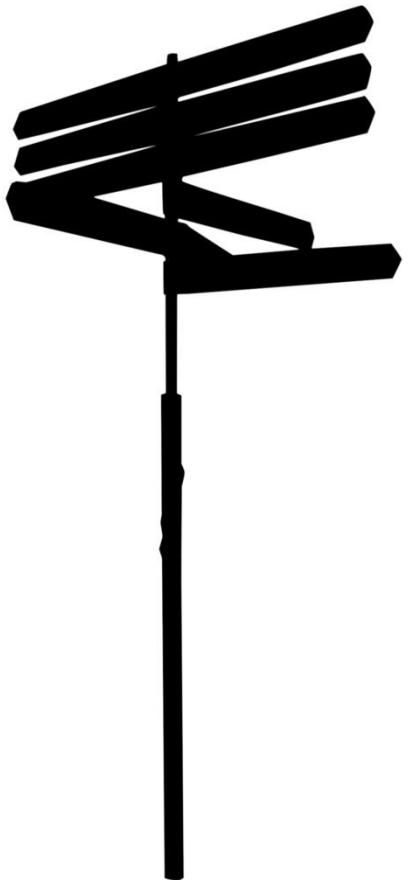
Lessons From a DevOps Journey

Matt Callanan

mattcallanan.net/devops matt@mattcallanan.net @mcallana

Outline

- Background & History
- Ecosystem Description
- Automation Examples
- Lessons Learnt



What is DevOps?

- Patrick Debois, 2009
  - DevOps = collaboration
  - *"DevTestSecurityOpsBusiness"*?
- CAMS (John Willis & Damon Edwards)
  - Culture : Automation : Measurement : Sharing

- #devops

Background & History

Enterprise Background

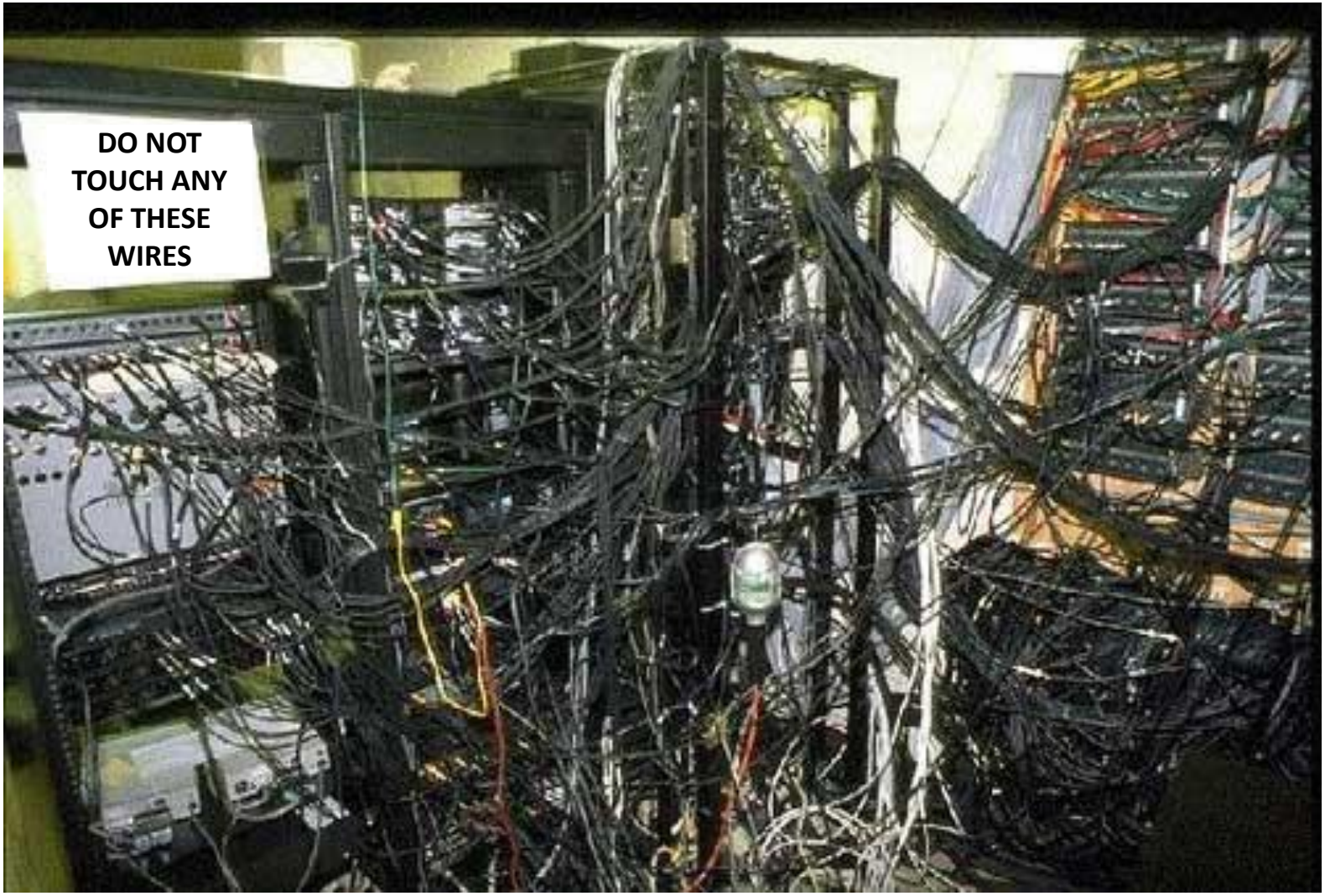
- 1,000+ IT personnel
- Many possible engagements between dev & infrastructure
- Can’t have a single team of 1,000+ people!
- Each dev team needs to find a way to work with a subset of infrastructure personnel

