



delegate

CRMUG - MS CRM Solution Packager
2016-02-08 @ Københavns Kommune





- Matching of expectations
- Short introduction: Speaker and Delegate A/S (CRM)
- MS CRM Solution Packager
 - What is it, why use it and mostly how to we use it “correctly”
- Live demo
- Summary (a small surprise)
- Q&A
- NDC London 2016 – Topics for later



- What are your expectations for talk?



- Last CRMUG meeting we heard that a lot of members had some difficulties in using Microsoft Solution Packager
- As we at Delegate A/S have been using the tool since 2011. This allows us to store the blue print of our solutions in the code repositories, which then gives us the possibility to recreate a solution based on a code commit tag. By using this approach, it's easier to maintain a solution as we are just creating software and hereby we can use known approaches to Application Lifecycle Management (ALM)
- Therefore we would like to showcase a demo where we go from A to Z explaining every step. Code samples will be made available @ GitHub after the talk.

Note: Feel free to interrupt and ask questions underway



- Ramón Soto Mathiesen
- MSc. Computer Science from DIKU (Minors in Mathematics)
- Managing Specialist | > CTO of CRM Department @ [Delegate A/S](#)
 - ER-modeling, WSDL, OData (REST API)
- F# / C# / JavaScript / C++: [Delegate A/S @ GitHub](#)
- Blog: <http://blog.stermon.com/>
- Talks:
 - 2016-02-08: CRMUG – MS CRM Solution Packager
 - 2016-03-12: SPBG – A combination of MS SharePoint & CRM to ensure atomic transactions
 - 2016-05-09: CRMUG – MS CRM in the Cloud from technical point of view (pending)



- IT-Consultancy known for their SharePoint solutions and established in 2006
- The CRM department was established in February 1, 2013 when I arrived. The head of our Department is Kaspar Bøgh Christensen, former Dynamics CRM director at Avanade, and we are a total of 14 consultants working at our two locations: Aarhus and Copenhagen
- The company's motto: *"We must be the best, not necessarily the biggest"*

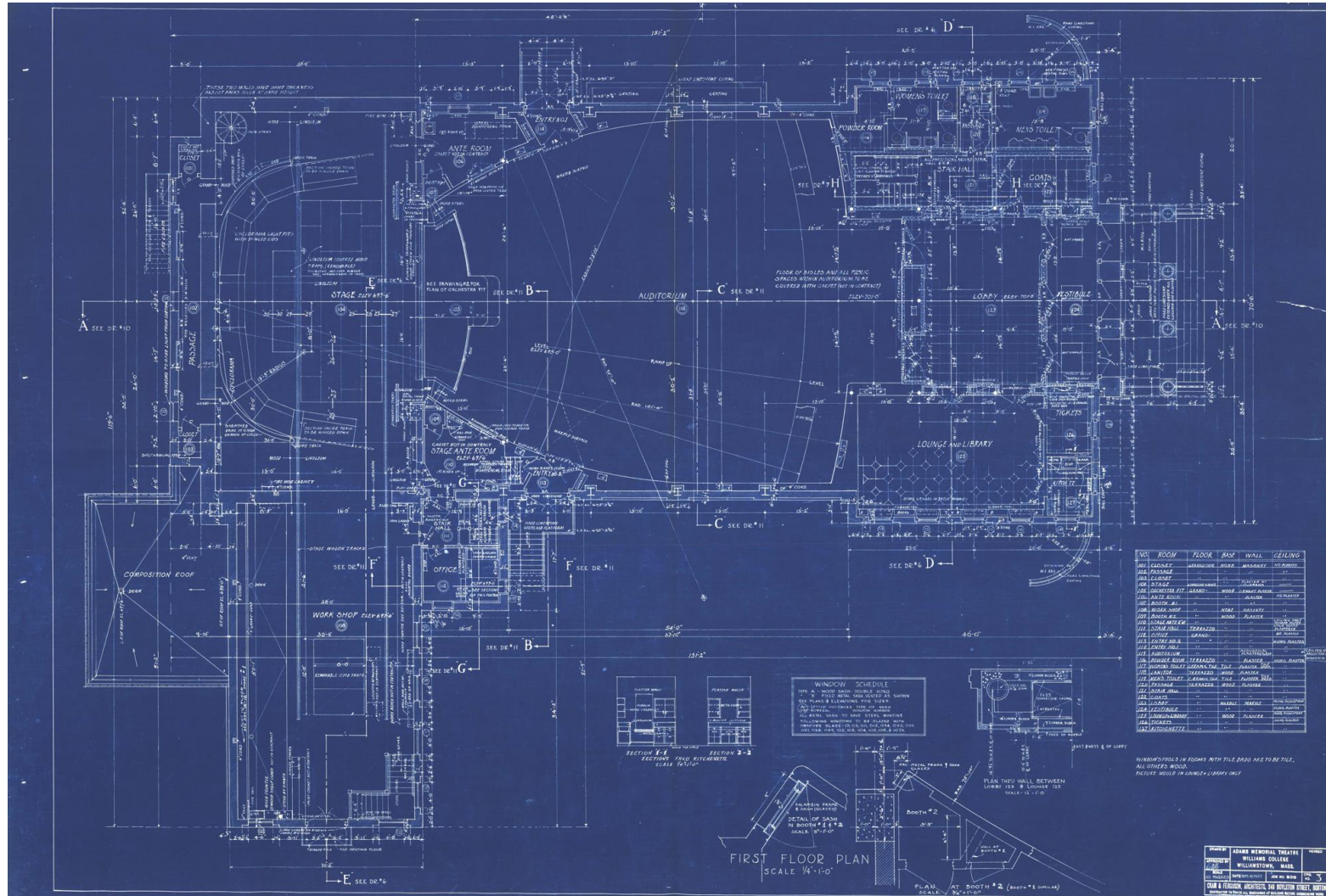




- **SolutionPackager** is a tool that can reversibly decompose a Microsoft Dynamics CRM compressed solution file into multiple XML files and other files so that these files can be easily managed by a source control system
- Solution component file reference (folder structure and file naming scheme):
 - <https://msdn.microsoft.com/en-us/library/jj602990.aspx>



