

**Bernhard Merkle – Find(ing) Bugs Is Easy**

**Charles Bailey – The Rant Of Three**

**Simon Sebright – SharePoint for Thinking Developers**

**Phil Nash – Why I Do What I Do**

**Matt Turner - Fluency**

**Chris Oldwood – Not Only But Also**

**Frank Birbacher – Style C++ for Version Control**

**Frances Buontempo - TDD**

**Astrid Byro – A Cry For Help**

**Didier Verna – Why?**

# SharePoint for Thinking Developers in 5min

[simon.sebright@1stQuad.com](mailto:simon.sebright@1stQuad.com)

SharePoint Consultants & Developers

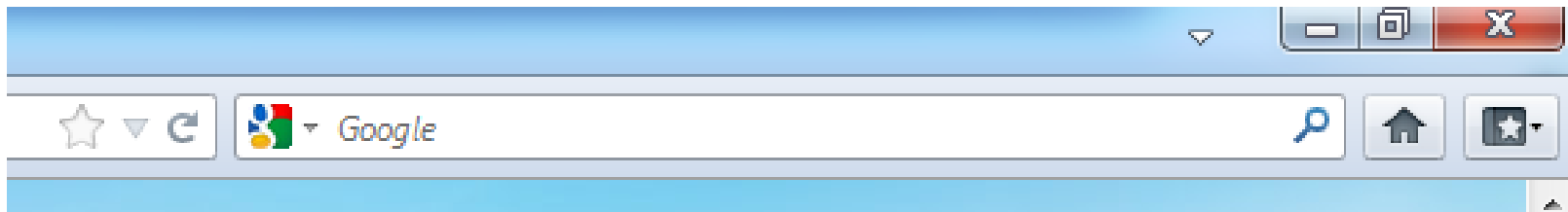
# So how does that work?

An ACCU Member =>  
professionalism in programming

A SharePoint Developer =>  
A hacker

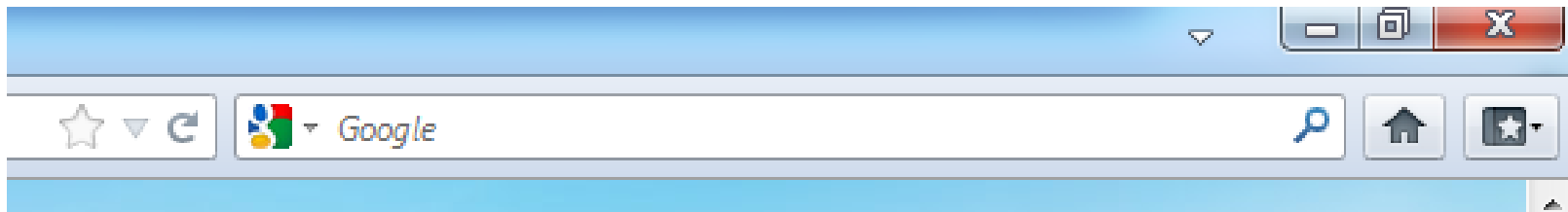
# Hacking Part 1

Development Tool Number 1:



# Hacking Part 1

Development Tool Number 1:



BDD: Blog-driven Development

# Hacking Part 1

Google for Blog entry

Copy code

Build, Deploy

Try it

lisreset

Restart some services

Reboot

Goto Top

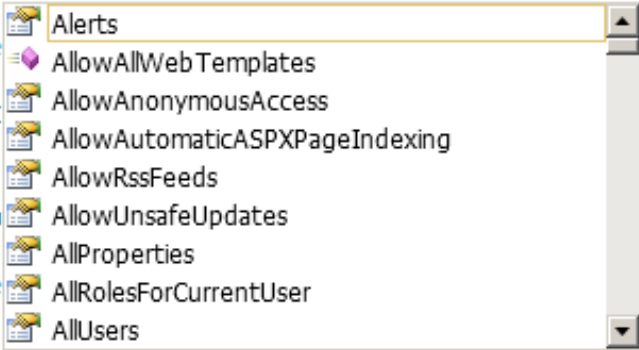
# Hacking Part 1

But you do have to know how to search  
effectively

# Hacking Part 2

## Development Tool Number 2:

```
SPList destList = web.Lists[destDocLib];  
  
if (copyOption == File  
{  
    DeleteDestinationI  
}  
  
string destName = sou  
  
if (copyOption == File  
{  
    try
```



