Abstract

"An ounce of prevention is worth a pound of cure." In software, these expressions translate into the common observation that the longer a defect stays in process, the more expensive it is to fix [10]. Moreover software defects are expensive and time consuming. The cost of finding and correcting defects represents one of the most expensive software development activities. And that too, if the errors get carried away till the final acceptance testing stage of the project life cycle, then the project is at a greater risk in terms of its Time and Cost factors. A small amount of effort spent on quality assurance will see good amount of cost savings in terms of detecting and eliminating the defects.
To gain a deeper understanding of the effectiveness of the software process, it is essential to examine the details of defects detected in the past projects and to study how the same can be eliminated due to process improvements and newer methodologies. This paper will focus on finding the total number of defects that has occurred in the software development process for five similar projects and aims at classifying various defects using first level of Orthogonal Defect Classification (ODC), finding root causes of the defects and use the learning of the projects as preventive ideas. The paper also showcases on how the preventive ideas are implemented in a new set of projects resulting in the reduction of the number of similar defects.

Reference

- Linda Westfall, Defect Density http://www.westfallteam.com/Papers/defect_density.pdf

Index Terms

Computer Science
Software Process
### Key words

<table>
<thead>
<tr>
<th>Defect</th>
<th>Defect Analysis</th>
<th>Defect Prevention Root Cause</th>
</tr>
</thead>
<tbody>
<tr>
<td>Analysis</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Analysis