

معرفی:

امروزه تست کردن کدها به دلیل وجود ابزارهای مختلف زیادی، کار آسانی شده است. اما بعضی‌ها در web application ها، یکی از تست‌هایی را که خیلی هم مهم است را فراموش می‌کنند که آن هم تست UI است. شما را در این مقاله با یکی از روش‌های خوب تست UI آشنا خواهیم کرد. ابزارهای زیادی برای تست UI وجود دارد که کار کردن با آنها نه تنها زمان بر بلکه بسیار خسته کننده می‌باشند و به خاطر همین خیلی‌ها از انجام تست UI صرف نظر می‌کنند.

WatIn چیست؟

WatIn مخفف Web Application Testing in .Net می‌باشد؛ که یک فریم ورک تست web application ها است. WatIn این اجازه را به شما می‌دهد که با استفاده از IE ویا FireFox عناصر داخل صفحات را مقدار دهی کنید و یا حتی رویدادی را برای عناصر فراخوانی کنید.

شروع کار با WatIn:

در زیر یک نمونه از کار با WatIn را می‌توانید مشاهده کنید:

```
[TestMethod]
public void SearchForWatinOnGoogle()
{
    using (var browser = new IE("http://www.google.com"))
    {
        browser.TextField(Find.ByName("q")).TypeText("Watin");
        browser.Button(Find.ByName("btnG")).Click();
        Assert.IsTrue(browser.ContainsText("Watin"));
    }
}
```

WatIn یک فریم ورک کاربر پسند است و در ادامه متوجه می‌شوید که استفاده از این فریم ورک چه مزایایی دارد. برای نصب، WatIn را می‌توانید [از اینجا](#) دانلود کنید ویا اگر خواستید می‌توانید با NuGet هم این فریم ورک را دانلود کرده و نصب نمایید. برای شروع کار با Watin باید reference هایی را به پروژه تان اضافه کنید که یکی از این reference ها Watin.Core.dll می‌باشد و برای استفاده از IE ویا FireFox باید فضای نام Watin.Core را اضافه کنیم. Watin چند فضای نام دیگری را هم به همراه دارد که در زیر به توضیح مختصری از آنها می‌پردازیم:

1-Watin.Core.DialogHandlers: این فضای نام این امکان را به شما می‌دهد تا دیالوگ هایی را که مرورگر می‌تواند به کاربر نمایش دهد، مدیریت کنید. از handlerهای این فضای نام AlertDialogHandler, ConfirmDialogHandler, LoginDialogHandler و FileUploadDialogHandler, PrintDialogHandler می‌باشد.

2-Watin.Core.Exceptions: این فضای نام دارای یک سری exception می‌باشد و این امکان را به ما می‌دهد تا یک سری رفتارهای ناخواسته را کنترل کنیم. بعضی از این exception ها ElementNotFoundException, IElementNotFoundException, TimeoutException و WatinException می‌باشد.

3-Watin.Core.Logging: این فضای نام کلاس هایی را در اختیار ما می‌گذارد تا بتوانیم عملیاتی را که در کدمان انجام می‌دهیم log کنیم.

مثالی از watin که در بالا نشان دادیم به این صورت عمل می‌کند که مرورگر IE را باز کرده و به سایت google خواهد رفت. در این صفحه جعبه متنی یا TextBox با نام "q" را پیدا کرده و عبارت "Watin" را در آن تایپ می‌کند و همچنین Button ی با نام "btnG" پیدا کرده و آن را کلیک می‌نماید و در آخر بررسی می‌کند که در مرورگر متنی شامل Watin وجود دارد یا خیر. مشاهده کردید که به همین سادگی یک تست UI نوشتیم. به نظر شما جالب نبود؟ فرض کنید که اگر می‌خواستید با مثلاً Microsoft Test Manager این کار را انجام دهید چه دردسرهایی را باید تحمل می‌کردید. حالا تست UI برای همه برنامه نویسی‌ها جذاب خواهد شد.

به جای مثال بالا می‌توانیم به صورت زیر هم عمل کنیم:

```
[TestMethod]
public void SearchForWatiNOnGoogle()
{
    using (var browser = new IE("http://www.google.com"))
    {
        browser.TextField(Find.ByName("q")).Value="WatiN";
        browser.Button(Find.ByName("btnG")).ClickNowait();
        Thread.Sleep(3000);
        Assert.IsTrue(browser.ContainsText("WatiN"));
    }
}
```

تفاوت کد دوم با کد اول این است چون در کد اول از متد `TypeText` استفاده کردیم یک مقدار سرعت تست را پایین می‌آورد ولی اگر از `Value` و یا از `SetAttribute` استفاده کنیم دیگر عمل تایپ را انجام نداده و مقدار را مستقیماً در مقدار `TextField` قرار می‌دهد. شاید بپرسید چرا بعد از متد `ClickNowait` چند ثانیه صبر می‌کنم؟ چون صفحه برای اینکه بارگذاری شود و نتیجه جستجو را نشان دهد کمی طول کشیده و `Assert.IsTrue` شما `Failed` می‌شود. البته به جای `Thread.Sleep` می‌توانیم از متدهای مربوط به `Watin` هم استفاده کنیم مانند `WaitUntilComplete` و یا از `WaitUntilContainsText`.

نظرات خوانندگان

نویسنده: آرش خوشبخت
تاریخ: ۲۱:۳۰ ۱۳۹۲/۰۱/۲۴

اگر سوالی یا مشکلی در کارکردن با این فریم ورک داشتین می‌توانید اینجا بپرسین

مقدمه:

مدیریت آزمون مایکروسافت یا Microsoft Test Manager یک ابزار تست نویسی است که به تسترها این اجازه را می‌دهد تا بتوانند برای UI برنامه‌های خود یا sprintهای پروژه خود تست بنویسند. این ابزار برای نوشتن آزمون‌های پیشرفته و مجتمع سازی مدیریت طرح‌های تست یا test plans همراه با مورد‌های تست یا test case در طول توسعه برنامه است. یکی از مزایایی که این ابزار دارد این است که در طول انجام تست می‌توانید اشکالات تست را ثبت کنید و هم چنین می‌توانید شرحی در مورد انجام تست یا اشکالی که در آن تست وجود دارد، ثبت کنید. همچنین می‌توانید گزارشی از تست‌هایی که انجام داده اید و پاس شدن یا پاس نشدن تست‌ها و تاریخ انجام آن‌ها را نیز مشاهده کنید. قبل از کار با نرم افزار MTM باید یک سری مطالب مهم را در مورد انجام تست و مفهوم Agile بدانیم.

