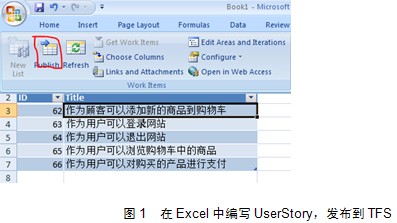
**Visual Studio 2010在敏捷开发中的实践应用**

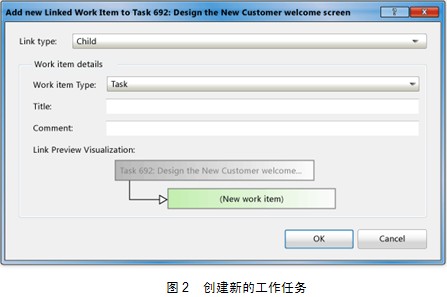
发布时间：2010.09.05 15:25      来源：赛迪网     作者：段君毅

【赛迪网报道】从事敏捷开发，我们首先要熟悉UserStory的编写。通常情况下我们在项目初期都在Excel中写好UserStroy，排好优先级别，通过TFS来管理的时候，需要把这些UserStory变成一个个WorkItem放进来。那么团队人员就会面临这样一个问题，当用Excel计划好大量的userstory后能否快速地将这些内容变为WorkItem。答案是肯定的，VS 2010为我们提供了方便的功能来操作Excel表格，如图1所示。



在创建新的userstory时候，除了要填写上面的一些主要内容以外，在implementation栏目中需要填写实现这个UserStory的任务，如开发任务和测试任务等。

VS 2010给我们提供了非常方便的可视化操作，可以选择新建任务，如图2所示。



还可以选择将现有的任务连接到这个UserStory上，如图3所示。



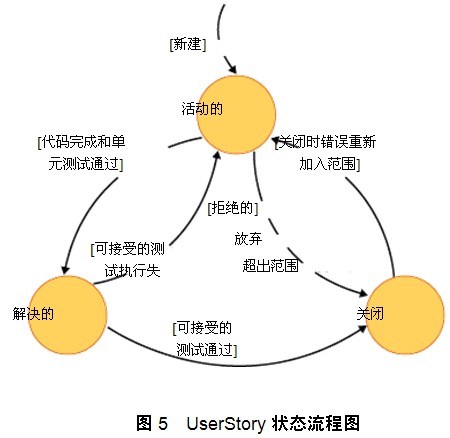
同样，在TestCase项中可以选择新建和关联TestCase、Bug等到这个UserStory中。

另外，还可以上传相应的附件和创建超连接，如图4所示。



（2）UserStory流程

在UserStory中我们可以通过设置其状态来跟踪其流程，主要状态的关系如图5所示。



如图3-32所示，在这些状态中主要由Active、Resolved、Closed组成，为了更清晰地说明什么状况下设置相应的状态，我们对它一一说明如下。

①Active状态

a. 新建

UserStory被团队成员创建完成后，VS 2010将会将自动获得以下数据：

λ Created By：创建UserStory的成员；

λ Created Date：创建时间（服务器时间）。

b. 从Active 到Resolved状态

我们可以将一个Active状态下的UserStory设置为Resolved状态，当完成该UserStory所有的代码编写工作并其单元测试已经通过，这样的话，该UserStory才能正确进入下一个工作流程。

当设置成为Resolved状态时，系统会自动获得以下项的数据：

λ Resolved By: 完成该工作项目的成员；

λ Resolved Date: 完成时间；

λ State Change Date工作项状态变更时间。

c. 从Active 到Closed状态

事实上，当我们遇到下列三种情况的时候该UserStory应该从Actived状态变更到Closed状态：

λ Rejected：认为该UserStory是没有用处的，不符合当前的需求，没有价值；

λ Abandoned：该UserStory被认为没有必要实现；

λ Out of Scope：该小组没有足够的资源来实施对用户的当前迭代的故事。用户故事可能被确定为超出范围，因为团队没有足够的时间去识别风险阻止问题出现。同时需要更新迭代状态，如果放在下一个版本中，那么需要留出空白，填写变化原因说明。