- **Application Lifecycle Management (ALM):** By splitting the solutions customizations file into smaller parts which can be stored in a source control system, it gives us the possibility to re-create the solution at any given code commit which is almost indispensable when making/maintaining software solution
- **Multiple developers:** It will also allow several people working with their own local solution and when a component is ready, then merge into the staging solution which will be built and deployed to a TEST solution and afterwards to PROD



- Up-to-date Blueprint of “something” will always allow you to re-create it the same way



- *“When the **SolutionPackager** tool extracts the component files it **will not overwrite** existing component files of the same name if the file contents are identical. In addition, the **tool honors the read-only attribute** on component files producing a warning in the console window that particular files were not written. **This enables the user to check out, from source control, the minimal set of files that are changing.** The **/clobber** parameter can be used to override and cause read-only files to be written or deleted. The **/allowWrite** parameter can be used to assess what impact an extract operation has without actually causing any files to be written or deleted. Use of the **/allowWrite** parameter with verbose logging is effective”*
- Use source control with solution files:
 - <https://msdn.microsoft.com/en-us/library/jj602997.aspx>
- Use the SolutionPackager tool to compress and extract a solution file:
 - <https://msdn.microsoft.com/en-us/library/jj602987.aspx>



- *“This enables the user to check out, from source control, the minimal set of files that are changing. The /clobber parameter can be used to override and cause read-only files to be written or deleted.”*
- TFS as source control “is broken”. The reason is that if you have made a change to a file (Edit) and then you delete it, then TFS will complain that it can’t commit the edited file because it’s deleted (it’s stores both actions). Also there is the issue that TFS handles files that are read-only in the file system as committed and those who are not as checked-out. Unless you have to, please don’t use TFS as your source control
- It’s possible to make SolutionPackager work with TFS, but we discovered it made us change our natural code commit behavior (always commit before Extract/Pack even though code was not ready). If you want to set up TFS correctly, please follow the guide in the next slides

Remark: Not even Microsoft use TFS as source control, they use Git, as the rest of the world



New Team Project on tfsserver.delegate.dk\CRM

Specify the Team Project Settings

The New Team Project Wizard uses the team project name you type here when creating various components. After the team project is created, the name is used by team members to locate the team project.

Make sure that the name you pick for the team project is not already in use by Team Foundation Server or any other software used in the deployment (for example, SharePoint Products or SQL Server Reporting Services).

What is the name of the team project?

What is the description of the team project?

< Previous Next > Finish Cancel

New Team Project on tfsserver.delegate.dk\CRM

Select a Process Template

The process template defines key aspects of how the team project is managed. The process template may include work item types, work products, reports, queries, and process guidance for your team project.

Which process template should be used to create the team project?