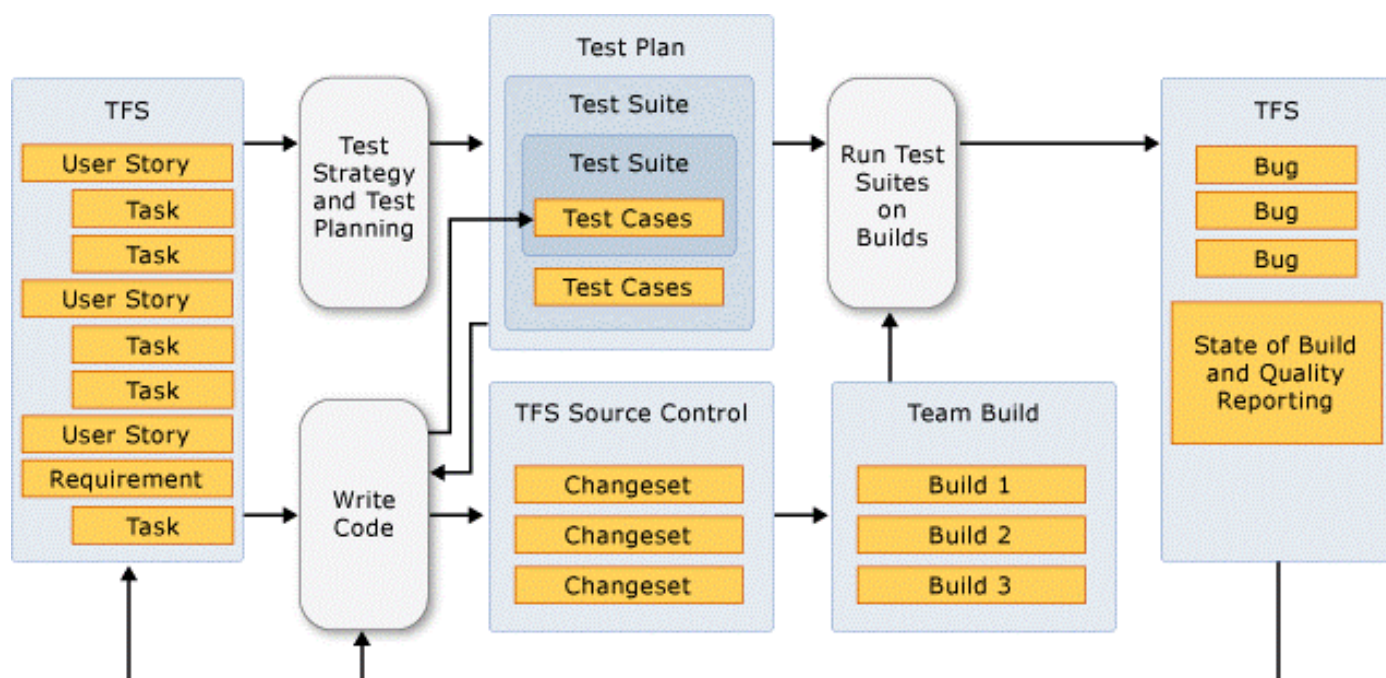
استراتژی تست:

زمانی که شما تست Agile را معرفی می‌کنید تیم برنامه نویسی شما می‌تواند بر روی تست‌های شما هم در سطح sprint و هم در سطح پروژه تمرکز کنند. تست در سطح sprint شامل تست‌هایی می‌شود که همه user story ها در بر بگیرد یعنی در واقع همان تست‌های واحد شما می‌شود. در سطح پروژه هم شامل تست‌هایی می‌شود که چندین sprint را در بر می‌گیرد که در واقع می‌توان تست‌های integrated گفت. بهتر است زمانی که تیم برنامه نویسی کدنویسی می‌کنند شما طرح تست‌های خود را بسازید و برای انجام تست کاملاً آماده باشید. این تست‌ها شامل تست واحد، تست performance، تست امنیتی و تست usability و غیره می‌باشد.

برای آماده کردن تست Agile در ابتدا شما باید یک تاریخچه یا history از برنامه یا سیستم خود داشته باشید. شما می‌توانید با استفاده از Microsoft Test Manager طرح تست خود را برای هر یک از sprint های پروژتان بسازید و مورد‌های تست را مشخص کنید.

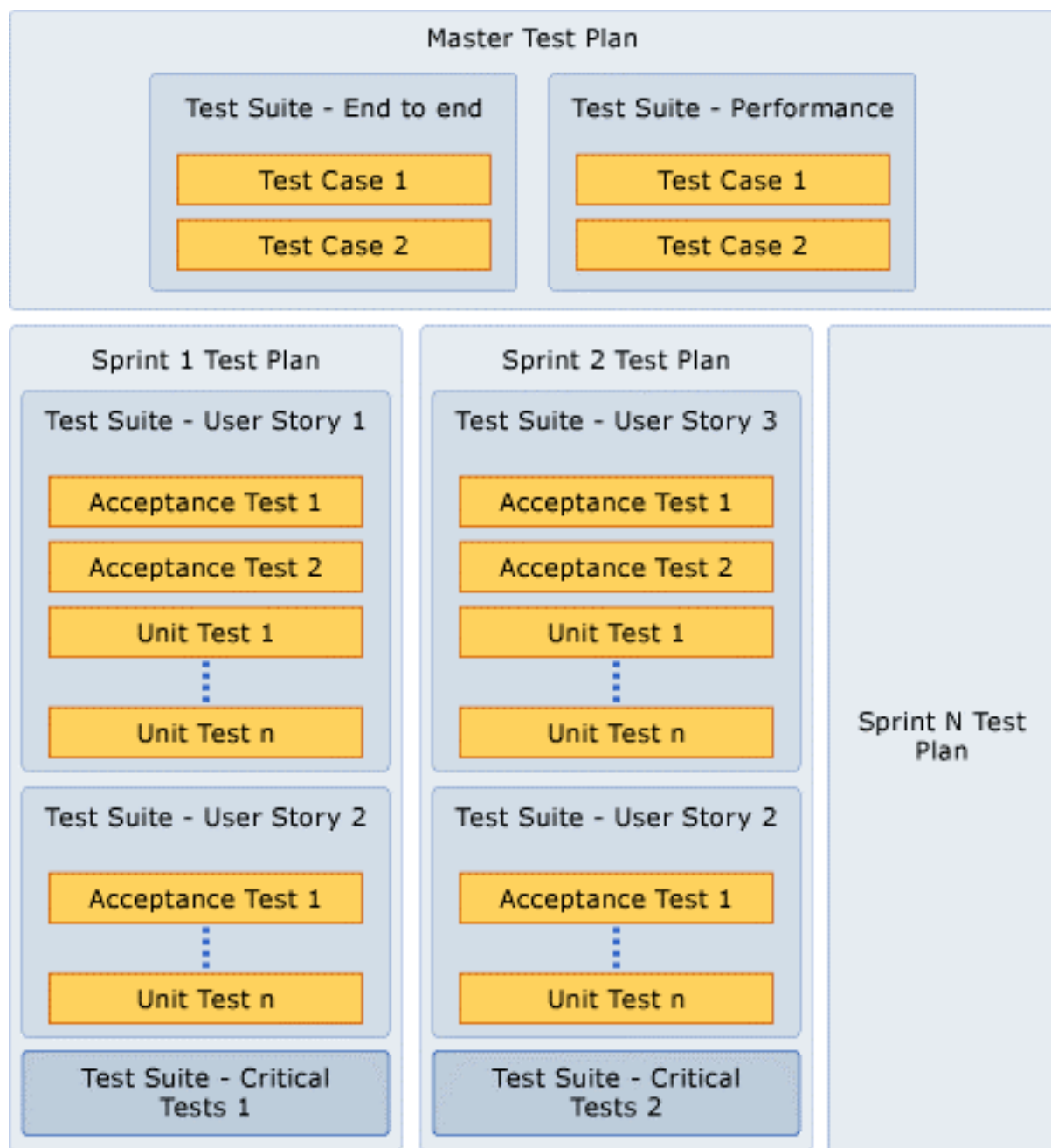
سپس باید کدهایی که برنامه نویسان می‌نویسند قابلیت تست را داشته باشند و شما به عنوان یک تستر باید آشنایی کاملی از ساختار و الگوهای برنامه تان داشته باشید.