当设置成为Closed状态时，系统会自动获得以下项的数据：

λ Closed By: 关闭工作项的成员名称；

λ Closed Date: 工作项关闭日期；

λ State Change Date: 工作项状态变革日期。

②Resolved状态

当UserStory的代码实现已经全部完成后，开发组长可以将其状态设置为Resolved状态，分配给测试人员，以便开始进行测试工作。

a. 从Resolved到Closed

当相应的UserStory所有的测试用例都通过后，可以将其设置为关闭状态，同时将该UserStory分配给产品组长。同样当设置成为Closed状态时，系统会自动获得以下项的数据：

λ Closed By：关闭工作项的成员名称；

λ Closed Date：工作项关闭日期；

λ State Change Date：工作项状态变革日期。

b. 从Resolved到Active装

当遇到至少一个Userstory的测试失败后，需要将其状态设置为Active，同时这个UserStory将会指派给开发组长，测试人员也必须为测试失败创建Bug，当UserStory被重新激活时，系统会自动获得以下项的数据：

λ Activated By：激活工作项的成员名称；

λ Activated Date：工作项激活日期；

λ State Change Date：工作项状态变更日期。

③Closed状态

通常Closed状态作为一个UserStory的最终归宿，是不能随便变更状态的，但是当范围变更时，业务分析师和项目经理可以重新激活一个关闭的UserStory。

当项目遇到以下情况时候，可以把UserStory状态从Closed 设置到Active状态：

λ Reintroduced in Scope：有可用资源去实现UserStory,需要明确的是实现任务、测试用例以及UserStory的详细描述都必须完成；

λ Closed in error：分配给该UserStory的任务、测试用例或者Bug等没有被关闭的时候。确保执行任务、测试用例，并且为用户故事细节定义明确，足以支持其发展。

当一个UserStory被重新激活时，系统会自动获得以下项的数据：

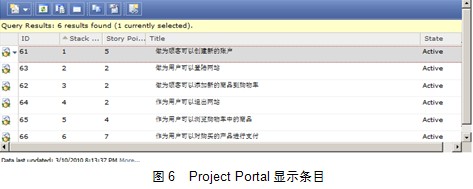
λ Activated By：激活工作项的成员名称；

λ Activated Date：工作项激活日期；

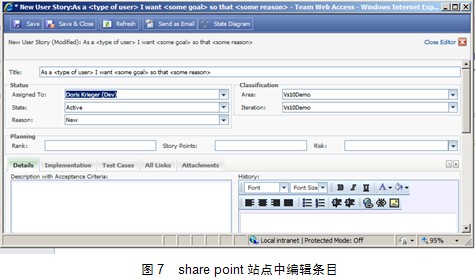
λ State Change Date：工作项状态变更日期。

从上面我们可以看到，UserStory在过程中存在一定的生命的周期，当日期被设置关闭后，这个Userstory也就相应地完成。在VSTS中把UserStory最为WorkItem的一种类型，所以它的工作流程必然和所有的WorkItem一样受TFS的管理，同样也必然能够通过TFS来获取相关状态信息，在下面“白板和报表”一节中我们将看到在项目工程中如何通过VSTS和TFS来获得所有工作任务的状态和报告，以帮助项目团队监控项目，为项目更好地执行提供帮助。

另外在TFS中，我们也可以通过Web的Sharepoint 站点来管理我们的Product Backlog和UserStory，如图6所示。



在share point的Project Portal中，我们可以对该story进行编辑等操作，如图7所示。

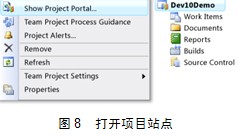


3、白板和报表（Dashboards and reports）

白板和报表作为Agile开发的一个重要组成部分，往往能够非常清晰地反映当前项目的状况。在项目团队中会被大量使用，在VISUAL STUDIO 2010和TFS中为我们提供一个电子化的白板和报表服务来帮助我们更好地管理和监控项目。