Scrum

Agile

CMMI

Scrum

MSF for Agile Software Development 2013.3

Microsoft Visual Studio Scrum 2013.3

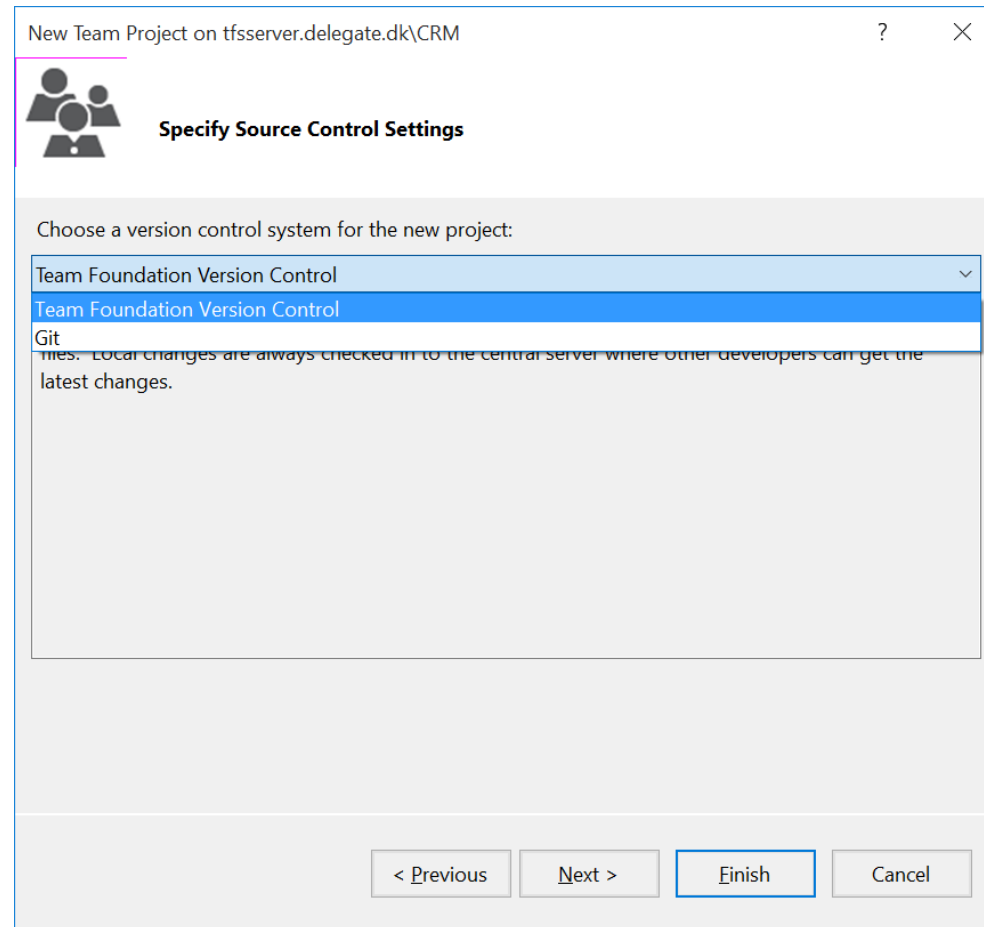
MSF for CMMI Process Improvement 2013.3

[View more information about process templates](#)

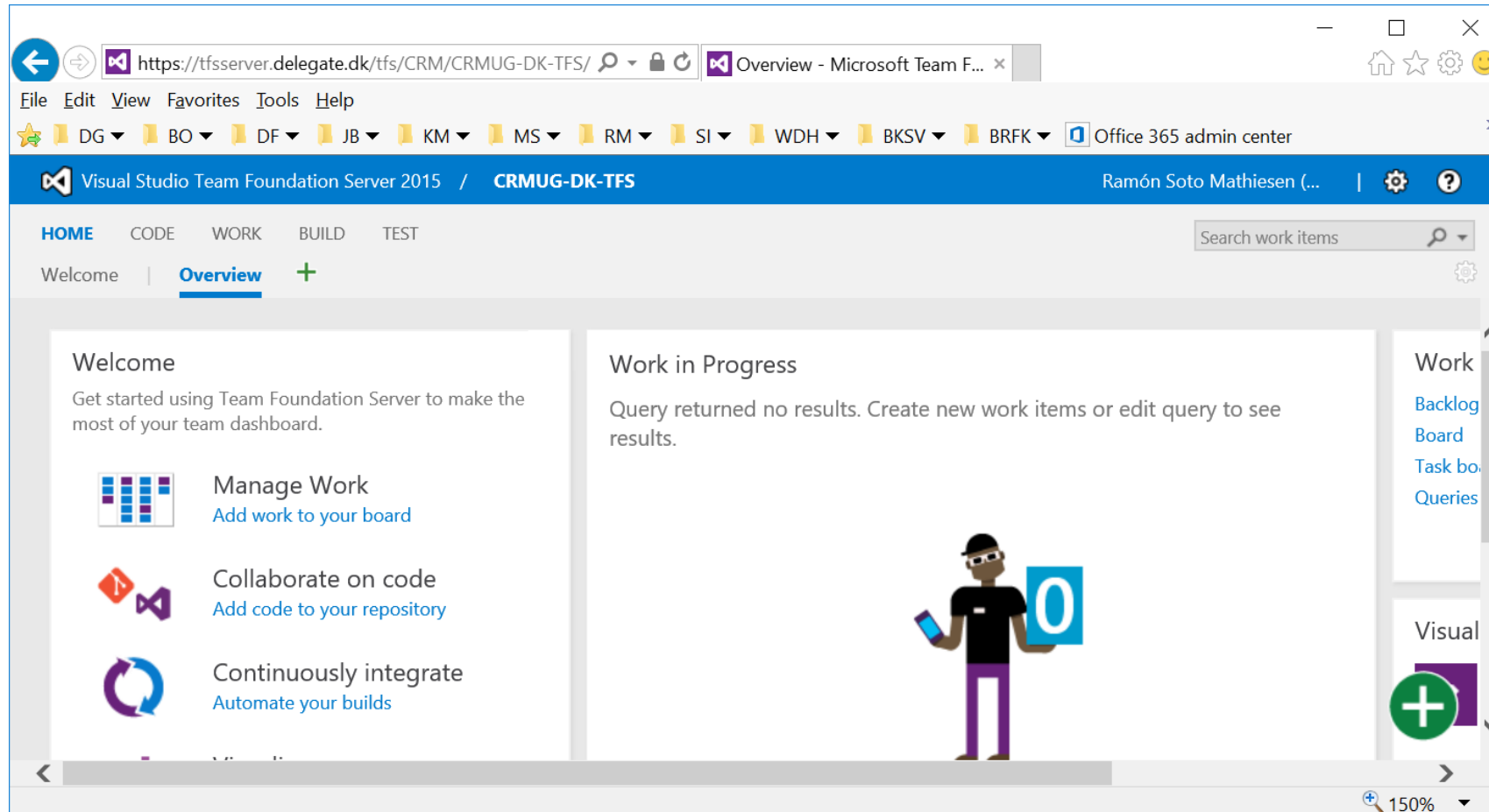
Find Microsoft Visual Studio Certified Process Templates.