تست یک فرآیند تکراری می‌باشد که همزمان با اجرای پروژه تان صورت می‌گیرد در زیر می‌توانید فرآیند کار تست و انجام کدنویسی را مشاهده نمایید:



: Test Planning

Test Planning فرآیندی است که به تیم شما کمک می‌کند تا درک درستی از پروژه داشته باشند و همچنین تیم را برای انجام هر گونه تستی آماده کند. تست Agile در سطح Sprint انجام می‌شود که در هر Sprint تیم شما تست‌هایی را ایجاد می‌کنند تا user story‌هایی که در هر Sprint وجود دارد، مورد بررسی قرار گیرند. در شکل زیر قالبی از test plan‌های شما در یک پروژه را نمایش می‌دهد:



البته این قالب‌ها بر اساس سلیقه شخصی است اما در کل می‌توانیم قالب تست را به صورت بالا در نظر بگیریم.

همیشه باید این را در نظر داشته باشیم که در طول هر sprint حتماً باید تست‌ها را اجرا کرده و در صورت وجود خطا، آن خطا را رفع کنیم تا در مراحل بالاتر با مشکلی مواجه نشویم. در قسمت بعد با Microsoft Test Manager و روش‌های نوشتن sprint و تست‌ها آشنا خواهیم شد.

نظرات خوانندگان

نویسنده: سیروان عقیفی
تاریخ: ۱۳۹۲/۰۲/۰۵ ۰:۵

با تشکر از شما، مطلب خوبی بود.

نویسنده: آرش خوشبخت
تاریخ: ۱۳۹۲/۰۲/۰۱ ۸:۶

خواهش می‌کنم امیدوارم مطالبم خوب نوشته شده باشه چون در نوشتن کمی ضعیف هستم

نویسنده: مهدی
تاریخ: ۱۳۹۲/۰۲/۰۳ ۱۹:۱۱

با سلام خدمت دوست عزیز و تشکر از این مقاله مفید.
لطفاً اگر می‌شود در مورد اصطلاحاتی که بیان می‌کنید در اول مقاله به تعریفی از آنها بیان کنید.
با تشکر.

نویسنده: آرش خوشبخت
تاریخ: ۱۳۹۲/۰۲/۰۳ ۲۱:۱۲

خواهش می‌کنم ولی منظور شما کدام اصطلاحات است؟ چون در قسمت دوم خیلی‌های این اصطلاحات رو گفتم اگر اصطلاحی رو متوجه نشدین بگین تا واستون توضیح بدم

تا اینجا متوجه شدیم که test plan چیست و چگونه ساخته می‌شود و برای نوشتن تست‌ها چه مراحل را باید طی کنیم. در این مطلب قصد بر این است که آموزش نوشتن تست‌ها با استفاده از MTM را آموزش دهیم. در این آموزش فرض بر این است که شما آشنایی کمی با محیط این ابزار، نیازمندی‌ها و Story ها، اشکالات یا Bug ها و Task ها دارید.

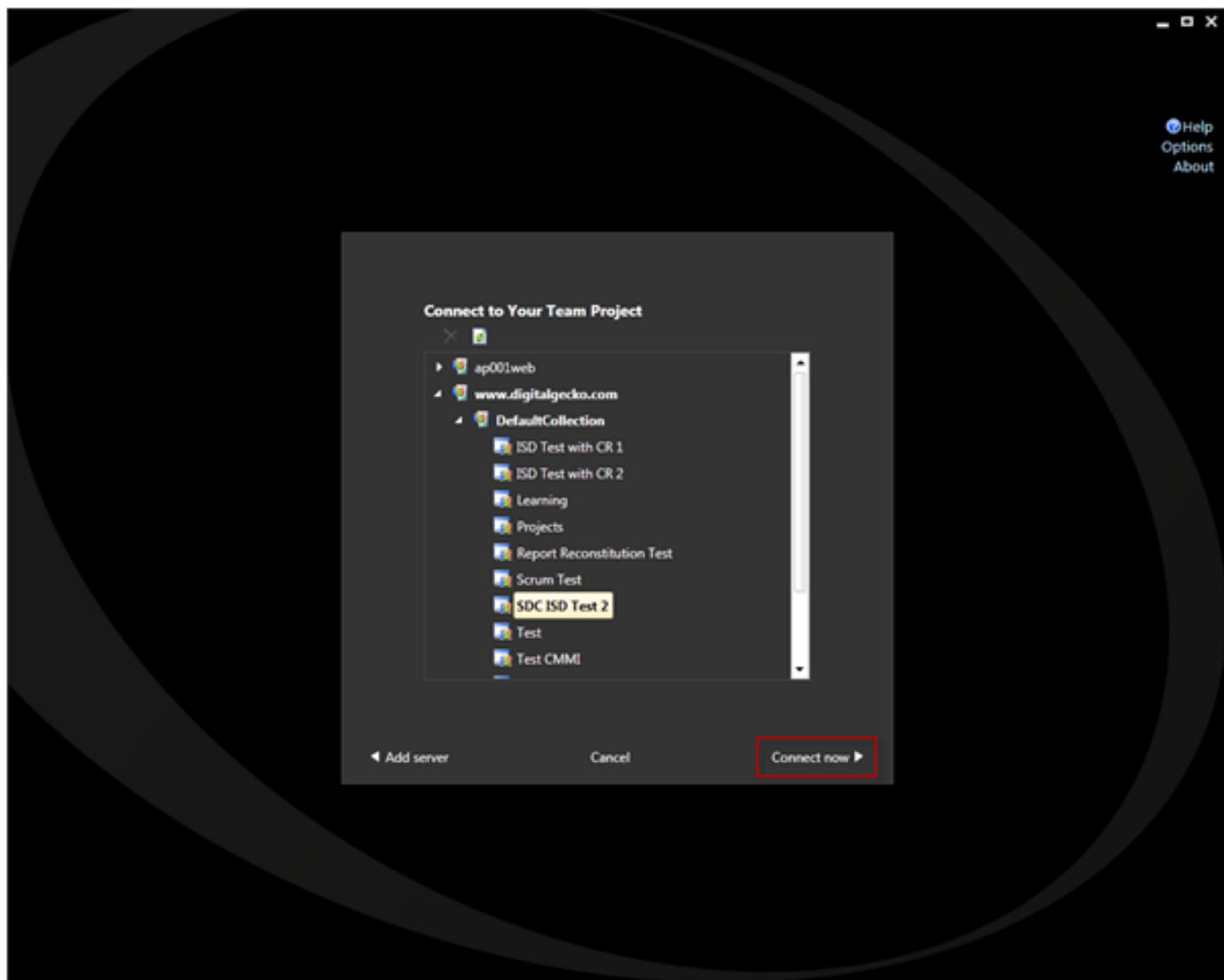
در MTM سه لایه وجود دارد:

1- **Test Plan** : شما در آغاز کار با MTM ابتدا باید Test Plan خود را ایجاد کنید.