Alerts	SPAlertCollection SPWeb.Alerts Gets the collection of alerts for the site or subsite.
--------	--

Intellisense or Dumbnesssense?



# What can we do? Use the language

```
public void DoItToSubWebs(SPWeb web)
{
    foreach (SPWeb subweb in web.Webs)
    {
        DoIt(subweb);
    }
}
```

# What can we do? Use the language

```
public void DoItToSubWebsBetter(SPWeb web)
{
    foreach (SPWeb subweb in web.Webs)
    {
        using (subweb)
        {
            DoIt(subweb);
        }
    }
}
```

# What can we do? Use the language

```
public void DoItToSubWebsBetter2(SPWeb web, Action<SPWeb> action)
{
    foreach (SPWeb subweb in web.Webs)
    {
        using (subweb)
        {
            action(subweb);
        }
    }
}
```

```
DoItWithSubWebsBetter2(web, DoIt);
```

# What can we do? Use the language

```
public IEnumerable<SPWeb> GetSmartWebs(SPWeb web)
{
    foreach (SPWeb subweb in web.Webs)
    {
        using (subweb)
        {
            yield return subweb;
        }
    }
}
```

```
foreach(SPWeb subweb in GetSmartWebs(web))
{
    DoIt(subweb);
}
```

# What can we do? Automated Tests:

```
[Test]
public void History_content_is_copied()
{
    using (SPWeb web = SPObjets.EnsureCleanWeb(TestBaseUrl, TestSiteName))
    {
        Setup();
        sourceList.RootFolder.Files.Add("a.txt", Encoding.UTF8.GetBytes("This is some text"));
        sourceList.RootFolder.Files.Add("a.txt", Encoding.UTF8.GetBytes("This is version 2"), true);
        sourceList.RootFolder.Files.Add("a.txt", Encoding.UTF8.GetBytes("This is version 3"), true);

        FileCopier.CopyFileWithVersions(TestSiteUrl + "/SourceLib/a.txt", "DestLib",
FileCopyOptions.Overwrite);

        SPFile destFile = destList.RootFolder.Files[0];
        Assert.That(destFile.Versions.Count, Is.EqualTo(2));
        Assert.That(destFile.Item.Versions.Count, Is.EqualTo(3));

        Assert.That(Encoding.UTF8.GetString(destFile.Versions[0].OpenBinary()), Is.StringMatching("This is some
text"));
        Assert.That(Encoding.UTF8.GetString(destFile.Versions[1].OpenBinary()), Is.StringMatching("This is
version 2"));
        Assert.That(Encoding.UTF8.GetString(destFile.OpenBinary()), Is.StringMatching("This is version 3"));
    }
}
```

# What can we do? Onions/Testability

```
public class NotificationTimerJob : SPJobDefinition
{...

public override void Execute(Guid targetInstanceId)
{
    base.Execute(targetInstanceId);

    NotificationTimerJobSettings settings =
        NotificationTimerJobSettings.FromAdminWebUrl(
            WebApplication.Properties[Constant.Name].ToString());

    Notifier.Notify(
        settings,
        new ActualRecipientProvider(settings),
        new ActualRecipientProcessor(settings));
}
```