（1）白板（Dashboards）

在项目执行过程中，通常我们通过白板展示项目目前的状态来对项目进行跟踪。TFS2010在项目创建的时候就为我们创建了一个Sharepoint的项目站点，项目成员通过在Visual Studio 2010的TeamExplorer中单击鼠标右键就可以浏览项目站点了，如图8所示。



打开项目站点后，就会看到很多由WebPart编写的DashBoard被展现在项目站点中，如图9所示。

这样便降低了采用流程的门槛，从而可以自动收集跨功能的项目衡量标准，而无需与手动报告有关的开销。

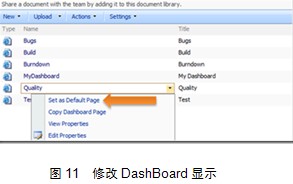


提示

实际上我们可以通过网站左侧菜单来设置默认显示的Dashbaord，如图10所示。

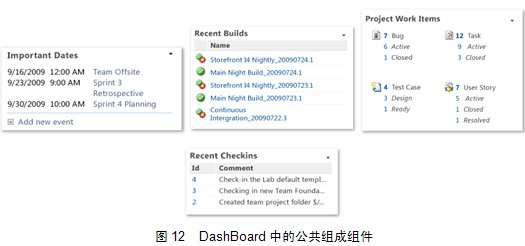


单击左侧连接，将质量作为默认显示页，如图11所示。



TFS还提供了丰富的自定义功能来管理这些WebPart，如你可以使用WebPart来从TFS中提取数据，可以将Vistual Studio中管理ALM的数据，进行分析后展现出来，也可以将工作项、工作任务展现出来。

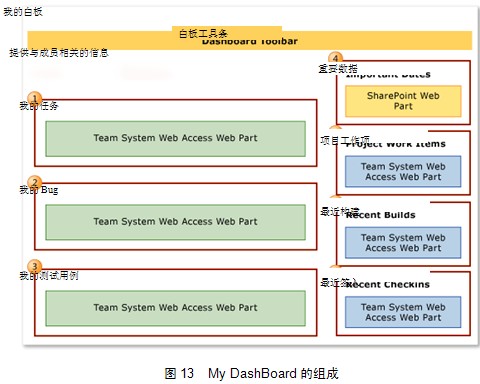
在DashBoard中默认会列出很多与项目相关的项，其中包括项目的重要事件信息、项目的工作项、Build信息和当前Check-in的信息，如图12所示。



在TFS中将DashBoard分为如下几种类型。

① 队成员自己的工作白板（My DashBoard）

可以通过这个白板（My DashBoard）快速访问自己需要完成的工作项和监控工作任务，它的组成结构如图13所示



由图3-40可见，通过以上这些WebPart来组成了我的工作模板，除了以上我们提到的一些对于项目的公共webpart组件外还包含了以下内容。

a. 我的任务（My Tasks）

列出了团队成员自己当前没有关闭的工作任务。

b. 我的Bug（My Bugs）

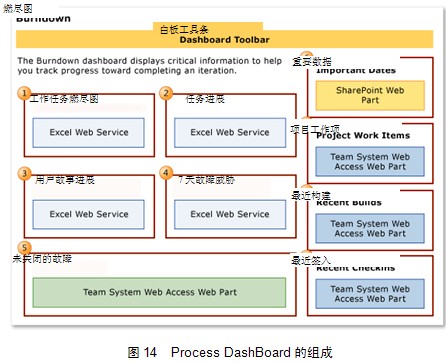
当由测试人员提出Bug以后，TFS就会发送这个Bug信息到这这里，团队成员就会很容易的看到当前正处于活动状态的Bug。

c. 我的测试用例（My TestCase）

列出了没有关闭的当前用户的测试用例。

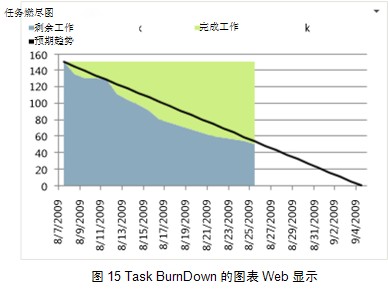
② 过程白板(Process Dashboard)