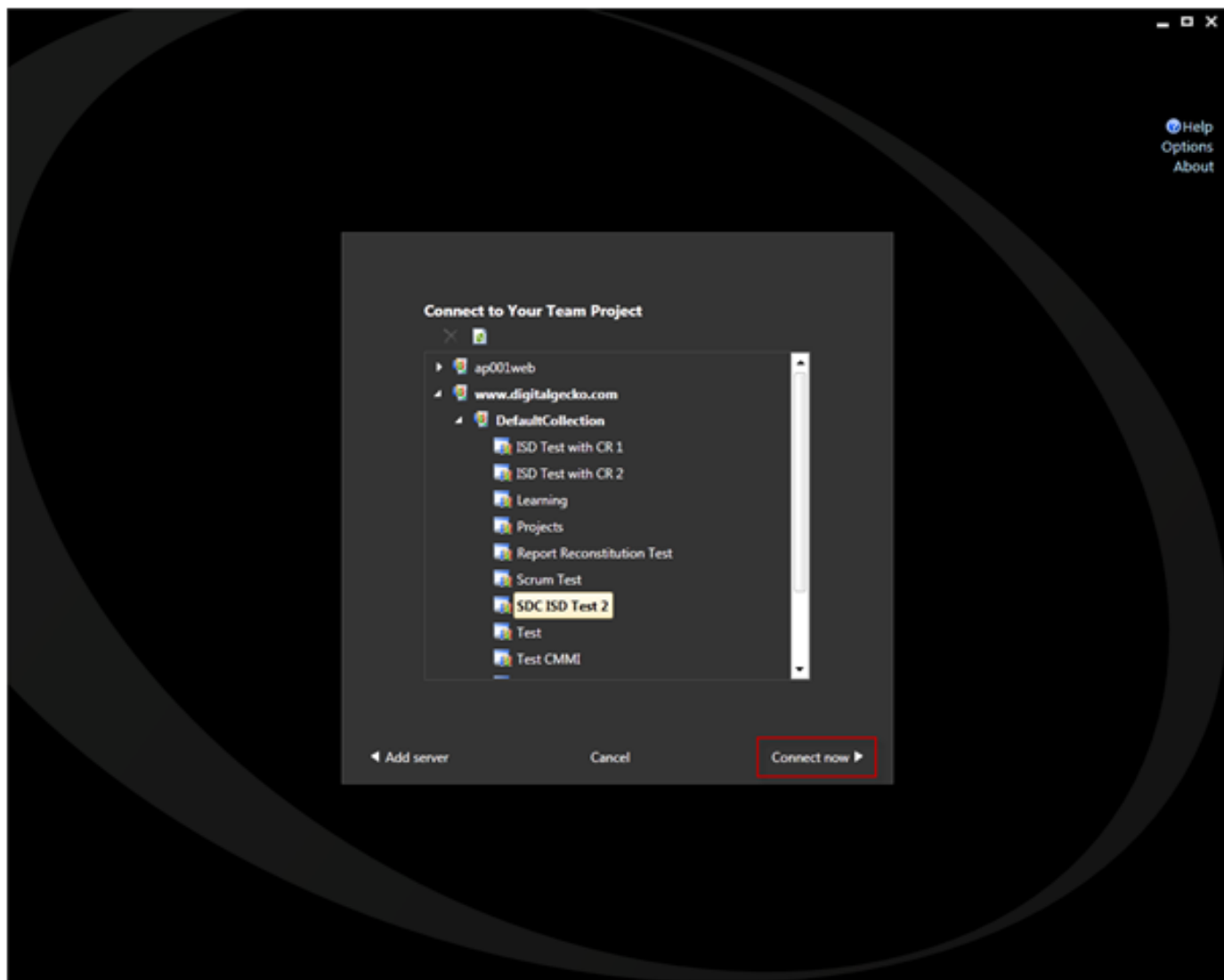
2- **Test Suite** : در هر Test Plan شما می‌توانید چندین Test Suite ایجاد کنید.

3- **Test Case** : هر Test Suite از چندین Test Case ترکیب شده است.

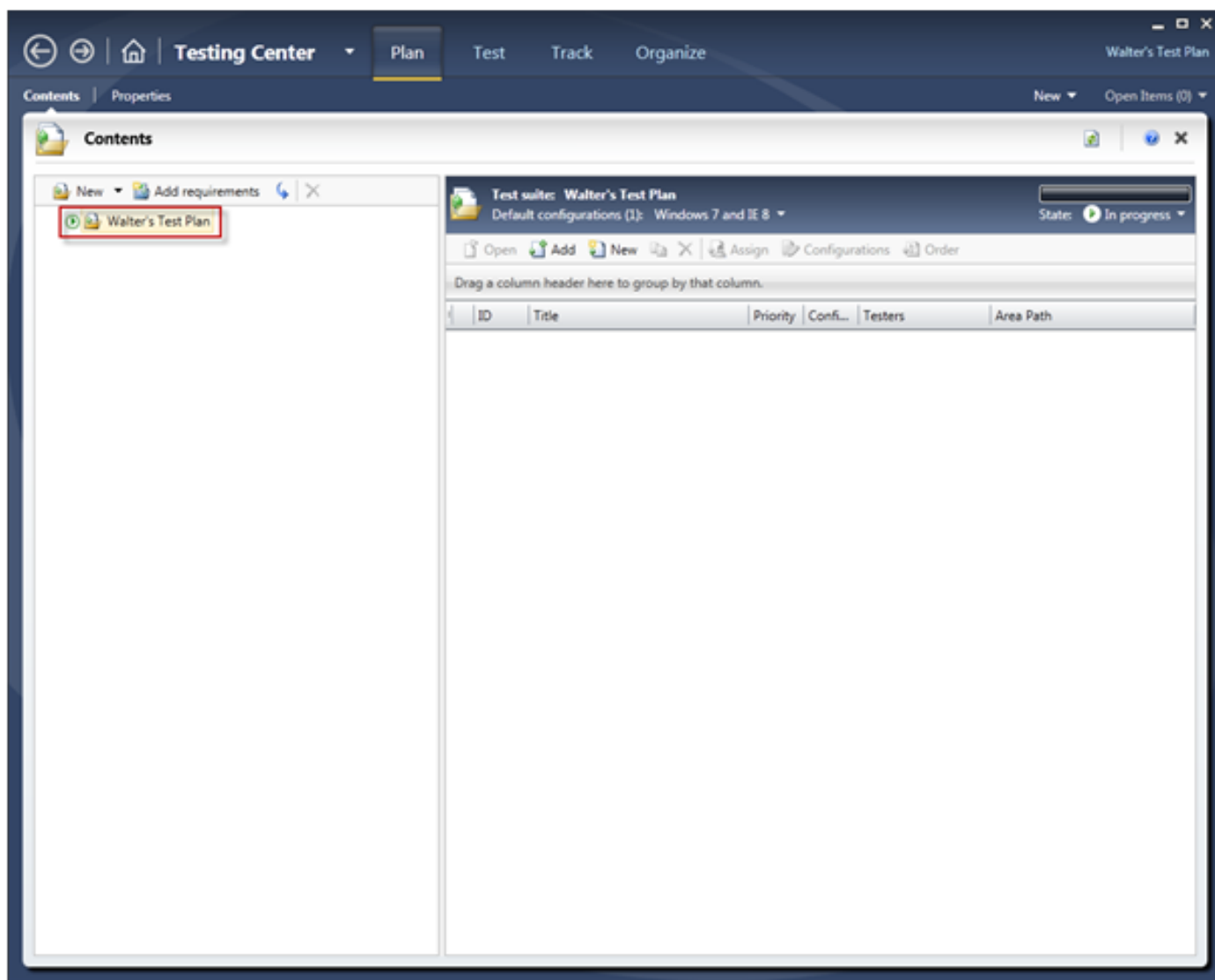
برای اولین بار که شما MTM را اجرا می‌کنید باید team project ی را که قرار است برای آن تست بنویسید را انتخاب کنید. می‌توانید در زیر نمایی از MTM و اتصال به team project را مشاهده کنید:



بعد از اینکه پروژه خود را انتخاب کردید، می‌توانید لیستی از طرح‌های تست تان که برای این پروژه ایجاد کرده اید را مشاهده کنید که می‌توانید از این لیست یک طرحی را انتخاب نمایید و یا یک طرح جدید را ایجاد کنید همانطور که در شکل زیر مشاهده می‌کنید.