Treasury Department

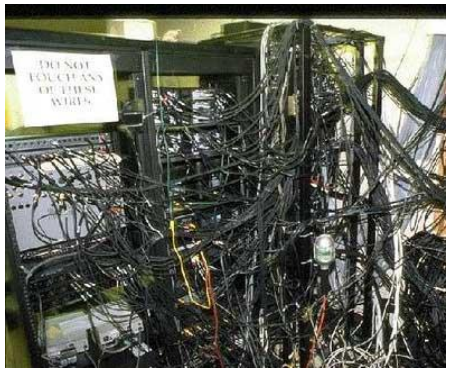
Mini trading floor

- Mission critical operations
- 150 end users across:
  - Front Office (Dealers/Sales)
  - Middle Office (Risk)
  - Back Office (Accounting/Settlement)
- Requires complex, specialist software

Good Old Days



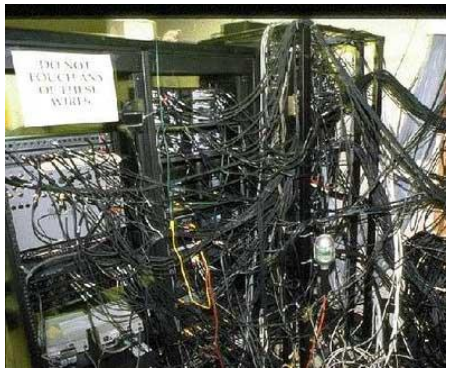
Good Old Days



Dodgy Environments

- Built in a piecemeal fashion
- Inconsistent environment setup
  - Nobody knew what state all the dev regions were in
  - Couldn’t rebuild new environment
- Maintenance nightmare
- Difficulty with:
  - Batch, backups, start/stop, organising daytime jobs
- Much on-call support required

Good Old Days



Disparate Interfaces

- Different languages/styles
- Obscure SQL code embedded in VB
- Poorly maintained
- Built up over 10 years
- No knowledge of change impact
  - what havoc will this change wreak?
- Inconsistent version control - Lost source code!
- Dodgy practices
  - Patching fixes directly in production

Treasury Upgrade Project

- Installation automation
- Interface replacement
- Data migration

Treasury Upgrade Project

Statistics:

18 **months**
30 **people**
27 **iterations**, 900 **Story Cards**
9 **releases** (to Business Testing)
40 vendor software **drops**
13 **environments**

Typical Vendor Products

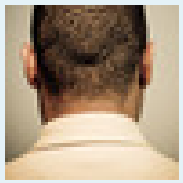
- Large installation manuals
- Complex manual installation procedures
- Vendor deployed onsite for months/years
- One-time install – not repeatable

DevOps Journey

- Started 4 years earlier
- Agile replacement project:
  - automated install
  - interface replacement
    - Created open-source EAI/ETL library
  - data migration

- Adventures of @WaterfallDave

Adventures of @WaterfallDave

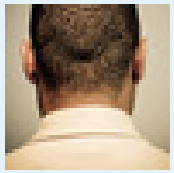


waterfalldave Waterfall Dave

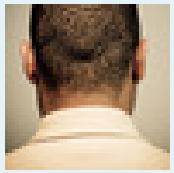
Hi my name is Dave. I'm still highly skeptical of many agile claims, so I can't show you my face.

