..."/>	<p><b>* Address</b></p> <input style="width: 90%; margin-top: 10px;" type="text" value="Enter text here..."/> <p style="font-size: 0.8em; margin-top: 5px;">Should not be more than 100 characters</p>	<p><b>* Zip Code</b></p> <p>Lorem ipsum dolor sit amet Lorem ipsum dolor sit amet Lorem ipsum dolor sit amet</p> <input style="width: 90%; margin-top: 10px;" type="text" value="Enter text here..."/>
<p><b>* City</b></p> <p>Lorem ipsum dolor sit amet Lorem ipsum dolor sit amet Lorem ipsum dolor sit amet Lorem ipsum dolor sit amet</p> <input style="width: 90%; margin-top: 10px;" type="text" value="Enter text here..."/> <p style="font-size: 0.8em; margin-top: 5px;">This should be a valid social security number</p>	<p><b>Date of Birth</b></p> <p>Lorem ipsum dolor sit amet Lorem ipsum dolor sit amet Lorem ipsum dolor sit amet</p> <input style="width: 90%; margin-top: 10px;" type="text" value="Enter text here..."/>	<p><b>High School</b></p> <p>Lorem ipsum dolor sit amet</p> <input style="width: 90%; margin-top: 10px;" type="text" value="Enter text here..."/>	<p><b>Country</b></p> <p>Lorem ipsum dolor sit amet Lorem ipsum dolor sit amet Lorem ipsum dolor sit amet Lorem ipsum dolor sit amet</p> <input style="width: 90%; margin-top: 10px;" type="text" value="Enter text here..."/>

**\* Maiden Name**

Lorem ipsum dolor sit amet Lorem ipsum dolor sit amet Lorem ipsum dolor sit amet



### Example Field-level Validation Messaging:

There is additional visual indication on the appropriate field itself when the user hovers over the specific error message link in the group-level summary. This helps to locate the appropriate field, among other competing information in the user's field of view. See the example below, where the user is hovering on the first link (or has moved to the link via keyboard):

**Overseas - 6 errors**

- First Name: Must be 10 digits
- Last Name: Too many typos
- Zip Code: Required characters are missing
- Date of Birth: This is not a valid address
- High School: Does not compute
- Maiden Name: Required characters are missing

We would like to know a bit more about yourself


<p><b>* First Name</b></p> <p>Enter text here...</p>	<p><b>* Last Name</b></p> <p>Enter text here... </p> <p>Must be answered with your eyes closed</p>	<p><b>Address</b></p> <p>Enter text here...</p>
<p><b>City</b></p> <p>Enter text here...</p>	<p><b>* Date of Birth</b></p> <p>Enter text here... </p> <p>Enter the value as it appears on your tax</p>	<p><b>* High School</b></p> <p>Enter text here...</p>

There is also a different visual indication-state on the field itself when the user selects it and can enter data. The tooltip that contains the error message text is also displayed, contiguous with the field, when that field is selected. The example below shows the placement of the tooltip to be immediately above the field, with its left margin beginning at the center of the field, but the default placement algorithm and the visual treatment details will be determined in the specifications to come later. Consider this simply a 'low fidelity' mockup and proof-of-concept, with the exact rendering to be determined.

The fundamental requirement is that the tooltip's default placement be contiguous with the field while not occluding it. See example below:

**\* Zip Code**

Required characters are missing




Enter text here... 

This should be a valid social security number

The other aspect that is fundamental to the tooltip construct is that it remains visible with the field while the user enters the data.

### Example field-level tooltip content:

If there are multiple messages associated with the same field, they are included in the single tooltip associated with the field, with the highest level severity messages first:




-  Server-side Validation error text.
-  Server-side Warning text.
-  Server-side Informational text.

Below is an example of a client-side validation tooltip, which will be visually distinct from a server-side validation tooltip, since the behavior differs (the precise visual treatment TBD). A client-side validation tooltip pops up immediately upon exiting the field in question, so the user can correct it immediately, before submitting the form:


**Client-side Validation error tooltip:**  
 Lorem ipsum aejjtetlele;lawrj

It is possible that after a page is returned with server-side validation errors, a user could create a new client-side error in trying to fix a field. In this case, the client-side validation error will be added to the top of the server-side tooltip that is already associated with the field:

**Client-side Validation error tooltip:**  
 Lorem ipsum aejjtetlele;lawrj

-  Server-side Validation error text.
-  Server-side Warning text.
-  Server-side Informational text.