وقتی plan یا طرحی را انتخاب می‌کنید به صفحه testing center وارد می‌شوید که به صورت پیش فرض در کاربرگ plan و بخش contents قرار دارید.



همانطور که در تصویر بالا مشاهده می‌کنید و در سمت چپ پنجره، plan شما در ریشه قرار دارد و test suite هایی را که ایجاد می‌کنید به عنوان فرزندان plan تان قرار می‌گیرند. در سمت راست test case های شما قرار می‌گیرند که با توجه به test suite ی که شما در سمت چپ انتخاب کرده اید test case های مربوط به آن در سمت راست قابل مشاهده است. برای ایجاد test suite به plan تان، باید روی plan راست کلیک کرده و گزینه new suite را انتخاب کنید و برای آن عنوانی را وارد می‌کنید. وقتی روی plan راست کلیک می‌کنید پند گزینه وجود دارد که می‌توانید با توجه به کارتان این گزینه‌ها را انتخاب کنید:

1- وقتی new suite را انتخاب می‌کنید یک suite خالی برای شما ایجاد می‌کند.

2- وقتی گزینه new query-based suite را انتخاب می‌کنید این اجازه را به شما می‌دهد که از test case های موجود در پروژه خود یک یا چندین مورد تست را انتخاب نمایید که پنجره ای مانند زیر باز می‌شود که می‌توانید با اعمال فیلتر، test case های موجود در پروژه را پیدا و یک یا چندین مورد را به suite خود اضافه نمایید.

Create a Query-Based Suite

Name:

| And/Or | Field | Operator | Value |
|------------------------------|----------------|----------|--------------------|
| ► | Team Project | = | @Project |
| And | Work Item Type | In Group | Test Case Category |
| * Click here to add a clause | | | |

Run
Column options
Open
Create copy
Create test case from bug

| ID | Title | Assigned To | Area Path |
|---|-------|-------------|-----------|
| <p>Use the query builder to add clauses to limit the work items returned by the query. Click Run to see the work items returned by the query.</p> | | | |

Create test suite
Don't create suite

3- گزینه add requirement to plan این اجازه را به شما می‌دهد تا بتوانید از plan‌های موجود در TFS تان استفاده نمایید. بعد از انتخاب این گزینه پنجره ای مشابه تصویر بالا باز می‌شود که می‌توانید با اعمال فیلتر موردی تست را پیدا کرده و به آن بیافزایید.

Add existing requirements to this test plan

Query Type:
Work Items and Direct Links

| And/Or | Field | Operator | Value |
|------------------------------|----------------|----------|---------------------|
| ► | Team Project | = | @Project |
| And | Work Item Type | = | [Any] |
| And | State | = | [Any] |
| And | Area Path | Under | PorsemanDevelopment |
| And | Title | = | Login Successful |
| * Click here to add a clause | | | |

Filters for linked work items

| And/Or | Field | Operator | Value |
|------------------------------|----------------|----------|----------------------|
| ► | Work Item Type | In Group | Requirement Category |
| And | Title | = | Login Successful |
| * Click here to add a clause | | | |

Linking Filters

Run
Column options
Open
Create copy
Create test case from bug

| ID | Link Type | Work Item... | Title | Assigned To | Area Path |
|---|-----------|--------------|-------|-------------|-----------|
| <p>Use the query builder to add clauses to limit the work items returned by the query. Click Run to see the work items returned by the query.</p> | | | | | |

Add requirements to plan
Don't add

4- با انتخاب گزینه copy suite from another plan می‌توانید از suite‌های مربوط به plan‌های دیگر کپی برداری کنید.

نظرات خوانندگان

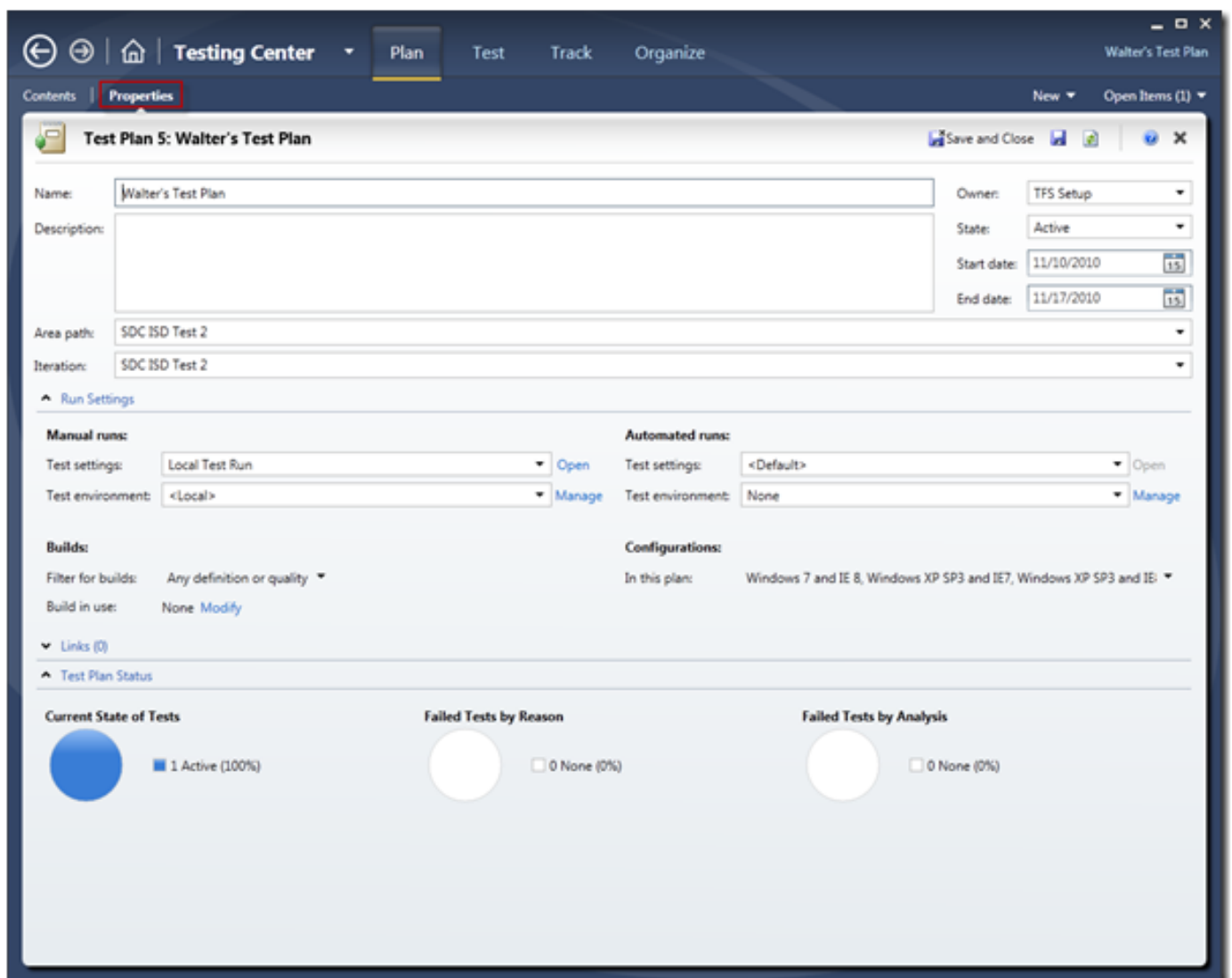
نویسنده: علیرضا پونه
تاریخ: ۱۳۹۲/۰۲/۰۲ ۸:۴۹

