Client's UIX
Release Notes
Production release on 2 March 2022
Version 1.0

Clients UIX

Contents

Client's UIX

Purpose

This document provides an overview of the enhancements, bug fixes, and new features that have been added in the new version.

Scope

This document is intended for the users to be familiar with the changes, bug fixes, enhancements, and new features that have been added in the new version.

Enhancements

The below table shows the improvements that have been made to the existing version:

Ticket	Summary
On opening/closing any popup, the page gives a shifting effect because the scroll bar hides/shows on opening any popup	Enhanced to avoid the earlier shifting effect. Now, the user can see a scroll bar on the main page when a pop-up is opened and vice-versa when it is closed.
The tooltip text for Aberrant ACE % bar, when already opened, should display "Click to see Less", instead of "Click to See more"	Enhanced the tooltip text. Now, the user can see the correct tooltip text on Aberrant ACE % bar as "Click to see less" when it is opened and vice-versa when it is closed.

Bugs

The below table shows the various bug fixes that have been made to the existing version:

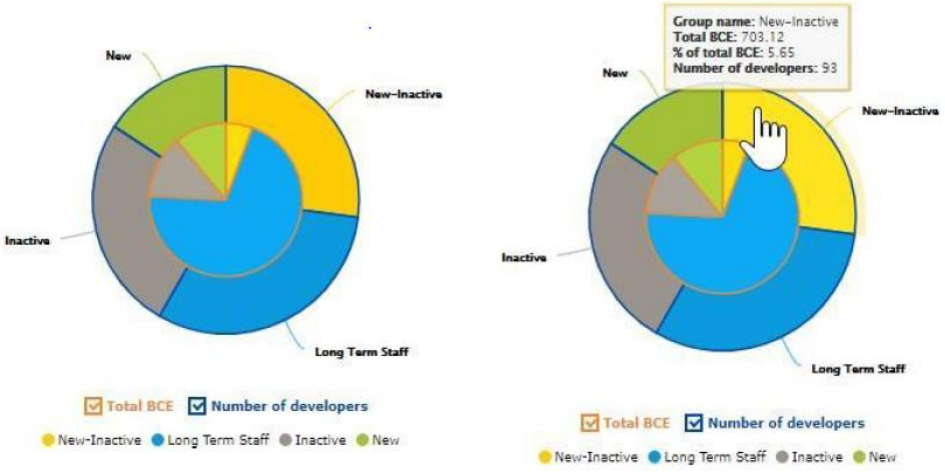
Issue	Description
Unable to see the updated details on the ‘History of changes to role' page	Fixed the issue. Now, when the user updates page access for a role, it gets reflected on the 'History of changes to role' page
Unable to export files after editing a developer	Fixed the issue. Now, the user could edit the developer on the manage developer page even though the developer was already edited.
Unable to scroll down after 100 entries on DCH page	Fixed the issue. Now, the user can scroll down after 100 entries on the DCH page.

New Features

A new feature has been added that allows the user to see the analysis of the attrition of developers in the Client's software estate using a Pie-chart comparing Total BCE and number of developers.

This chart provides information corresponding to different groups namely New, New-Inactive, Long-Term Staff, and Inactive represented with green, yellow, blue, and grey colors. The second chart gives a detailed explanation of its respective group name, values of Total BCE, % of Total BCE, and the number of developers whenever a user hover-over mouse on any individual group name.

Overall, for total BCE, the new group made the least contribution, while the long term staff made the highest contribution. In comparison, for the number of developers, the New-Inactive group made the least contribution, while the long term staff made the highest contribution.



Glossary

The below table provides definitions of terms that came across this document:

Term	Definition
Aberrant ACE	Developers' Coding Effort that compromises stability and maintainability of the software being developed. This is calculated based on amount of source code change (in terms of Coding Effort) that breaches thresholds defined in relation to the norms in the distribution of measures of code files for your organization. Expressed as %, which is the proportion of Aberrant ACE.
ACE	Actual Coding Effort: The amount of intellectual effort, expressed in hours, that software developers expend productively changing code, calculated from 36 base measures that evaluate three domains of coding activity; Volume, Complexity and Interrelatedness.
BCE	Billable Coding Effort: The Coding Effort delivered across working days (i.e. adjusted for weekends and holidays) that is billable to the organization. This deliverable is capped at 5 hours within a day, with the remainder of the time in a day recorded as Billable Non-Coding Effort. When developers "store up" change and commit it in large chunks to source code repositories, the Billable CE is pro-rated back to the preceding days where the developer was present but not active in the source code repositories.
Client ART	Analysis of Relative Thresholds (ART) is Client's approach to measuring software quality; specifically software stability and maintainability (how easily another developer can set about working on that code). The purpose of the measure is to enable software development managers to identify the most likely causes of maintainability and stability issues. ART is expressed as % of Aberrant ACE
Tenure	The time period in days from when a developer first made revisions in the code base to the time of the last revision in the code base by a developer.
UIX	User Interface XML