# What can we do? Onions/Testability

```
public static void Notify(  
    NotificationTimerJobSettings settings,  
    IRecipientProvider recipientProvider,  
    IRecipientProcessor recipientProcessor)  
{  
    Logger logger = new Logger(settings);  
    logger.Log("Notify called");  
  
    DateTime startTime = DateTime.Now;  
    List<Exception> errors = new List<Exception>();  
  
    if (settings.RunTimerJob)  
    {  
        foreach (RecipientInfo info in  
            recipientProvider.GetUnconfirmedRecipients())  
        {  
            try  
            {  
                logger.Log(String.Format("Processing recipient {0}", info));  
                recipientProcessor.ProcessRecipient(info);  
            }  
            catch { }  
        }  
    }  
}
```

# JavaScript is also Programming

```
var docLibNameCtrl;
```

```
function TPARetrieveConfig() {  
    docLibNameCtrl = document.getElementById("destDocLibName");  
    docLibNameCtrl.value =  
configXml.selectSingleNode("/NWActionConfig/Parameters/Parameter[@Name='DestDocL  
ibName']/PrimitiveValue/@Value").text;  
}
```

```
function TPAWriteConfig() {  
  
configXml.selectSingleNode("/NWActionConfig/Parameters/Parameter[@Name='DestDocL  
ibName']/PrimitiveValue/@Value").text = docLibNameCtrl.value;  
  
    return true;  
}
```



# JavaScript is also Programming

```
var FIRSTQUAD = FIRSTQUAD || {}
```

```
function TPARetrieveConfig() {  
    FIRSTQUAD.docLibNameCtrl = document.getElementById("destDocLibName");  
    FIRSTQUAD.docLibNameCtrl.value =  
        configXml.selectSingleNode("/NWActionConfig/Parameters/Parameter[@Name='  
            DestDocLibName']/PrimitiveValue/@Value").text;  
}  
  
function TPAWriteConfig() {  
    configXml.selectSingleNode("/NWActionConfig/Parameters/Parameter[@Name='  
        DestDocLibName']/PrimitiveValue/@Value").text =  
        FIRSTQUAD.docLibNameCtrl.value;  
  
    return true;  
}
```

# Mysterious Things

Blog entry says do this and it works:

```
web.AllowUnsafeUpdates = true;
```

Hmmm...

# Mysterious Things

## SPWeb.AllowUnsafeUpdates Property



SharePoint 2010

[Other Versions](#) ▾

This topic has not yet been rated - [Rate this topic](#)

Gets or sets a Boolean value that specifies whether to allow updates to the database as a result of a GET request or without requiring a security validation.

**Namespace:** [Microsoft.SharePoint](#)

**Assembly:** Microsoft.SharePoint (in Microsoft.SharePoint.dll)

**Available in Sandboxed Solutions: Yes**

[Available in SharePoint Online](#)

### ▲ Syntax

C#

VB

```
public bool AllowUnsafeUpdates { get; set; }
```

### Property Value

Type: [System.Boolean](#)

**true** if unsafe updates are allowed; otherwise, **false**.

### ▲ Remarks

#### ⚠ Caution

Setting this property to **true** opens security risks, potentially introducing cross-site scripting vulnerabilities.

Thanks!

SharePoint for Thinking  
Developers in 5min

[simon.sebright@1stQuad.com](mailto:simon.sebright@1stQuad.com)

SharePoint Consultants & Developers

C:\Users\simon.sebright\whoami

You are Simon Sebright

An ACCU Member

A SharePoint Developer

Among other things

# SharePoint's 2 Faces

It promises everything (tick tick tick)

It doesn't do it very well OOTB

=>

The need for Services

=>

But they think they should already have it!

# SharePoint's 2 Faces

Project Size 3-100 days

Typically 10-20 days

Service Business =>

No budget left? No refactoring! No Test After!

# Random Interjection #105



Why can't I edit my code if the TFS is not available?



# Random Interjection #43



+41791291405

Please enter a valid number.

Since when were phone numbers numbers?

Too much time on one's hands can be the  
source of evil...