

Metrics Reporting Plan for Defects

Summary

This plan is designed to show the defects that have arose throughout the sprints in the project and the current status of said defects. It shows us what are project is missing from previous sprints.

Collected Metrics Measurement

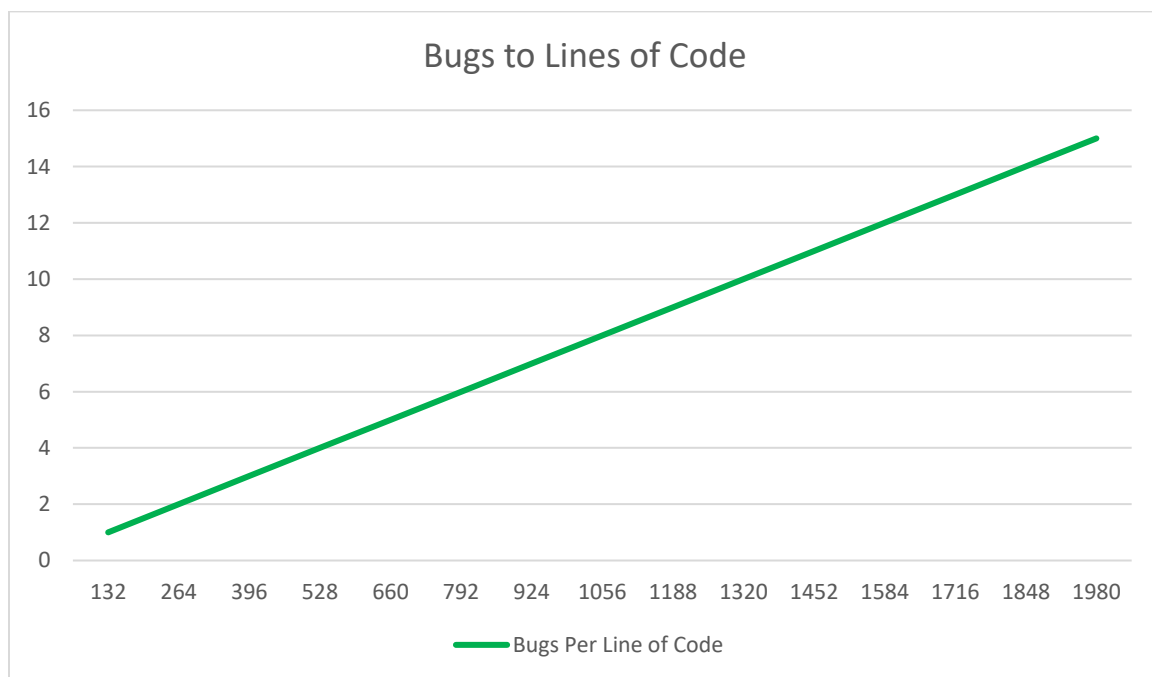
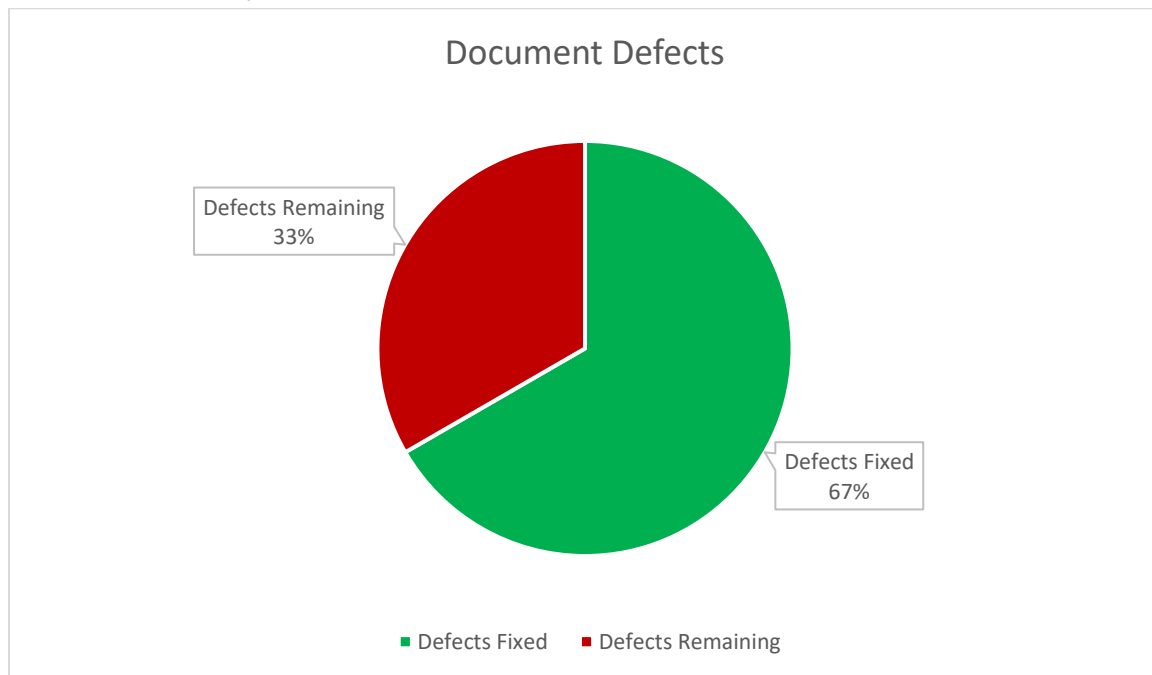
SPR#	Document Defect Description	Status
SPR-1	Incorrect link location	
SPR-1	Unspecified prototype candidate	
SPR-1	Document Versioning missing	
SPR-1	Coding standards missing	
SPR-1	Meeting schedule missing	
SPR-1	V&V document missing	
SPR-1	Requirements specifications missing	
SPR-2	V&V document missing	
SPR-2	Submitted incorrectly	
SPR-3	GUI images missing	
SPR-3	Backlogs missing	
SPR-3	Code doesn't run	
SPR-3	No readme.txt	
SPR-4	Missing backlogs	
SPR-4	Missing updated class diagram	
SPR-4	Missing User Acceptance Testing	
SPR-4	Missing Functional/Non-functional Testing	