27 Apr 09 ☆ Favorite ↻ Retweet ↩ Reply

Adventures of @WaterfallDave



waterfalldave Waterfall Dave
was overhead saying "Some aspects of Agile enabled this to work" after the automatic installation of a system previously done manually
25 Mar 09 ☆ Favorite ↻ Retweet ↩ Reply

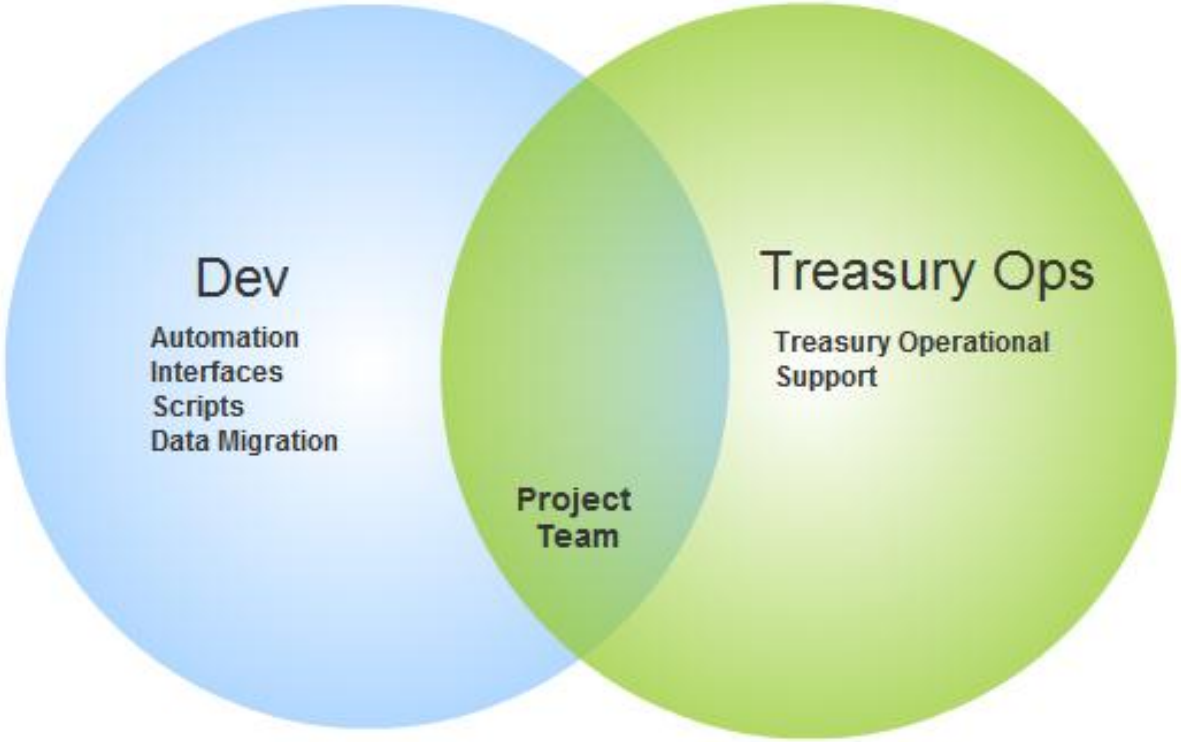


waterfalldave Waterfall Dave
realises the horrible truth. I have (well almost) returned to my team, bringing with me agile goodies. May god save us all!
31 May 09 ☆ Favorite ↻ Retweet ↩ Reply