ممنونم. فقط اینکه تو هر پست مطلب رو کاملتر و قسمت بیشتری رو بگین تا در تعداد پست کمتری بشه همه چیز رو گفت و هم اینکه خواننده تا پست بعدی، خیلی از مطلب دور نشه. بازم بابت مطلب بسیار مهمی که دارین آموزش میدین خیلی خیلی ممنون.

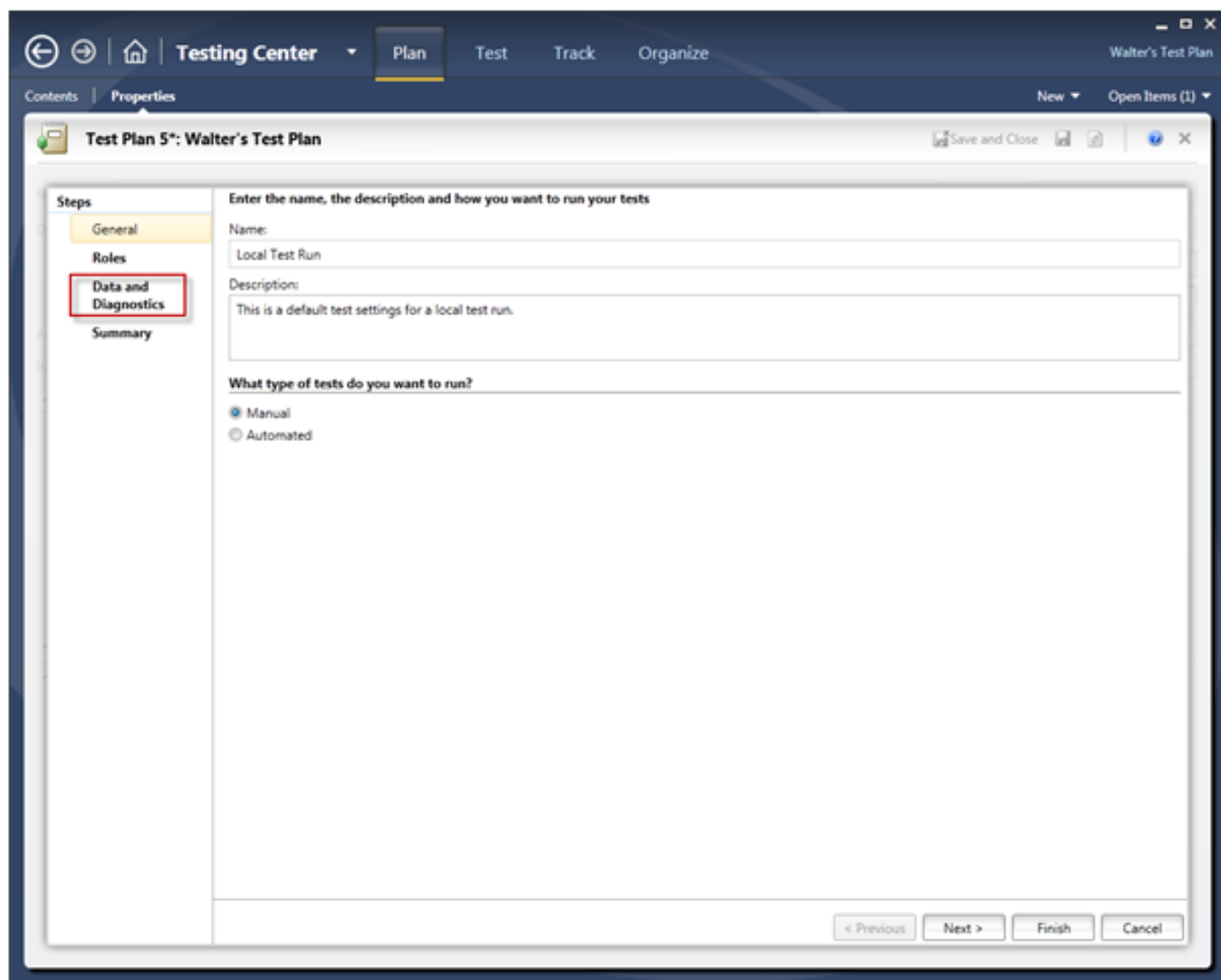
نویسنده: آرش خوشبخت
تاریخ: ۱۳۹۲/۰۲/۰۲ ۱۲:۱

دلیل اینکه مطلب زیاد نمیزارم چون می‌گم شاید کاربران خسته شن یا حوصله‌ی خوندن مطلب زیاد رو نداشته باشن و بعد اینکه مبحث جدیدی که بخواد شروع بشه مجبورم قسمت قبل رو قطع کنم قسمت بعدی در مورد یک سری تنظیمات در MTM است و ربطی به این بخش نداره

در کنار کاربرد contents کاربرگی با نام Properties وجود دارد که می‌توانید یک سری تنظیمات را برای plan خود انجام دهید. این تنظیمات از قبیل تغییر عنوان plan، تعیین مسیر پروژه، تاریخ شروع و پایان، کاربری که مالک این plan است، وضعیت جاری تست‌های plan و تعیین مرورگر و ویندوز نیز می‌باشد که می‌توانید در تصویر زیر آن را مشاهده کنید.