< Previous Next > Finish Cancel

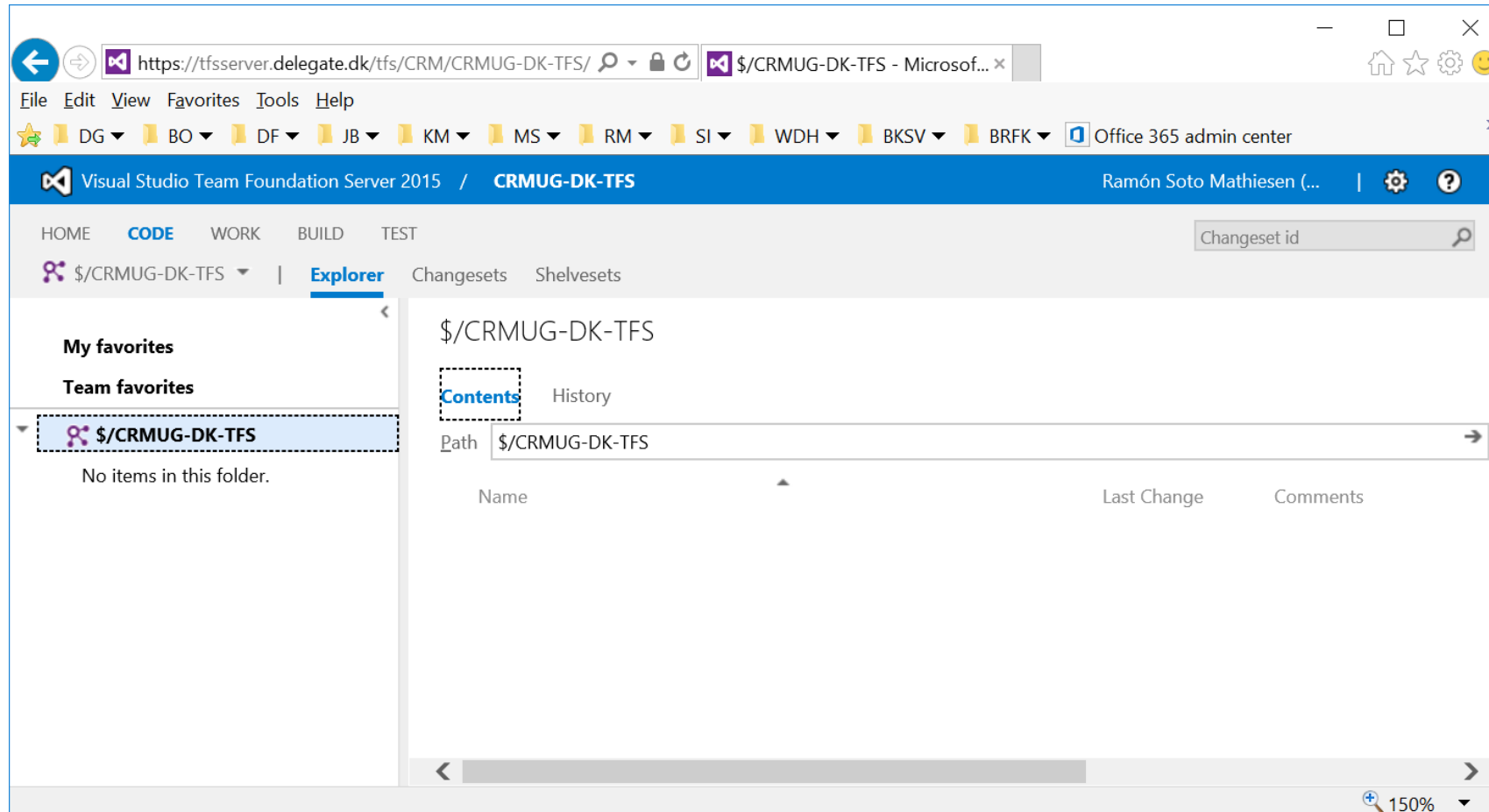
- Create a TFS project from Visual Studio (based on SCRUM template)