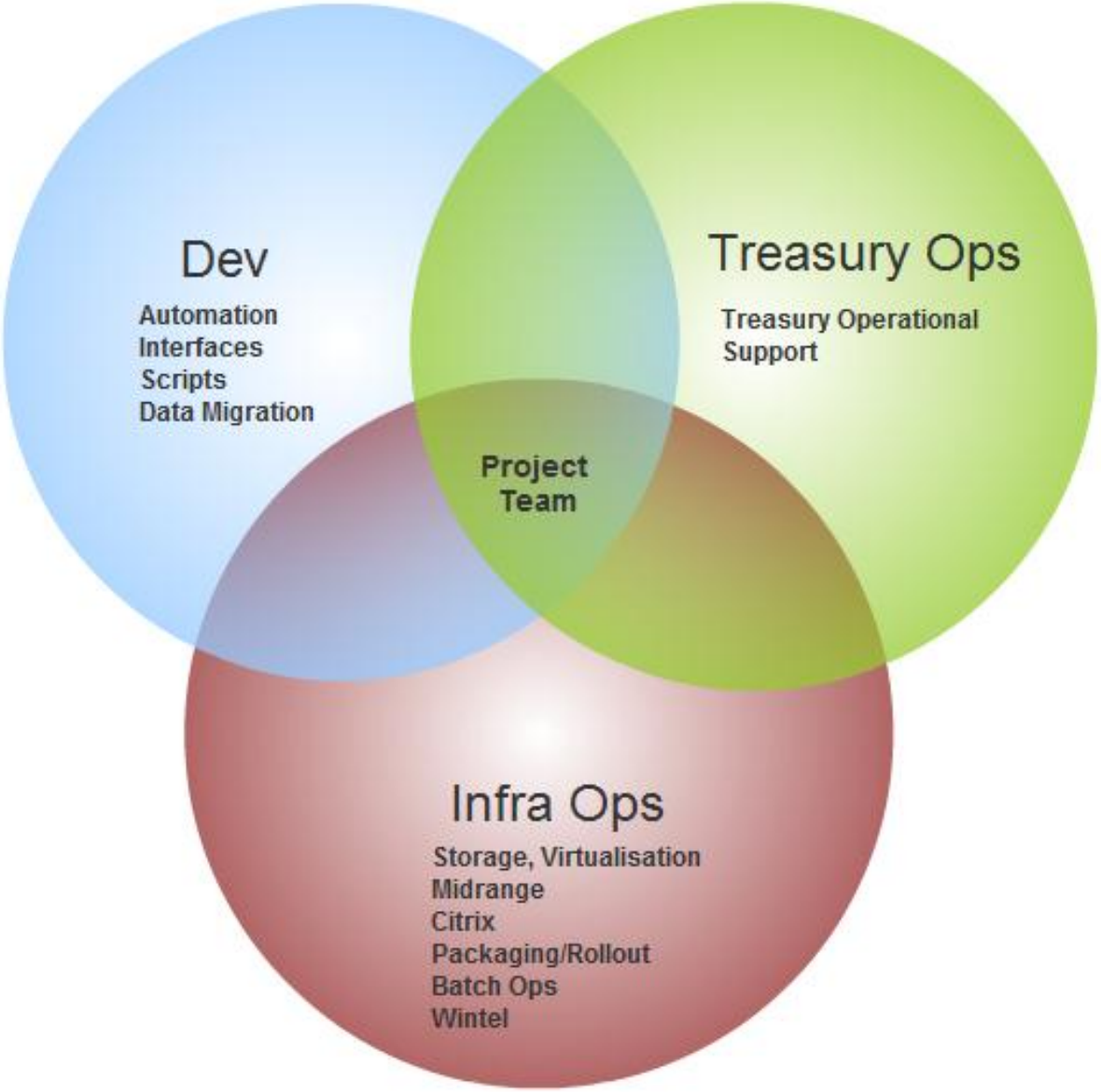
Treasury Upgrade Project @waterfalldave's Tech Choices

- Story cards
- Jira & Greenhopper
- Pairing
- Co-location
- Continuous Integration
- Standups, Retrospectives

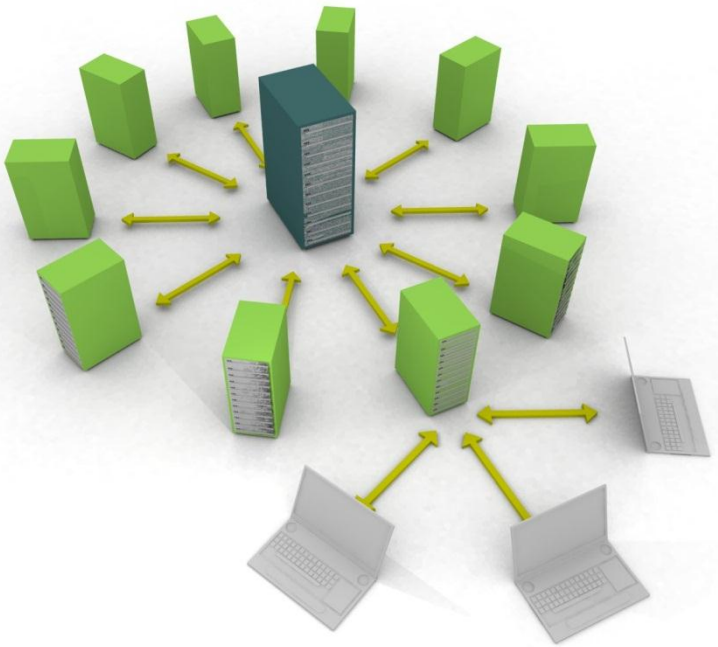
DevOps Team


Dev
Automation
Interfaces
Scripts
Data Migration
Project
Team
Treasury Ops
Treasury Operational
Support