When users have made a change to a field that has server-side validation messages associated with it, the tooltip changes slightly, in recognition that the user has already made changes to it, that might resolve the problem(s). However, because we can't tell until the form is resubmitted whether the change resolves the problem(s), the tooltip content remains associated with the field (when the user focuses on the field, the tooltip pops up):

 **You've made a change which might fix these - resubmit to validate!**

---

- Server-side Validation error text.
- Server-side Warning text.
- Server-side Informational text.

**Mocks and Diagrams summary:**

The strategy outlined above addresses the need for a summary of errors (when there are multiple) and the need to place the error indication and text as close to the field as possible -- helpful for all users and also needed particularly for low-vision users.

The tooltip strategy outlined above also solves the formatting problems we otherwise create when we place error message text within each field (whether immediately below or to the right), requiring other space to be pushed aside or down. Associated challenges include:

- We don't know how wide or deep the other aspects associated with the fields and sections will be, how adding the error text will affect the relationship/layout with other field groups on the page.
- Placing content inside a form field's label means that all this content will be announced by a screen reader as the field's accessible name, rather than as the field's accessible description or state information. This is too verbose.

This type of tooltip display of error text associated with form fields will be KRAD's default handling and is the recommended practice. However, KRAD is an open architecture, and will enable applications to turn off this default and turn on their own defined default (for example, an application could define their default to be placing the field-level error text directly below the field, thereby pushing down and aside all other content on the page.)

## Code Snippet Example:

You can try out an interactive example code snippet with this behavior here (thanks to Hans Hillen from The Paciello Group for this working prototype!):

[Error message handling - code snippet example](#) (Click the submit button toward the top left in the prototype, to see example behavior.)

## Performance

*If applicable, list expectations for performance (optimal and worst cases would be fine, give time in seconds):*

## References

See [tooltip requirement](#).

## Requirements Listing

### The following applies to server-side validation messaging:

#### Off-screen Validation Message Summary with link

- A hidden (offscreen) summary sentence with a link to put focus on the page-level summary should be provided for screen readers. (e.g. "Your form submission contained errors. Go to the *summary of errors*" – italicized string is a link to the page-level summary).
- This summary should have the ARIA role of "alert".

#### The browser window title (e.g. HTML "Page Title")

- After server-side error, on-loading the page (before displaying it), append the number of validation messages to the browser window title - the HTML page title (e.g., "*Page title - x#* error messages in form submission" – italicized # is a variable that gets filled with the appropriate #; italicized string contains the appropriate page title).
- If there are multiple levels of validation messages (errors, warnings, informational messages),
  - If all 3, the text appended to the page title gives just the number of errors and the aggregate number of other messages.
  - If just 2, the page title gives the # of each, with the highest severity category first (# of errors before # of warnings or informational messages; or # of warnings before # of informational messages).
- After a successful submit (no validation messages), on re-loading the page (before displaying it) the page title should revert to its original condition.