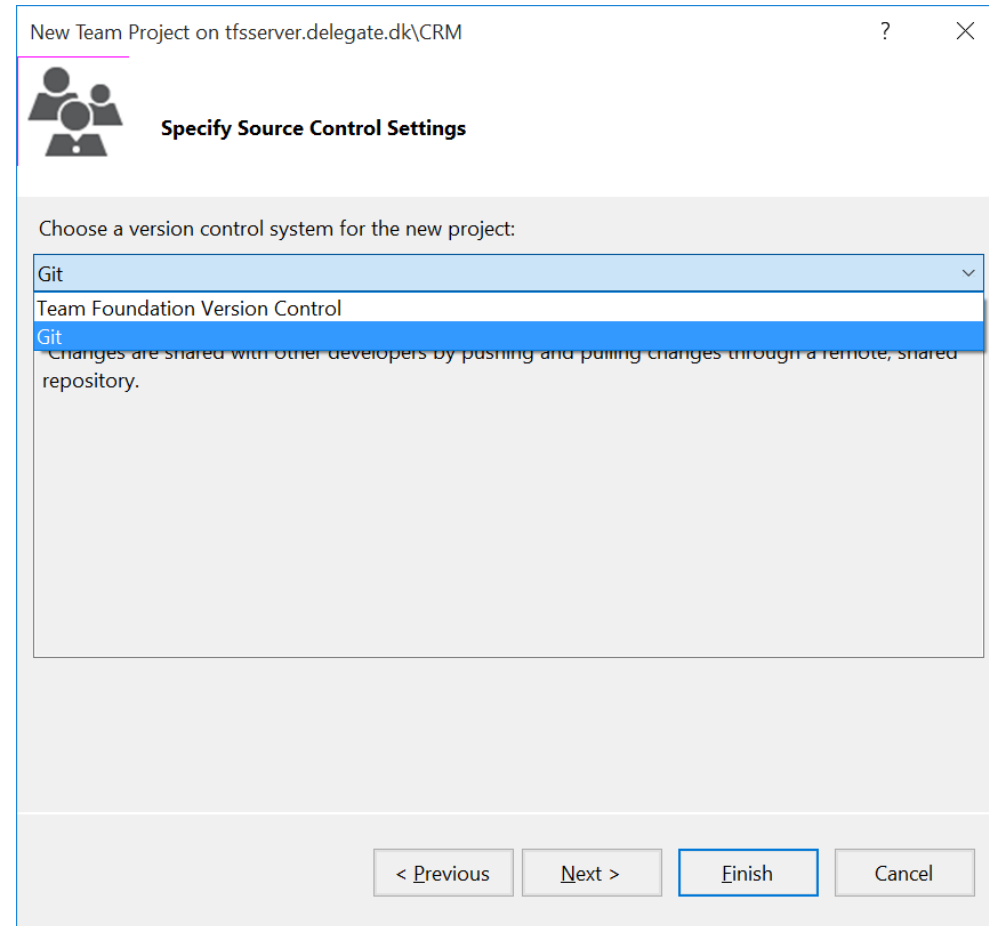
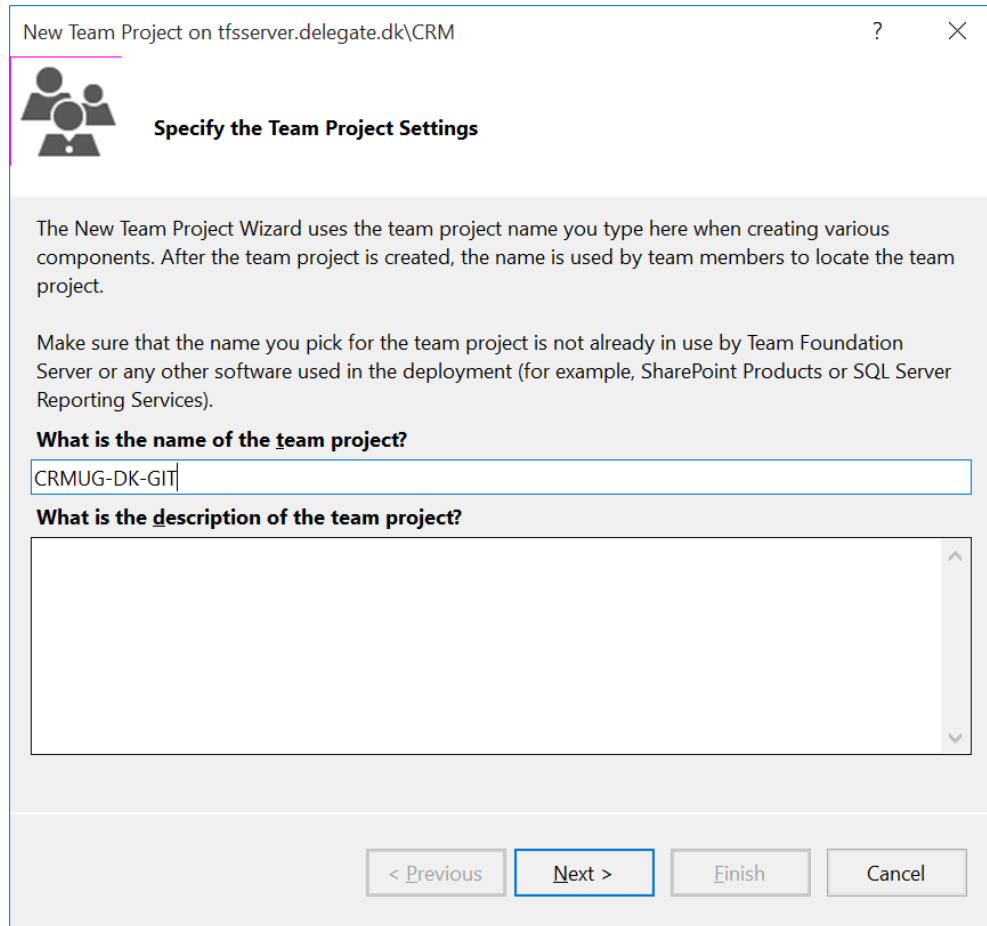
- Create a TFS project from Visual Studio (choosing TFS as source control, **WRONG !!!**)