从本质上讲，TFS提供的该Dashboard实际是就是在Scrume中的燃尽图展示（burn down chart），我们可以通过图14来看它的组成部分。



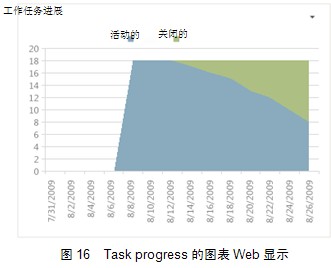
a. 工作任务燃尽图（TaskBurnDown）

该燃尽图通过Excel web services展示。它主要体现了当前工作任务的状态，通过可视化的展示方式来显示团队所有工作任务需要的总时间（单位以小时计），以及过去几周的工作成果，如图15所示。



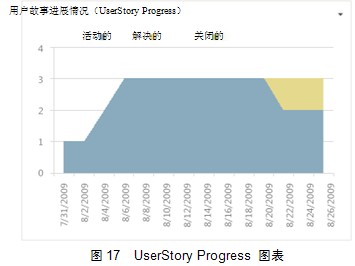
b. 工作任务进展（Task Progress）

该部分可视化展示了过去几周所有的工作任务状况，如图16展示了过去四周的工作任务的状况。



c. 用户故事进展情况（UserStory Progress）

通过对UserStory的状态分组，可视化地列出了四周内所有UserStory的状况，如图17所示。



通过上面的图表可以清晰地列出，活动的、已经解决的和关闭的UserStory。本例中还没有UserStory被关闭，大多数还在活动状态，也就是说，目前我们的项目还有很多内容没有实现。

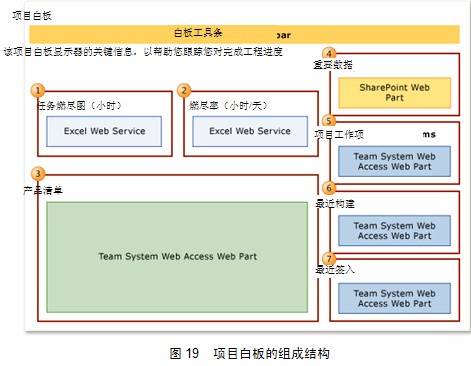
d. 四周内基于7天周期的风险威胁率趋势展示，如图18所示（表示7天为一周期风险对项目可能会造成的影响趋势图）。



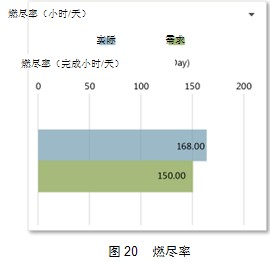
图19的曲线表示我们在过去四周内所有的风险数，以7天为一个周期的状态趋势展示，可以为我们以后的项目执行提供风险预测参考。