#### Page level summary

- Page level summary should be shown /expanded by default.
- Page Level summary of errors needs to have an HTML header (e.g H3).
- This header is the same as the page title: page title appended with the number of validation messages. See more details above in the page title requirements.
- the validation message icon that is appended to the page level summary is a single icon -- it is the icon that is associated with the highest severity message type that exists in the form/submitted: the error icon if at least 1 error exists, the warning icon if no error exists but at least 1 warning exists, and finally, the informational message icon if only informational messages exist. (tbd - other visual treatment/change associated with the page title)
- If the user collapses this page-level summary of validation messages, the validation message indication (icon and other visual treatment associated with the page-level header) and the header with the appended text is still visible, and can be re-expanded (via the standard expand/collapse control).
- Following the heading, the Page level summary includes a list.
  - Each list item is a link to the appropriate summary for that section/sub-section (the link text = section title and a summary of number of validation messages. E.g., "The *Personal* section has 4 errors and 2 other messages", "The *Office* section has 5 warnings and 2 informational messages"). See page title section above for more details.
  - If there are no sections/sub-section summaries, each link will go to the appropriate section/sub-section header. (Research question: the pros/cons of going to the first error in the section/sub-section instead.)
- For a page that has a mix of fields with validation messages, with some being within a section and some being outside of a section (directly in the page group), for the validation messages that are not in a section, the link in the page level summary gives a summary of the number of validation messages and describes the fields' position (e.g. "<x#> validation messages before <Section W>", "<Section W> has <y#> validation messages"<z#> validation messages after <Section W>"). For fields that are outside of sections and are between two sections, use the "After Section <#>" designation.
- For pages where there are no sections (i.e. all fields are displayed directly in the page group), display the page level summary header (same as above) but collapsed by default: Include an off-screen link with the title "Click for validation message details and links to the invalid fields" (for screen-reading users). When expanded, show the list of links, with each list item/link containing the validation message

text itself, and linking to the appropriate field, in this format: "*Field name: Validation message text*". For example, "*First name: Must be 10 digits*". (see also next bullet)

- If there are multiple validation message types (errors, warnings, informational messages) associated with a field, the link text should contain only the text for the first highest severity level message (error if there is an error, warning if there is no error but there is a warning, informational message if there is no error or warning but is an informational message associated with the field). It will also list in text that there is a warning or an informational message, as appropriate. Note that the user will see all validation messages' text in the tooltip associated with that field -- "just in time", when they are in the field to correct it. This is described in the later section below on the tooltip content.
- For validation messages that apply to the entire page (cannot be corrected by the user by changing anything to resubmit), these messages are listed at the top of the page in the page-level summary, before any other validation message content. Example of this type of message could be "You are not authorized to submit this type of form. Contact your system administrator for permission." With this type of message, all other error validation and reporting **should** be suspended (don't bury the user in correcting fields, when they can't re-submit).
- (Secondary: When focusing or hovering over items in a page-level validation summary, the corresponding section -- or field, if it is a single error -- will be highlighted visually. Because this is secondary, the work effort/sizing should determine whether we do this now in 2.2 or defer in the interest of other work items.)

## Section Level summary

- Section-level summary of validation messages needs to have an HTML header (e.g., H4).