ID#	Bug Description	Change Description	Status
1	-The GUI currently functions, but the data being operated on (employee objects) is currently hard coded into the program. All new data, changes, etc. is lost when program closes.	-Add JSON text files as databases so we can add, edit, and retrieve records for employees, timecards, and receipts. -Add modules for the employee class that will add, edit, and retrieve records from the employees JSON file.	
2	-Tkinter message boxes aren't being imported when the program loads. (Only for development in windows)	-Include the specific library that applies to Tkinter libraries.	
3	-The GUI page for payroll did not have the run payroll or run csv file buttons functional.	-Add a CSV and handling module for the imported csv's that need to calculate the final pay for hourly, and compensated employees. -Add JSON text file for a database as a backup and add accessibility functions for this database in a module.	

4	-The ID was being keyed off of the number of employees in the database. With testing, and perhaps future functionality, deleting employees would cause conflicts with newly created employee ID's.	-Whenever a new employee is added, their Id is one plus the last employee in the database ID.	
5	-The user would have been unable to access the program as there was no user information provided when the program was loaded onto a new computer.	-Included a readme.txt file that included information regarding basic login for admin. From there, the user would be able to look up other employee information and create / edit employees as necessary.	
6	-There was trouble for the shareholder originally with starting up the GUI. They were trying to launch the program from an incorrect file. File naming isn't explanatory.	-There were some irrelevant files that existed in the next version of the program that could have caused more confusion. These were removed, and the file that the program needs to be launched from was renamed to main.py for clarity. This was then mentioned in the readme.txt file.	
7	-There were issues caused by the validation function when it tried to update the field-specific error messages.	-The issues were caused by a logic error in the validation function attempting to alter label widgets that had not been created. The issue was solved by including a try/except block that checks for widget presence before alteration.	
8	-The error messages on all GUI pages were hardly noticeable.	-Solved by implementing a specific style with larger, more colorful text for error messages.	
9	-All fields were editable in the edit page, including things that realistically should not be changed, even by an admin (e.g. employee id, start date, etc)	-Solved by enabling readonly mode on select entry boxes. -Admins can change everything except employee id, start date, end date, and employee status, while general permission employees editing themselves can change only personal information and not anything related to the company like their pay and job title.	
10	-There were issues attempting to set the entry boxes of the view page to "readonly" when in general view	-Issue was caused by a logic error in the method meant to set all entry boxes to "readonly". It would change the widgets created by the admin view, which included many not used for the general view. Issue was solved by creating a second list to be passed to the readonly method when not in admin view.	

11	-The shareholder did not approve of the ability to add timecards and receipts individually	-Solved by reverting the pay page of the GUI back to its initial state with only the ability to import the timecard and receipt csv files.	
12	-The search page on the GUI was not very user-friendly, requiring prior knowledge of the database to operate easily.	-Implemented a listbox widget below the search bar that shows all the employees in the database that match the current contents of the search bar.	
13	-There were issues moving from the search page to the view page of the searched employee when logged in under general permission.	-Issues were caused by the program attempting to alter widgets that had not been created (they are created only in admin view or when viewing oneself). Solved the issues by placing the code that altered those widgets within an if statement that checks whether they exist or not first.	
14	-The pages of the GUI would not correctly refresh/update when switching between pages (e.g. old data on the add page would remain instead of being cleared).	-Created a new method for the Application class to be called with every page change that destroys and then recreates the pages to allow them to update properly.	
15	-There was trouble getting the entry boxes of the view page to correctly update their contents/configurations for inheritance with the add and edit pages.	Moved the code for creating the entry boxes out of the constructor for the View_Page class and into a method that can then be called by that class or any of its children as necessary.	

Data Summary



Report Analysis

Summary

- 2/3 of the defects from previous sprints have been fixed, but 1/3 of defects still need to be addressed
- there is an average of 132 lines of code for every bug
- all bugs have been addressed and fixed

Future prediction of current project

- Major defects will be fixed but minor defects will be left alone until they become important enough to cause problems
- Bugs will continue to be dealt with as the code is wrote

Possible improvements for Future

- Address defects in the sprint that they are found in
- Document dates of bugs