③ 目白板（project Dashboard）

项目白板包含了项目所有的信息展示，其中包含了燃尽图和产品清单，如图19所示。

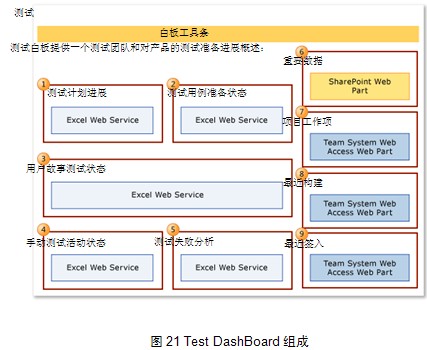


其中燃尽率（Burn rate）：表示当前项目团队的工作状态，即实际工作和需要完成的工作的比例，如图20所示。



④ 试白板（Test Dashboard）

在测试白板中，会给出与测试相关的所有内容，大部分是通过ExcelWeb Services提供的数据报表，它的组成如图21所示。



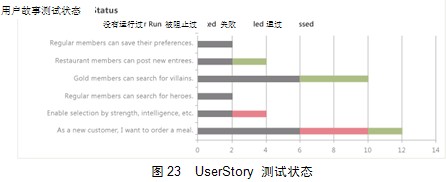
其中：

a. 测试用例准备状态（Test Case Readiness）：最近四周内测试用例的准备状态，如有多少测试用例已经被测试团队定义、有多少测试用例已经可以准备运行、还有多少测试用例需要继续编写、所有的测试用例是不是能够在迭代结束之前成功运行结束等，都可以通过如图22所示展示出来。



b. 用户故事测试状态（UserStoryTest Status）：UserStory相关的测试用例执行情况，包括以下几种状态：Passed (green)、Failed (red)、Blocked (purple)、Not Run (gray)，如图3-50所示。

根据这个图23，会跟踪我们原来在编写UserStory时候关联的TestCase执行的状态。



c. 测试执行失败分析（Test Failure Analysis）：执行失败的测试用例分析，如图24所示。

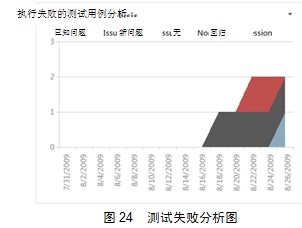


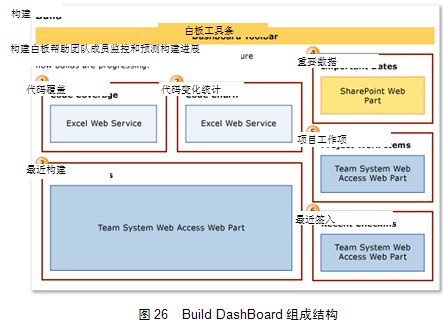
图25为我们展示了在执行测试的时候，除了测试失败，我们的人员会将测试失败的状态进行分析，给出结果。如果一个用户登录功能测试失败，那么经过分析是因为某某程序错误，那么就可以将其标为Know Issue。

d. 手动测试激活状态（Manual Test Activity）：手动测试用例的执行状态，如图25所示。



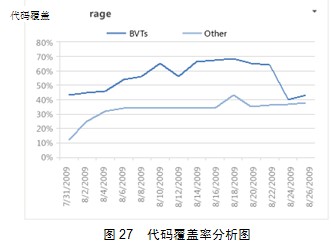
图3-52显示了在四周内进行的手动测试，因为在VISUAL STUDIO 2010和TFS2010中引入了手动测试的功能，所以在这里可以通过图表对其进行分析。

⑤每个Build的白板（Build Dashboard），呈现Build创建过程中的信息。组成结构如图26所示。

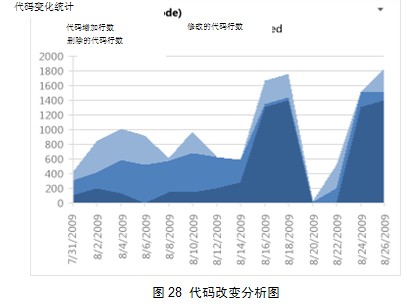


其中：

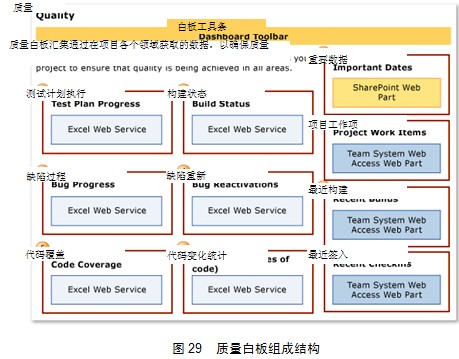
a. 代码覆盖（Code coverage）：描述了四周内通过BVT和其他测试方式测试的代码的百分比，如图27所示。



