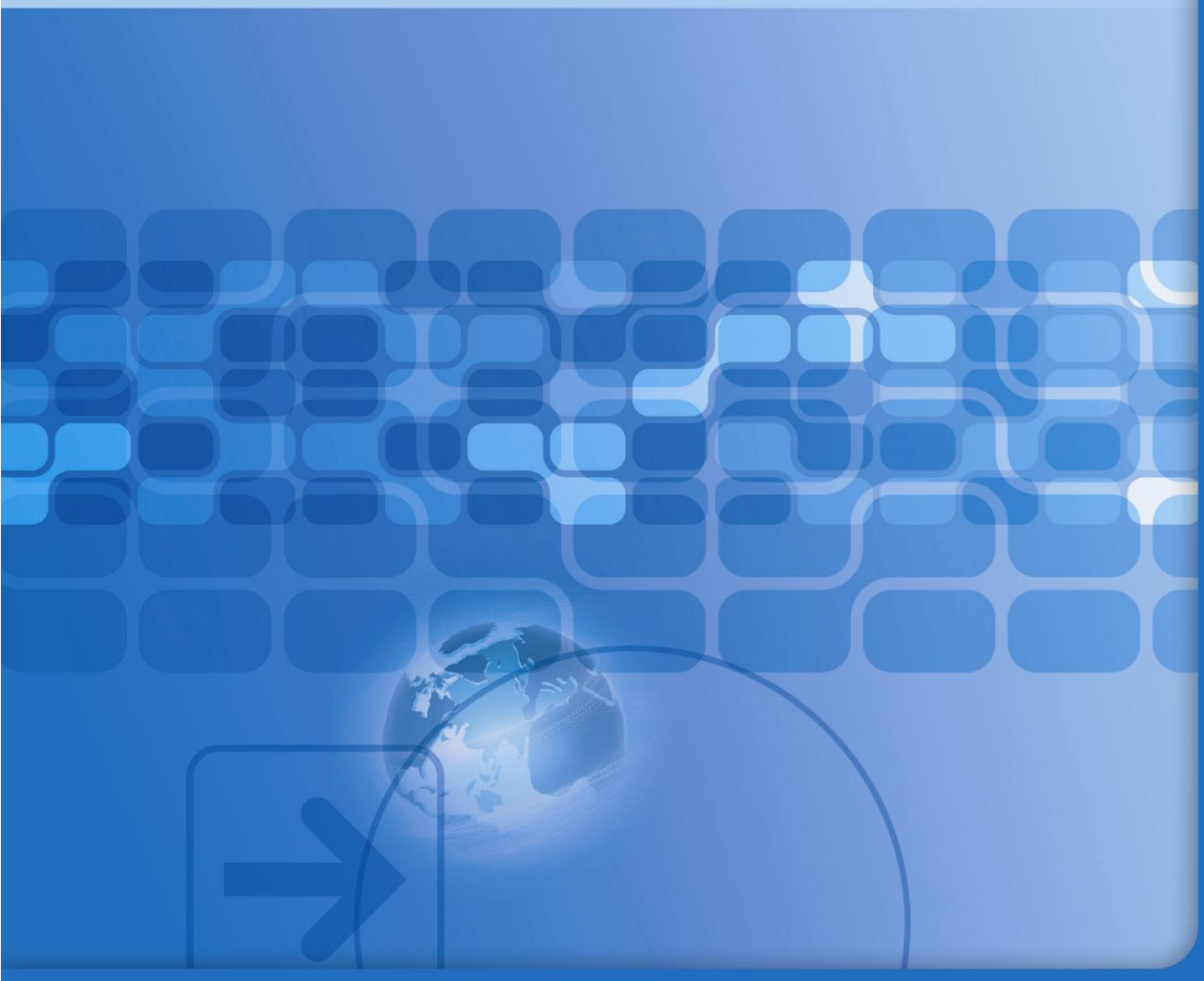
Microsoft

About the Bug Trends Report



patterns & practices

Information in this document, including URL and other Internet Web site references, is subject to change without notice. Unless otherwise noted, the example companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted herein are fictitious, and no association with any real company, organization, product, domain name, e-mail address, logo, person, place, or event is intended or should be inferred. Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Microsoft Corporation.