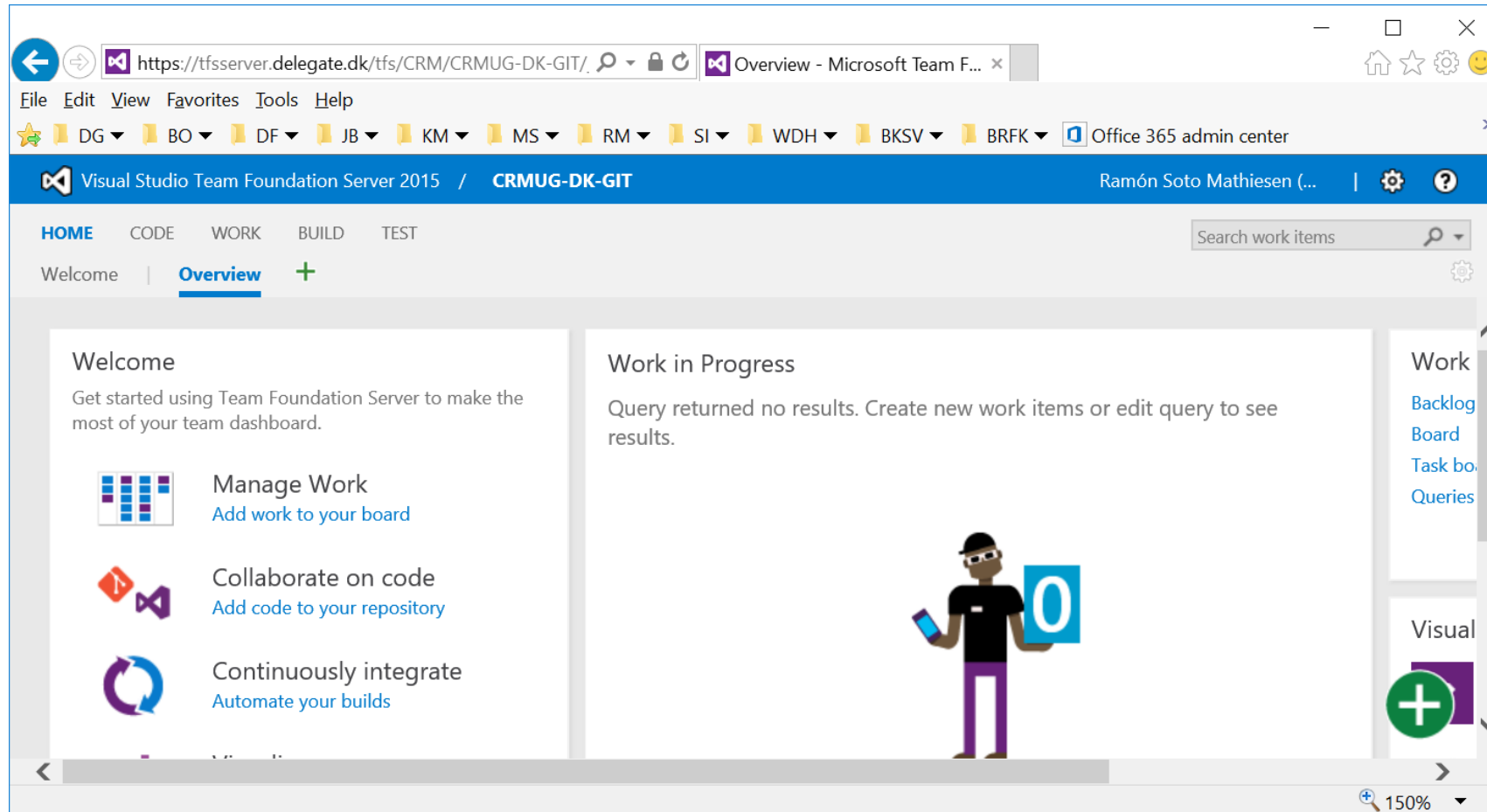
- Create a TFS project from Visual Studio (Management pages)



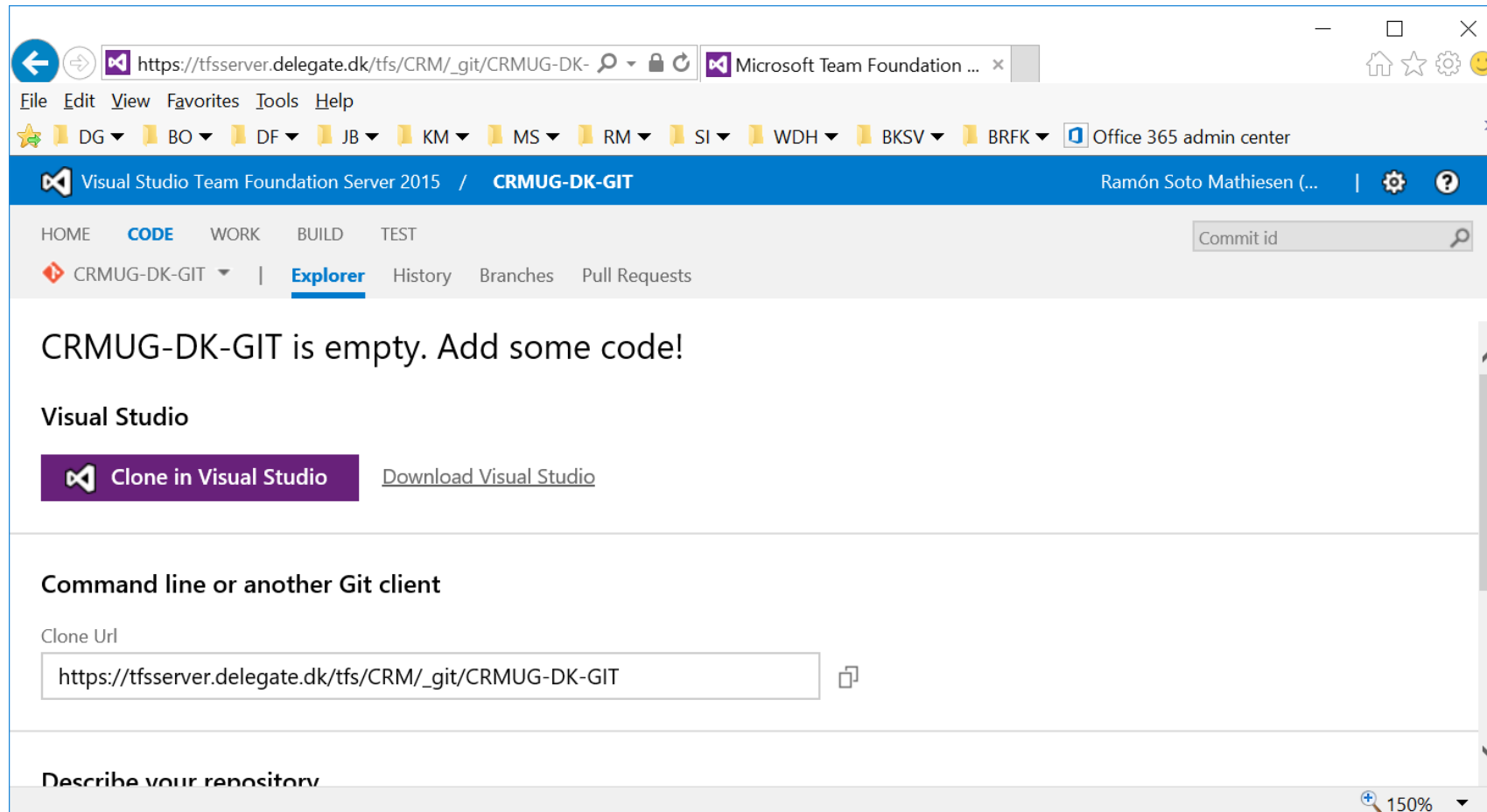
- Create a TFS project from Visual Studio (TFS as source control)