b. 代码变化统计（Code Churn）：描述团队成员最近四个星期内对代码进行的添加，删除更改的行数，如图28所示。

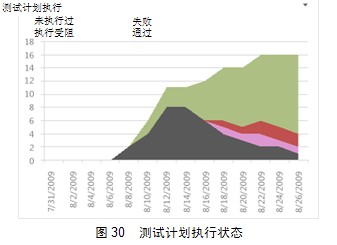


⑤ 量白板（Quilty Dashboard），对于质量管理来讲，默认提供的白板中包含了我们在build状态部件、代码变动和代码覆盖率部件，以及部分测试部件，组成结构如图29所示。

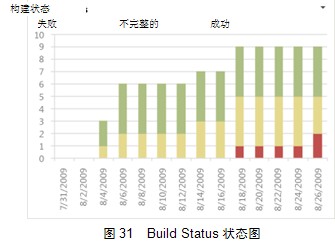


其中：

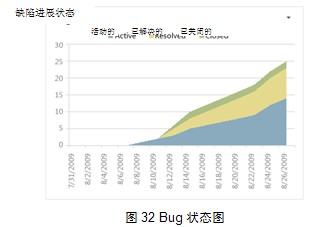
a. 测试计划执行（Test Plan Progress）：显示所有测试结果用例执行得结果，通过状态分组：Never Run、Blocked、Failed和Passe，如图30所示。



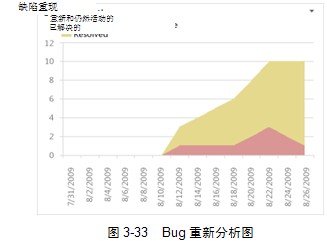
b. 构建状态（Build Satus）：最近四周内创建成果和失败的Build，如图31所示。



c. 缺陷进展状态（Bug Progress）：最近四周内所有的Bug状态，如图32所示。

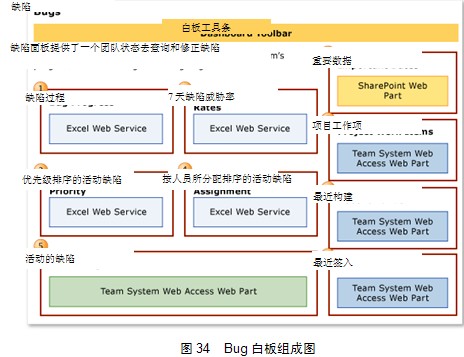


d. 缺陷重现（Bug Reactivation）：列出所有重现的Bug，如图33所示。



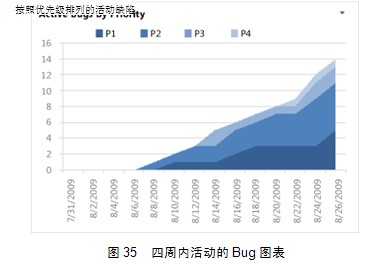
e. 还有上面介绍过的代码测试覆盖率和代码修改行数统计。

⑦Bug白板（Bug Dashboard），在这个白板中除了上面质量白板中包含的Bug状态图部件以外还包含很多与Bug相关的部件，如图3-34所示。



其中：

a. 按优先级排列的活动缺陷（Active Bugs by Priority）：按照优先级列出所有活动的Bug，如图35所示。



b. 按指派人员排列的活动缺陷（Active Bugs by Assignment）：根据优先级分组列出每个分配在团队人员头上的活动Bug，如图36所示。

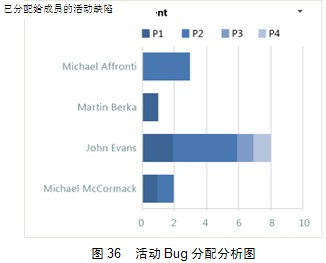
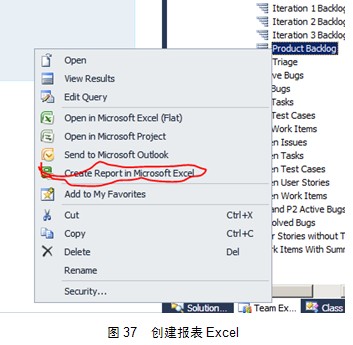