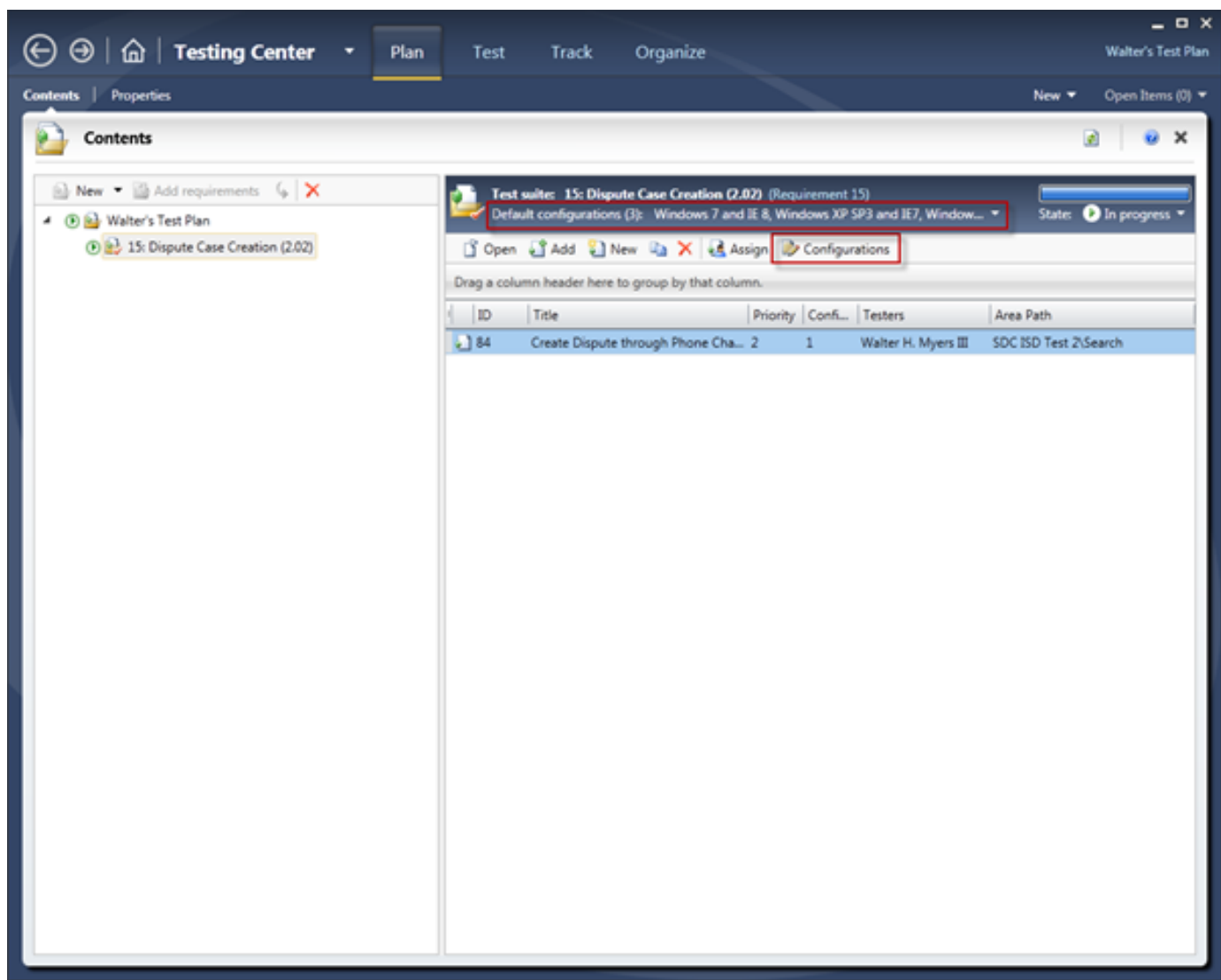


اگر در لیست کشویی مربوط به test settings مقدار <default> قرار داشت می‌توانید با انتخاب آیتم new از لیست settings جدیدی را ایجاد نمایید و یا می‌توانید لیست test settings هایی را که قبلاً ایجاد کرده اید انتخاب نمایید و برای ویرایش آن با کلیک بر روی لینک open که کنار لیست قرار دارد، می‌توانید تنظیمات را ویرایش نمایید.

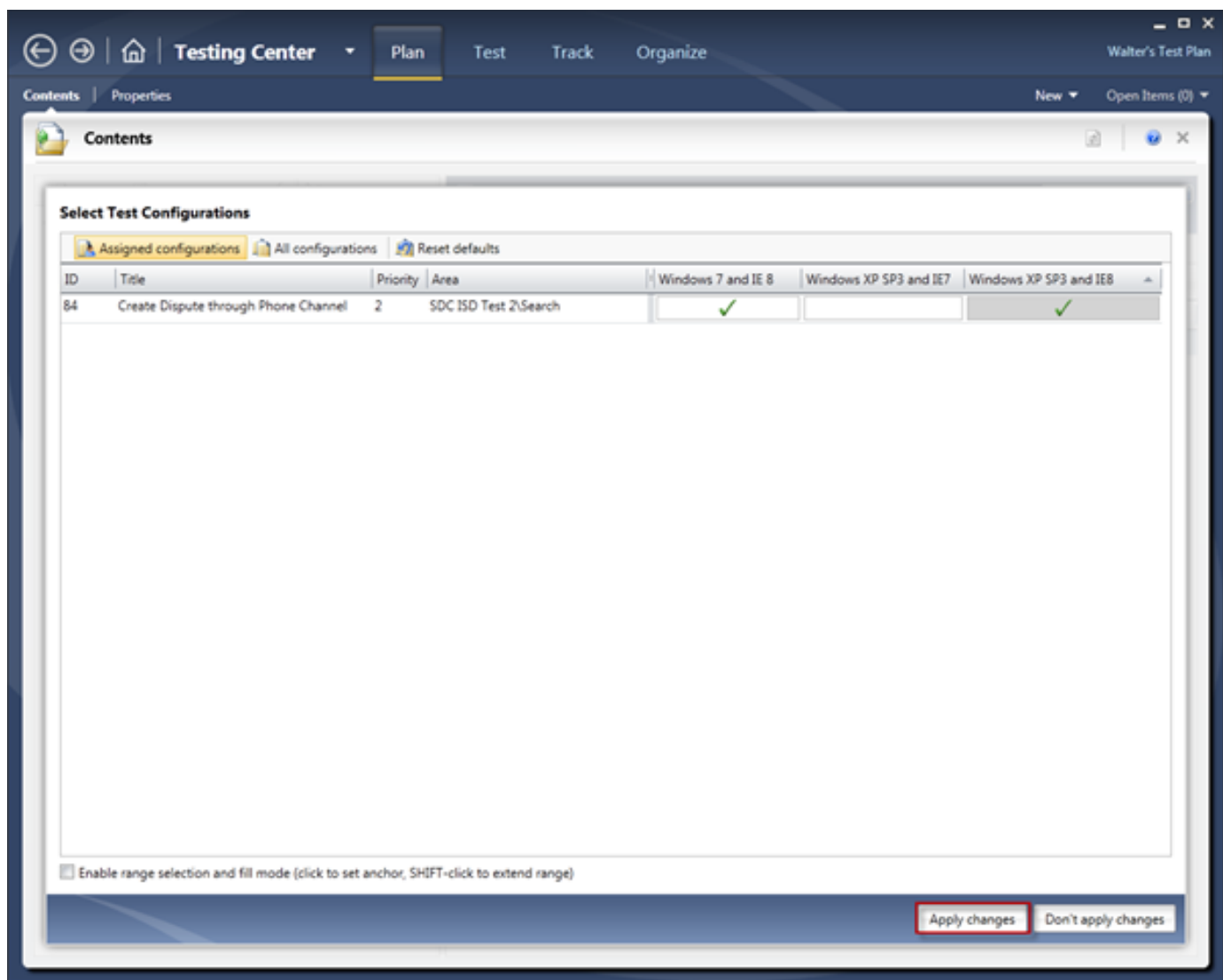


همانطور که در تصویر بالا مشاهده می‌کنید، در سمت چپ، بخش هایی برای انجام تنظیمات مربوط به تست وجود دارد. در قسمت general تنظیماتی از قبیل عنوان test settings، شرح و نوع اجرای دستی یا اتومات بودن تستتان وجود دارد. در بخش roles می‌توانید نقش هایی را برای این تست انتخاب نمایید و در قسمت data and diagnostics می‌توانید یک سری اطلاعاتی را که می‌خواهید در زمان تست دریافت کنید، انتخاب کنید. برای اطلاعات بیشتر در مورد این بخش می‌توانید در [سایت مایکروسافت](#) مطالعه کنید.