- Create a TFS project from Visual Studio (choosing GIT as source control, Correct 😊)



- Create a TFS project from Visual Studio (Still same management pages)

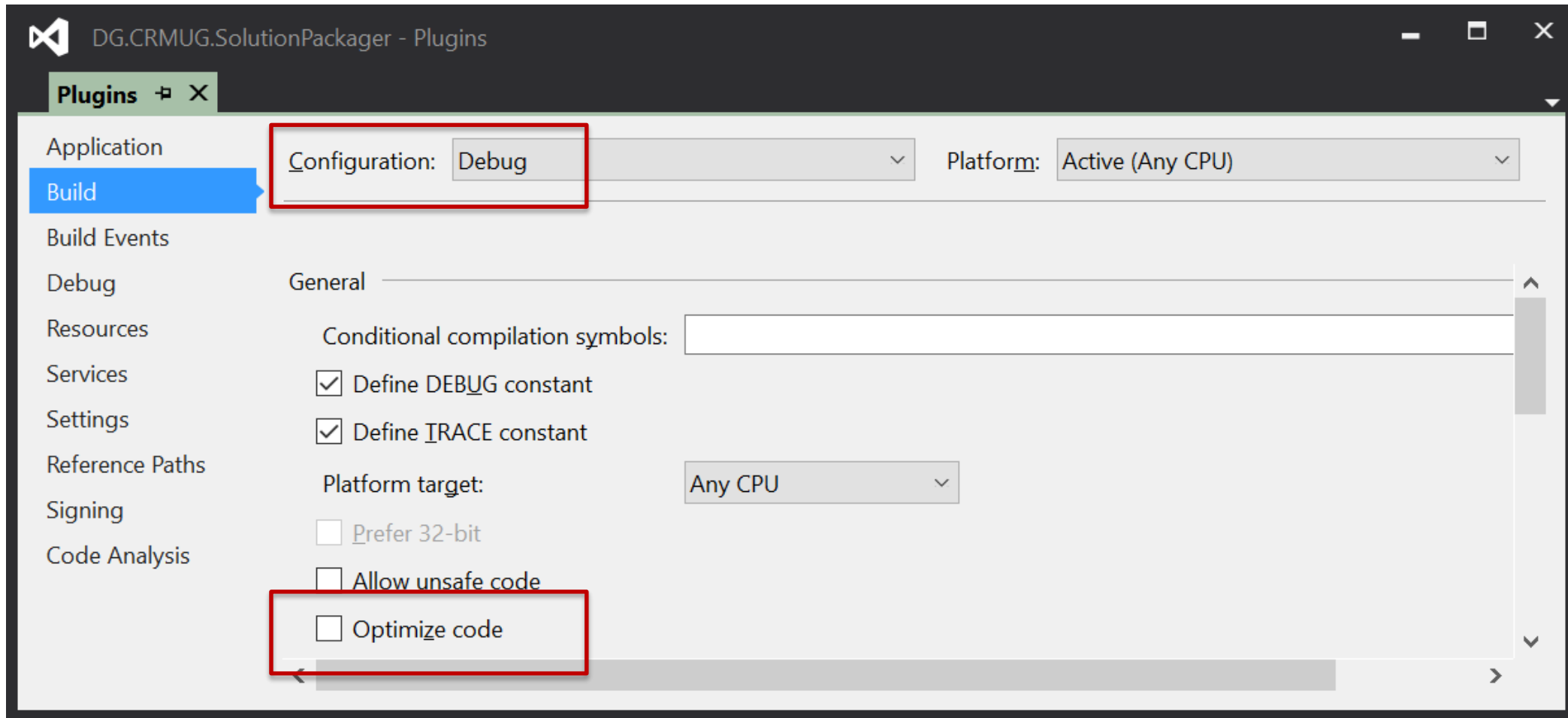


- Create a TFS project from Visual Studio (but only with different source control)