图36中纵轴列出了开发人员，横轴表示为活动Bug的数量，用来显示目前活动Bug中谁的Bug多。

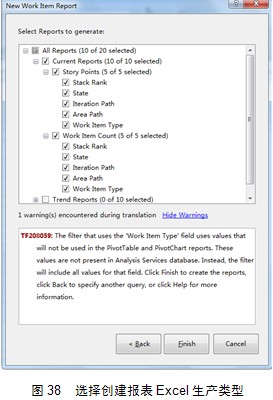
（2）报表

在TFS2010中通过Excel和Crystal report两种方式来实现项目报表展示。通过报表展示，我们可以获取项目当前状态的很多信息。

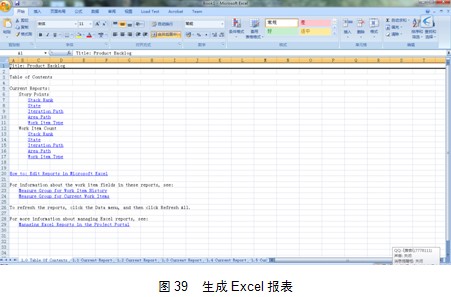
① 用Excel来创建报表，如图37所示。



下一步选择要创建报表的项，如图38所示。

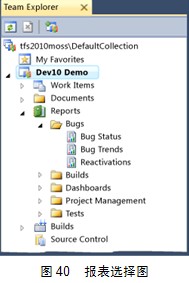


接下来就能生成报表，如图39所示。



对于报表来讲，我们可以生成我们想要的大多数报表，如可以生成燃尽图、Bug追踪、工作任务分配、状态监控等报表，在前面讲到的通过项目门户站点中列出的所有报表，我们基本上都可以通过TFS的Excel Services在Excel中展现出来。

② 外，还可以通过TFS为我们提供Crystal Report来展示相应的报表，如图40所示。

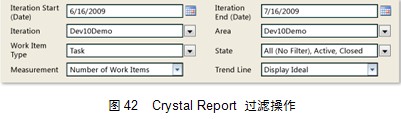


在图40的reports选项中我们可以有很多报表分类，而且都是采用Crystal Report来展示的。下面例举如下。

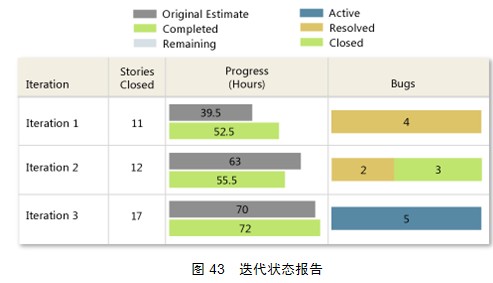
a. 燃尽图的CrystalReport展示，如图41所示。



可以通过CrystalReport提供的过滤工具对其进行过滤操作，如图42所示。



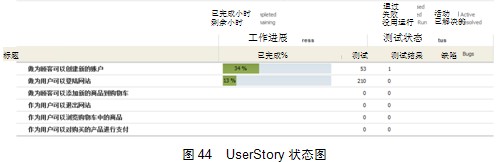
b. 迭代状态报表，如图43所示。



从这个报表可以直观地看出所有迭代执行的情况。可以判断迭代计划和执行是否正常。在本例中呈现的是一个状态良好的迭代报告。

c. UserStory图表，如图44所示。

TFS使用SQL Server Reporting Services生成相应的报告来展示项目的进度和质量。这些报告通过从工作项、版本控制、测试结果汇总数据，然后创建。



其中多数报告提供了过滤器，可以使用它进行报表内指定的内容的过滤。这些报表涉及用户的故事、测试用例、任务和错误的工作项等。对于项目团队来讲，一定要学会如何通过报表来搜寻有用的信息以帮助项目成功。