حالا بر می‌گردیم به بخش contents و موارد تست خود را می‌سازیم. همانطور که در تصویر پایین مشاهده می‌کنید در بخش contents و در سمت راست پنجره یک گزینه ای به نام configuration وجود دارد.

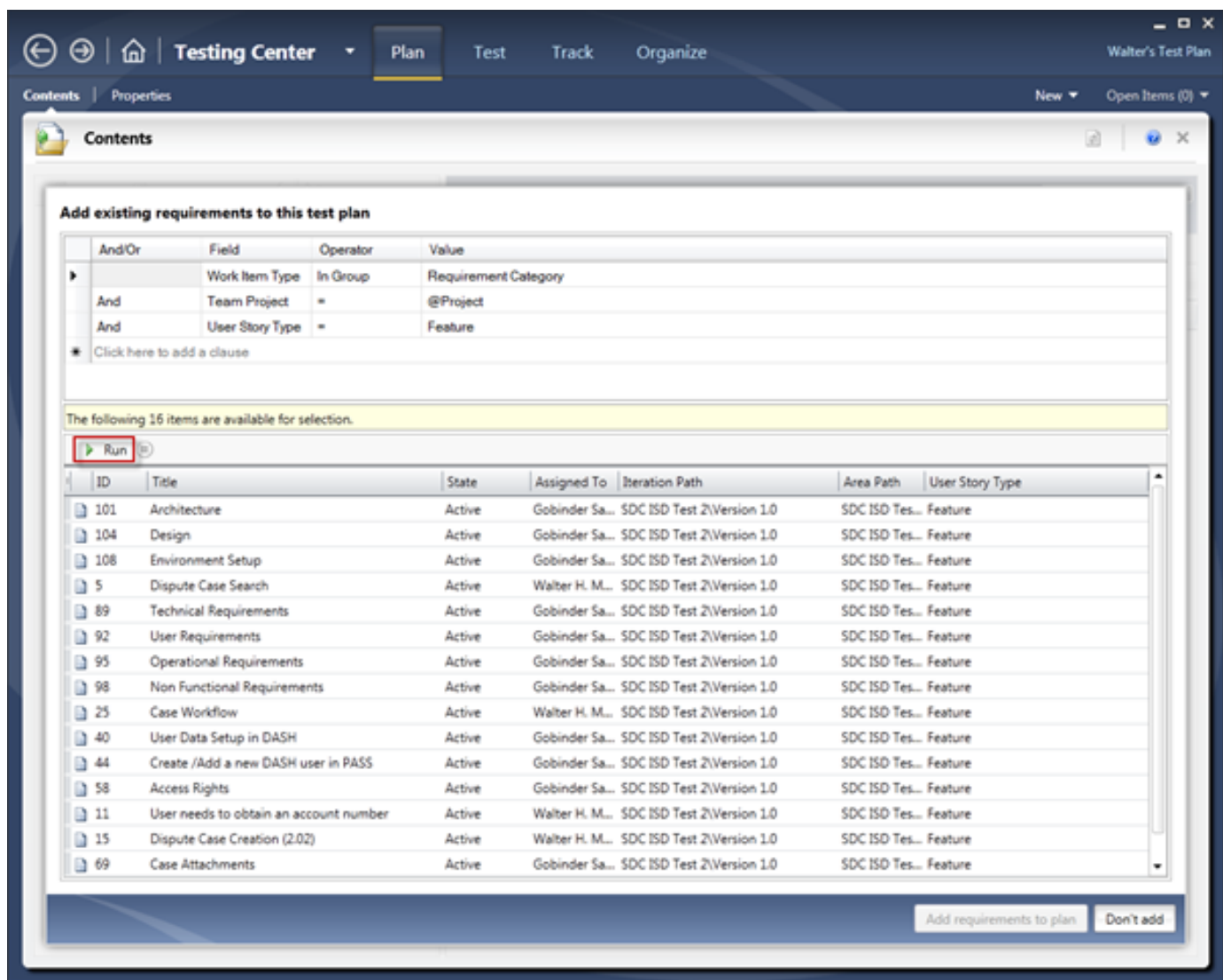


در configuration شما می‌توانید یک سری تنظیمات مربوط به test شما انجام دهید مثلاً نوع مرورگری که می‌خواهید تست خود را اجرا کنید و یا اولویت تست را مشخص نمایید یا حتی نوع سیستم عامل را مشخص کنید. هم چنین می‌توانید چندین configuration تعریف کنید و از هر کدام برای یک test suite استفاده کنید. به صورت پیش فرض test suite از تنظیمات config والد خودش یعنی test plan استفاده می‌کند.



دوباره برمی گردیم به بخش contents و می خواهیم یک test suite با استفاده از add requirements بسازیم. همانطور که در بخش های قبل توضیح دادم می توانیم به چند روش test suite بسازیم که یکی از آن ها همین add requirements بود که می توانستید از test suite هایی که قبلا ساخته اید به این پروژه تستتان اضافه کنید.

با انتخاب گزینه add requirements پنجره ای باز می شود که می توانید همه test suite ها را مشاهده کنید و حتی می توانید براساس عنوان و یا وضعیت تست و ... فیلتر کنید.



بعد از اینکه در قسمت بالا کوئری خود را تنظیم کردید با انتخاب گزینه run می‌توانید کوئری خود را اجرا کرده و لیست test suiteها را براساس آن کوئری فیلتر کنید. می‌توانید یک یا چند سطر را انتخاب کرده و با زدن دکمه add requirements to plan آن‌ها را به plan خود اضافه نمایید. حالا ما یک test suite با استفاده از test suite هایی که قبلاً ساخته ایم ایجاد کردیم. حالا باید مورد تست‌های مان را به این test suite اضافه کنیم. در سمت راست با کلیک بر روی گزینه add پنجره ای مشابه پنجره بالا باز می‌شود که شما می‌توانید test caseها را فیلتر کنید و یک یا چند مورد را انتخاب کرده و با زدن دکمه add test cases آن‌ها را به test suite تان اضافه کنید. برای اضافه کردن مورد تست جدید هم می‌توانید با کلیک بر روی new که در کنار گزینه Add قرار دارد مورد تست جدیدی را بسازید.

در تصویر زیر می‌توانید بخش‌های مختلف تست را که در بخش‌های قبل هم توضیح دادم ببینید.

The screenshot displays the Microsoft Test Manager interface. The top navigation bar includes 'Testing Center', 'Plan', 'Test', 'Track', and 'Organize'. The 'Plan' tab is active. The left pane shows a tree view of test items: 'Walter's Test Plan' (labeled 'TestPlan') and '15: Dispute Case Creation (2.02)' (labeled 'TestSuite'). The right pane shows the details of the selected test suite, including a table of test cases. One test case is listed with ID 84, titled 'Create Dispute through Phone Cha...', with a priority of 2, configuration of 1, assigned to 'Walter H. Myers III', and located in the 'SDC ISD Test 2\Search' area path (labeled 'TestCase').

| ID | Title | Priority | Conf... | Testers | Area Path |
|----|-------------------------------------|----------|---------|---------------------|-----------------------|
| 84 | Create Dispute through Phone Cha... | 2 | 1 | Walter H. Myers III | SDC ISD Test 2\Search |