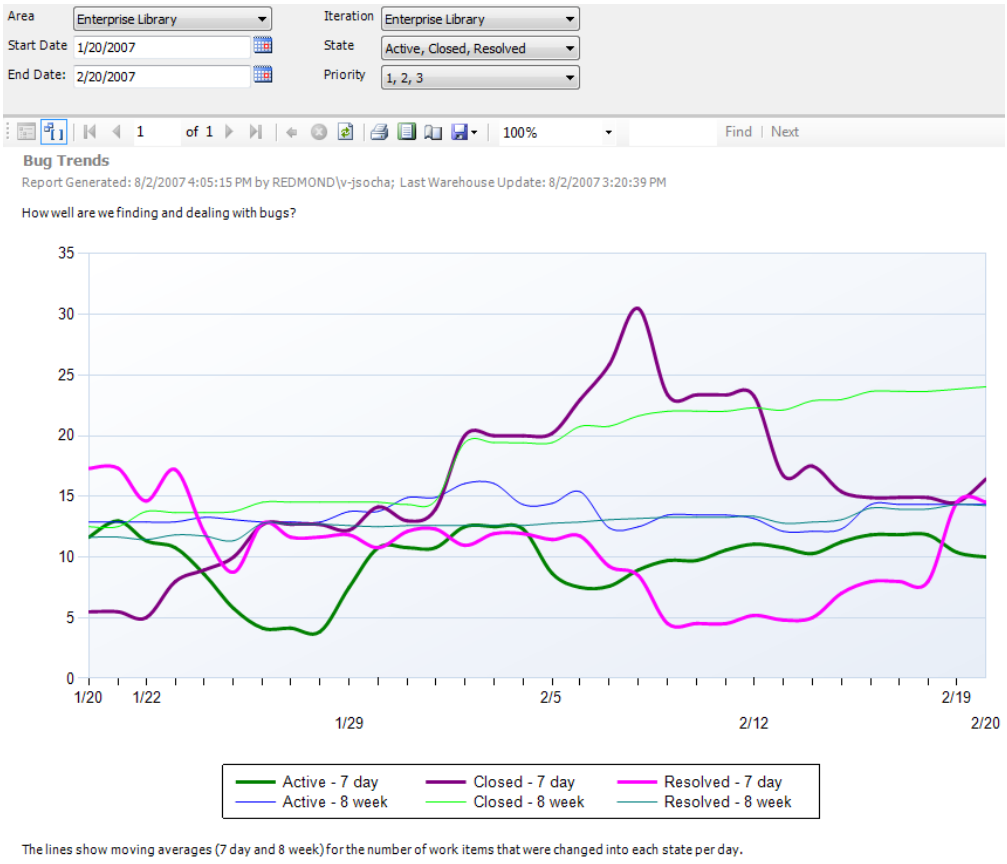
Microsoft may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Microsoft, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

© 2007 Microsoft Corporation. All rights reserved.

Microsoft, MS-DOS, Windows, Windows NT, Windows Server, and Visual Studio are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

The names of actual companies and products mentioned herein may be the trademarks of their respective owners.

About the Bug Trends Report



About This Report

Title: Bug Trends

Version: 1.1

Answers: How well are we finding and dealing with bugs?

How to Use This Report

Healthy projects show an occasional rise in the long term bug discovery rate ("Active – 8 week"), followed by longer periods in which it is exceeded by the resolution rate ("Resolved – 8 week"). As the project progresses, periods in which the resolution rate exceeds the discovery rate should predominate, showing that the software is stabilizing over time.

Notice that this report does not directly show the number of remaining defects. For that information, you should look at the Remaining Work report, with the work item type set to **Bug**.

Parameters

Start, **End Date**	Choose the range of dates you want the report to include.
Area, **Iteration**	Choose the scope you want the report to include.
Priority	Choose the priorities of bugs you want the report to include.
State	Choose the state lines you want the report to show on the diagram.

Methodology Requirements

This section describes the assumptions behind this report that might impact the way you use and update information in work items.

This report assumes the team is using the **Bug** work item type to track defects in the software. For this report to be relevant, the team needs to create a new **Bug** work item for each defect found, and accurately track the progress of work against those work items by updating the state of each bug (**Active**=newly found issue, **Resolved**=development has fixed or addressed the issue and it is ready for test, **Closed**=tested and fix verified, or other resolution accepted).

Fields and Values Assumed

We've tried to minimize the number of assumptions and requirements for using this report in your own projects. The following table describes the dependencies with fields and/or values in work items.