- What we normally see:
 - Add assemblies to DEV environment with Plug-in registration tool
 - Assemblies are built with the Debug Profile (see next slide)
 - Manually extraction of Default.zip (not even a subset solution)
 - Eventually save the hole .ZIP file in the source control system (SVN, TFS, ...)
 - Impossible to work with on a multiple DEV setup (diff of files in Visual Studio is not possible)
 - Deploy extracted .ZIP file from DEV → TEST and go through Test Cases
 - Finally deploy extracted .ZIP from DEV → PROD

MS CRM Solution Packager - How to use it? ("correctly")



Note: Finally deploy extracted .ZIP from DEV → PROD (code is not optimized and debug trace may affect the business logic with Debugger Launch points and I/O writing)



- What we would like to see:
 - Add assemblies to DEV environment with Plug-in registration tool (or better tooling 😊)
 - Assemblies are built with the Debug Profile
 - Retrieve the unmanaged and managed subset solution from DEV (ex: FooBar.zip)
 - Extract the solution to the source control system (*)
 - Merge diff locally before committing to the source control system as you would normally do when you are working on a software solution
 - Deploy the packed .ZIP from the source control system GIT → TEST and go through Test Cases
 - Ensure that Assemblies are built with the Release Profile
 - Finally deploy the packed .ZIP from the source control GIT → PROD

(*) – Managed and Unmanaged extracted solution are saved in the same folder structure. It's important to use the SolutionPackager mapping file in order to tell the tool that code (Plug-in and Workflows as well as Web Resources) are handled by a Visual Studio solution



- Two very important remarks:
 - SolutionPackager doesn't like long paths. Your source control folder should be as close to your drives root:
 - Wrong: C:\Users\foo\Documents\Visual Studio 2015\DG.CRMUG\DG.CRMUG.FooBar.sln
 - Correct: C:\git\DG.CRMUG\DG.CRMUG.FooBar.sln
 - When pointing to assemblies in the SolutionPackager mapping file, names must not have dots:
 - Wrong:
 - <FileToFile
 - » map="ILMerged.Delegate.CRMUG.SolutionPackager.Plugins.dll"
 - » to="..\..\Plugins\bin**\ILMerged.Delegate.CRMUG.SolutionPackager.Plugins.dll"
 - />
 - Correct:
 - <FileToFile
 - » map="ILMergedDelegateCRMUGSolutionPackagerPlugins.dll"
 - » to="..\..\Plugins\bin**\ILMerged.Delegate.CRMUG.SolutionPackager.Plugins.dll"
 - />



- MS CRM Online
 - Source: <https://crmugdksp00.crm4.dynamics.com/>
 - Target: <https://crmugdksp01.crm4.dynamics.com/>
- CRM Administrator
 - admin@crmugdksp00.onmicrosoft.com
 - admin@crmugdksp01.onmicrosoft.com



The screenshot shows the DAXIF# web application interface. The browser address bar indicates the URL is `http://localhost:30524/Daxif/diff`. The application header displays the 'delegate' logo and the title 'DAXIF# Delegate Automated Xrm Installation Framework v. 2.2.0.0 : Diff Report 2016-02-07T08:01:55.1878895+01:00'. The main content area is divided into two sections: 'Diff Tree' and 'Diff Viewer'. The 'Diff Tree' section shows a summary of changes:

Tree	Unchanged	Changed	Not in source	Not in target
+ Entities	39			
+ Other	17			
+ SdkMessageProcessingSteps	5			
- PluginAssemblies		1		
- ILMergedDelegateCRMUGSolutionPackagerPlugins-63FB4563-65CD-E511-80DB-6C3BE5BEEFC0		1		
• ILMergedDelegateCRMUGSolutionPackagerPlugins.dll				

The 'Diff Viewer' section shows a comparison between the source and target files for 'ILMergedDelegateCRMUGSolutionPackagerPlugins.dll':

Source: SolutionPackager.zip	Target: SolutionPackager_zip
5C2B765DAA3E924F2FB86FE6A29D4447263DF579	A6AE2F8DD905A814C8EDBEDF40F3351378593E48

The 'Diff Legend' on the right side of the interface defines the color coding for the diff report: Added (green AA), Changed (yellow AA), Removed (red AA), Ignored (red AA), Moved From (grey AA), and Moved To (grey AA).

- Demo Script:

1. Retrieve solution from DEV and Extract (Use Visual Studio to see Diff between CRM and GIT)
2. Deploy .ZIP file from DEV → TEST and see it fail
 - Show code where if Debug, code will fail
3. Pack from source control and deploy to TEST and see it succeed

Note: If time show diff between DEV/GIT and Managed/Unmanaged with Daxif.Solution.diff



®

open source
initiative

- As of today, well yesterday, **Delegate.Daxif** is now released under an **Open Source License**
 - Free to use for everybody 😊



- Code will be available @ <https://github.com/delegateas/DG.CRMUG.SolutionPackager>
- Slides will be available @ ppo@brf.dk
- Daxif:
 - NUGET package:
<https://www.nuget.org/packages/Delegate.Daxif/>
 - Github website with documentation and API Description:
<http://delegateas.github.io/Delegate.Daxif/>
 - Is constantly being developed:
http://delegateas.github.io/Delegate.Daxif/RELEASE_NOTES.html
- Other Open Source tools are available @ <http://delegateas.github.io/>



Q&A



- Microservices and Docker
 - Erlang: Only way to do it correctly 😊
- JavaScript: Iterators and generators (easier to write async code without callbacks)
- Entity Framework 7.0: No backwards compatible but support for NoSQL