- Configuration of whether to display section level (and sub-section level) validation messages should be per page, not per section/subsection.
- Section level summaries should be shown by default if/when there are validation messages in the section.
- Sections that have validation messages should be expanded by default.
- No section level summary if there is no validation message in that section.
- When there are validation messages in a section, the section header text is appended with the number of validation messages in that section – similar treatment as for the page-title (e.g., "*Section Y - x# error messages*" – italicized # is a variable that gets filled with the appropriate #; italicized string contains the appropriate section title). See page title section above for details.
- There is also a single validation message indication added to the section header - an error icon, warning icon, or information message icon - whichever is the most severe present in that section. For example if there is at least 1 error, the icon is the error icon. If there is no error but there is at least 1 warning, the icon is the warning icon. If there are no errors or warnings but there is at least 1 informational message, the icon is the information message icon. If the user collapses a section that has errors, this error indication (visual) and the appended section header is still visible.
- If there are sub-section level summaries, what follows the section-level summary header is similar to the page-level treatment above.
  - Following the heading, the section-level summary includes a list. Each list item is a link to the appropriate summary for the sub-section. (the link text = sub-section title and a summary of number of validation messages. E.g., "The *Address* sub-section has 2 error messages", "The *Telephone* sub-section has 1 error message"). See page-level summary section above for other example details.
  - Each link will go to the appropriate sub-section level validation message summary.
  - For a section that has a mix of fields with validation messages, with some being within a sub-section and some being outside of a sub-section (directly in the section group), for the validation messages that are not in a sub-section, the link in the section-level summary gives a summary of the number of validation messages and describes the fields' position (e.g. "<x#> validation messages before <Sub-Section A>", "<Sub-Section A> has <y#> validation messages", "<z#> validation messages after <Sub-Section A>"). For fields that are outside of sub-sections and are between two sub-sections, use the "After Sub-section <#>" designation.
- If there are no sub-section level summaries:
  - If there are no sub-sections on the page,
    - each list item/link contains the specific validation message (text) and each list item/links to the appropriate field.
    - if there are multiple validation message types (errors, warnings, informational messages) associated with a field, the link text should contain only the text for the first highest severity level message (error if there is an error, warning if there is no error but there is a warning, informational message if there is no error or warning but is an informational message associated with the field). It will also list in text that there is a warning or an informational message, as appropriate. Note that the user will see all validation messages in the tooltip associated with that field -- "just in time", when they are in the field to correct it. This is described in the later section below on the tooltip content.
  - If there are sub-sections of the page (just no sub-section level summaries),
    - For a section that has a mix of fields with validation messages, with some being within a sub-section and some being outside of a sub-section (directly in the section group), for the validation messages that are not in a sub-section, the link in the section level summary gives a summary of the number of validation messages and describes the fields' position (e.g. "<x#> validation messages before <Sub-Section W>", "<Sub-Section W> has <y#> validation messages"<z#> validation messages after <Sub-Section W>"). For fields that are outside of sub-sections and are between two sub-sections, use the "After Section <#>" designation.
- (Secondary: Same as for page-level summary -- When focusing or hovering over items in a section-level validation summary, the corresponding sub-section -- or fields, if there are no sub-sections -- will be highlighted visually. Because this is secondary, the work effort/sizing should determine whether we do this now in 2.2 or defer in the interest of other work items.)