Field	Assumptions
Work Item Type	Bug—This report shows only the results for the work item type **Bug**. You will need to modify this report if you have a different name or more than one bug work item.
State (Work Item.State)	All—Initially, all states are shown. You can use the **State** parameter of the report to show a subset of states.

Additionally, this report shows the results as rolling averages, using 7 days and 8 weeks for the two sets of lines shown. The next section describes how to customize these values in the report.

Customizing the Report

The RDL file is ready to install in a TFS project.

You can customize this report using the Report Designer, which is a set of tools hosted in Visual Studio after you install the SQL Server client tool set named Business Intelligence Development Studio.

Modifying the RDL to Work Inside Report Designer

You can create a Report Server project to work on this report. After you finish that, you'll need to make the following changes before you can customize the report:

1. Open the RDL file in Visual Studio's Report Designer.
2. Click either the **Data** or **Layout** tab.

3. On the **Report** menu, click **Report Parameters**.
4. Click **ExplicitProject**, and then clear the **Hidden** check box.

The **ExplicitProject** parameter is usually hidden, but during development you'll want it visible so you can type a project name and then press the TAB key to initialize the other parameters using that project.

You can also provide an explicit project name so you don't have to type it each time during development. To do this, click the **Non-queried** option button in the **Default values** area, and then enter the project name in the text box for the **ExplicitProject** parameter. After you do this, you can hide the **ExplicitProject** parameter, even during development.

We used "Enterprise Library" as the initial project in each dataset, area, and/or iteration when we developed this report. You'll want to select a project on your TFS server so you can run the queries and see the results.

Changing the Bug Work Item Type

This report assumes the work item type used for bugs is named **Bug**. If you've either changed this name, or if you have more than one work item type for bugs, you'll need to change some of the queries. This section describes how to make those changes.

Before proceeding with the changes, do the following:

1. In Visual Studio, open the RDL file for this report.
2. Click the **Data** tab to start working on the queries.

dsWorkItemHistory

3. In the **Dataset** combo box at the top of the window (just below the **Data** tab), click **dsWorkItemHistory**.
4. Click **{ Bug }** in Work Item row and Filter Expression column.

Dimension	Hierarchy	Operator	Filter Expression	Parameters
Team Project	Team Project.Team Project	Equal	{ Enterprise Library }	☑
Work Item	Work Item.Work Item Type	Equal	{ Bug }	☐
Date	Date.Date	Range (Inclusive)	1/1/2007 : 2/1/2007	☑ ☑
Work Item	Work Item.State	Equal		☑

5. In the dialog box that appears, select the check box next to each work item type that you use as a bug.

dsStates

6. To change the **{ Bug }** filter to the correct set of work items, repeat the preceding steps.

dsPriority

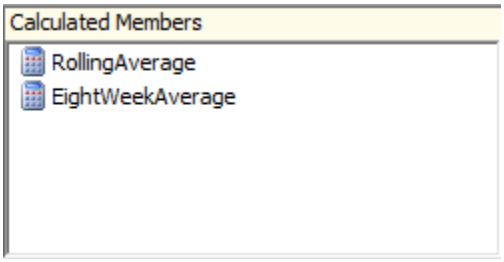
7. To change the **{ Bug }** filter to the correct set of work items, repeat the preceding steps.

Changing the Time Period for Rolling Averages

The report is initially set up to use 7-day and 8-week rolling averages. These are both easy to change to other values:

1. In Visual Studio, open the RDL file for this report.
2. Click the **Data** tab to start working on the queries.
3. In the **Dataset** combo box at the top of the window (just below the **Data** tab), click **dsWorkItemHistory**.

At the bottom of the window, there is an area for calculated members that looks like the following:



The first calculated member, **RollingAverage**, creates the data for the 7-day rolling average, and the second calculated member, **EightWeekAverage**, is for the 8-week rolling average.

To change either calculated member, do the following:

1. In this area, double-click the name you want to change. The **Calculated Member Builder** dialog box appears. For **RollingAverage**, the expression will look something like the following:
   Avg([Date].[Date].CurrentMember.Lag(6): [Date].[Date].CurrentMember, [Measures].[State Change Count])
2. In the **Lag** function, change the number to one less than the number of days you want to average (the current day is 0 days before, so 0..-6 covers a week).
3. Click **OK**.