## Subsection level summary

- Sub-section level summaries should be off by default (meaning there are no sub-section level summaries).
- If both section-level and sub-section-level summaries are turned on, the section-level summary exhibits same behavior as the page-level summary, while the sub-section level summary contains the specific validation messages (that are links to the appropriate fields).
- See additional details in the section-level above.

## The following applies to both server-side and client-side field-level validation messaging:

### Field Level - both server-side and client-side treatments

- Validation messages should have the ARIA role of "alert" so that they are programmatically announced during client-side validation.
- Use "aria-describedby" to programmatically associate field information (constraint text, instructions, tool tip and validation messages (validation messages IDs may need to be added via javascript during client-side errors) with the field itself using the field information IDs.
- (Note that "aria-required" and "aria-invalid" are already in use in KRAD 2.0 and need to remain in the code when appropriate.)
- Note: A client-side validation message could be created in a field that has a server-side validation message. For example, a form could be returned to the user with a server-side error. In changing the value entered into the field, the user could create a client-side error. This is handled as follows.
- By default, field level validation message is displayed as a tooltip - both server and client side validation messages - with visual differentiation (including different icon for error, warning, and informational message - exact treatments tbd). Note: we want noticeable difference between client-side and server-side errors since the behavior is different (error disappears immediately when fixed with client-side).
- Behavior with server-side:
  - Server-side error messages are handled in summary fashion (see material above) in addition to the handling at the field-level, described below.
  - There is visual indication that is apparent on the field, that indicates there is a server-side error associated with that field.
  - The server-side error tooltip opens when the user hovers (mouse, on hover) or puts focus in the field (keyboard, on focus). It closes when the user moves off. (Note that this applies as well to other user-manipulable form controls, in addition to fields, that could contain errors -- for example, to a checkbox group and a radio-button group.)
  - The server-side tooltip and visual indication remains associated with the field until the form is resubmitted. The server-side tooltip still closes onBlur or onMouseOut, but remains associated (i.e. part the DOM). When the user hovers on or focuses on the field, the tool tip will redisplay.
  - We'll change the visual style on server-side messages after the user interacts with it, so that the error state is still indicated with the field, but the user knows that they've changed that field.
- Behavior with client-side:
  - Client-side validation messages that are triggered immediately upon exiting a field, before submitting a page, are not handled in any summary fashion:
    - A client-side validation message tooltip pops-up immediately upon exiting a field, or other control, in a client-side validation error condition, and fades after a period of time (time length is tbd). It does not require hovering or field focus in order to be displayed. But after fading, it re-opens upon hover or focus on the field or control. (Note that this applies as well to other user-manipulable form controls, in addition to fields, that could contain errors -- for example, to a checkbox group and a radio-button group.)
    - If there is a server-side validation message already associated with the same field, the client-side validation error is included at the top of the server-side error message, in the same tooltip - different visual treatment will be applied (precise treatment tbd).
    - When a client-side validation error is corrected, the tooltip behavior depends on whether there is a server-side validation message associated with the field. If there is not, the tooltip closes and any other client-side visual indication disappears. If there is a server-side validation message tooltip associated with the field, the client-side error is removed from the tooltip. The tooltip remains open till the user moves out of the field.
  - Client-side validation messages that are triggered upon submitting a page, when a full-page client-side validation is performed before submitting to the server (which is the recommended practice), are displayed and handled in the same type of summary fashion as described above for server-side validation:
    - There is a page-level summary and, optionally, section level summaries and sub-section level summaries, in addition to the validation messaging at the field level. The section/sub-section summary options that have been chosen by the particular application for the server-side messaging will apply also to the *client-side-upon-submit messaging*. See sections above for details. See the following bullets for areas where the treatment will differ from those details.
    - When a client-side error/warning/informational condition is corrected, the tooltip and all field-level validation treatment will be removed from that field, AND all summary information will be updated to reflect this correction:
      - the page-level and any existing section/sub-section level summaries will be updated (the count will be decremented, the validation item will be removed from the summaries).
        - If the removal brings the number to 0 in a particular summary, the summary will disappear entirely. This includes at the page-level (if there are 0 validation messages on the page).
      - the html page title (browser window title) and any affected section and sub-section level headings will be updated (the error count will be decremented).
        - The validation indication will be removed entirely from the heading when the removal brings the number to 0 in a particular section or sub-section, and removed from the html page title when there are 0 remaining on the page.
      - The visual treatment will differ slightly between the client-side validation messages and the server-side validation messages.
  - Other:
    - Validation message indication on fields within table cells will not require an icon. There will be a perceivable border applied to the table cells that have been flagged by validation checking, that is the same visual treatment/border applied to input fields that have validation messages. (There may not be room to add an icon into table cells when they are too small for that additional content.)
    - A configuration option will allow field level validation messages to be displayed on the page instead of in a tooltip - in this case they will be displayed on the page after the field, between the field and its associated constraint text. Note that this configuration



option will not be the KRAD default (display in tooltip will be the default).

## Dependencies

List any functional or technical work that must be completed before work on this item can begin:

1. Research tooltip plug-ins (see [tooltip requirement](#))
2. Research and think through what should happen when ajax calls refresh a portion of a page and return server-side errors for just that sub-section/section of the page. Assume the default is field-level treatment, and summary for just that sub-section or section of the page - with the focus returned to that summary.

## Issues

List any issues that need to be resolved before work on this item can begin:

### Functional:

1. item
2. item

### Technical:

1. item
2. item

## QA or Regression Testing Plan

### Plan to test on:

- Firefox 10 and 11
- Chrome 18.0.1025.162 and 18.0.1025.163
- IE 8 and 9 (MS Windows only)
- Safari 5.0.6 and 5.1.5 (Apple MAC only)

**Required basic test:** Go through steps 1 and 2 using the typical, preferred navigation style (typical GUI users = selecting items and scrolling with mouse, entering data with keyboard).

1. Select a form to complete and submit: *(Note 1: Should we specify which form, and where in the build? We need to make sure we test with forms that have both client-side and server-side validation messages created for the fields, and have all 3 types of messages: errors, warnings, and informational messages.)*
  - a. Try leaving a required field blank and moving on to the next field. (Correct any client-side validations you get.)
  - b. Try putting incorrect data into any field(s). (Correct any client-side validations you get.)
  - c. Complete all required fields and correct any client-side validations you may have.
  - d. Save the form. Submit the form. *(Note 2: Should we specify which fields in which form to fill out with which values? We need to make sure we exercise a case which passes all client-side validations and has all required fields completed, but which will have errors, warnings, and informational messages on the form, with some but not all on the same field. Must have fields with messages in multiple sections.)*
2. After submitting a form that returns with multiple validation messages, review and correct each field that has an associated validation message(s). When there are no client-side validation messages and all fields that have server-side validation messages have been changed (to try to fix them), resubmit the form:
  - a. Check the navigation:
    - i. From the page-level summary, try out each link in the list to ensure it goes to the appropriate section header. (Come back to top of page using any method you choose, after following and checking each link.)
    - ii. From each section-level summary, try out each link in the list to ensure it goes to the appropriate sub-section header (if there are sub-sections) or to the appropriate field (if there are no sub-sections).
    - iii. (If there are sub-sections) From each sub-section level summary, try out each link in the list to ensure it goes to the appropriate field.
  - b. Check the behavior of a client-side validation message added to the tooltip of any of the fields flagged with server-side validation messaging:
    - i. Enter an invalid response into the field (e.g. numeric if only character data is valid and vice versa).
      1. Make sure the client-side validation message is added to the top of the tooltip for that field, upon moving out of the field.
      2. Make sure the client-side validation message is added to the appropriate message summaries (sub-section, section, page).
    - ii. Enter a valid response into the field (correcting the original server-side error as well as the client-side error you introduced):
      1. Make sure the client-side validation message is deleted from the tooltip for that field.
      2. Make sure the client-side validation message is deleted from the appropriate message summaries (sub-section,

section, page).

3. Make sure the server-side tooltip's and the field border's visual treatment are changed (colors changed to grey/black), to signal that you have made a change to this field that might correct the server-side message condition (we can't tell for sure until re-submit).

c. Fix all remaining fields that have validation messages associated with them, and re-submit the form.

#### **Additional tests:**

*Second: (This should be done by the devs before checking their code in for the milestone. It can also be included in QA testing.) Run a code-standards checker. See list of [code-checker alternatives](#). (Note - if this second test is run and checks well, the following checks may be redundant.)*

*Third: Select large fonts through the browser's settings (200%) and the browser setting to ignore the web page's font settings. Visually inspect the UI while repeating steps 1 and 2, to ensure that all UI elements inherit well the larger font size. If there are any problems, return the settings to their prior, normal state, and then set the screen to lower resolution, e.g., 800x600, or change the DPI to 200% via the operating system settings. Visually inspect the UI while repeating steps 1 and 2, to ensure that all UI elements inherit well these larger size settings.*

*Fourth: Select high contrast through the browser's settings as well as the browser setting to ignore the web page's color settings. Visually inspect the UI while repeating steps 1 and 2, to ensure that all UI elements inherit well the high contrast settings. If there are any problems, return the browser settings to their prior, normal state, and then select a high contrast theme in the operating system settings. Visually inspect the UI while repeating steps 1 and 2, to ensure that all UI elements inherit well the high contrast settings.*

*Fifth (Not testable yet in M1): Keyboard-only test. After completing these above, repeat steps 1 and 2, but without using the mouse, using only the keyboard. Unplug your mouse if you tend to forget and find yourself switching back and forth.*

*Sixth: Assistive technologies test: Repeat steps 1 and 2, using a screen-reader (NVDA, Jaws, and/or Voice-Over). Turn off your monitor if you tend to forget and find yourself using your vision to supplement the cues you are hearing. Note: it is not expected that every dev or every QA resource would do this. More efficient for 1 or 2 people to invest the time to become more skilled with screen readers' shortcut keys, etc., than for everyone to do so. But everyone is welcome to do so, using NVDA (free download for Windows), JAWS (free timed download for Windows) and Voiceover (comes free on the MAC). JAWS still has market share among screen-reading users.*

## **Checkoff**

Functional Analysis Complete? **No (completed by SME)**

Needs Review by KAI? **No (completed by SME)**

Technical Analysis Complete? **No (completed by DM)**

Needs Review by KTI? **No (completed by DM)**

Estimate: **30 hours (completed by DM)**

Technical Design: [Link Here](#) **(completed by DM)**

Jira: [KULRICE-6672](#) **(completed by SME)**

Final Documentation: [Link Here](#) **(completed by DM)**