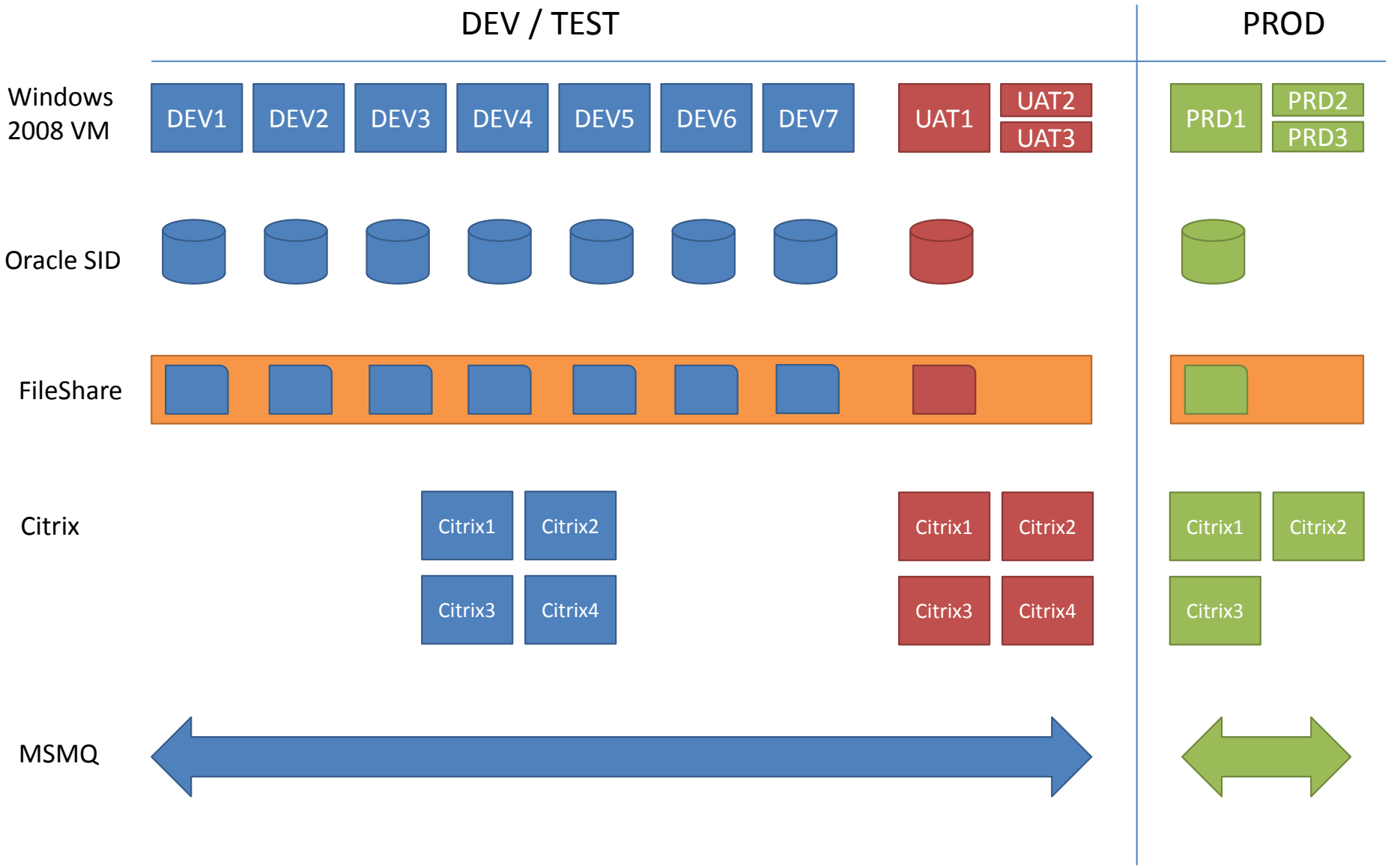
DevOps Team


Dev
Automation
Interfaces
Scripts
Data Migration
Treasury Ops
Treasury Operational
Support
Project
Team
Infra Ops
Storage, Virtualisation
Midrange
Citrix
Packaging/Rollout
Batch Ops
Wintel

Ecosystem

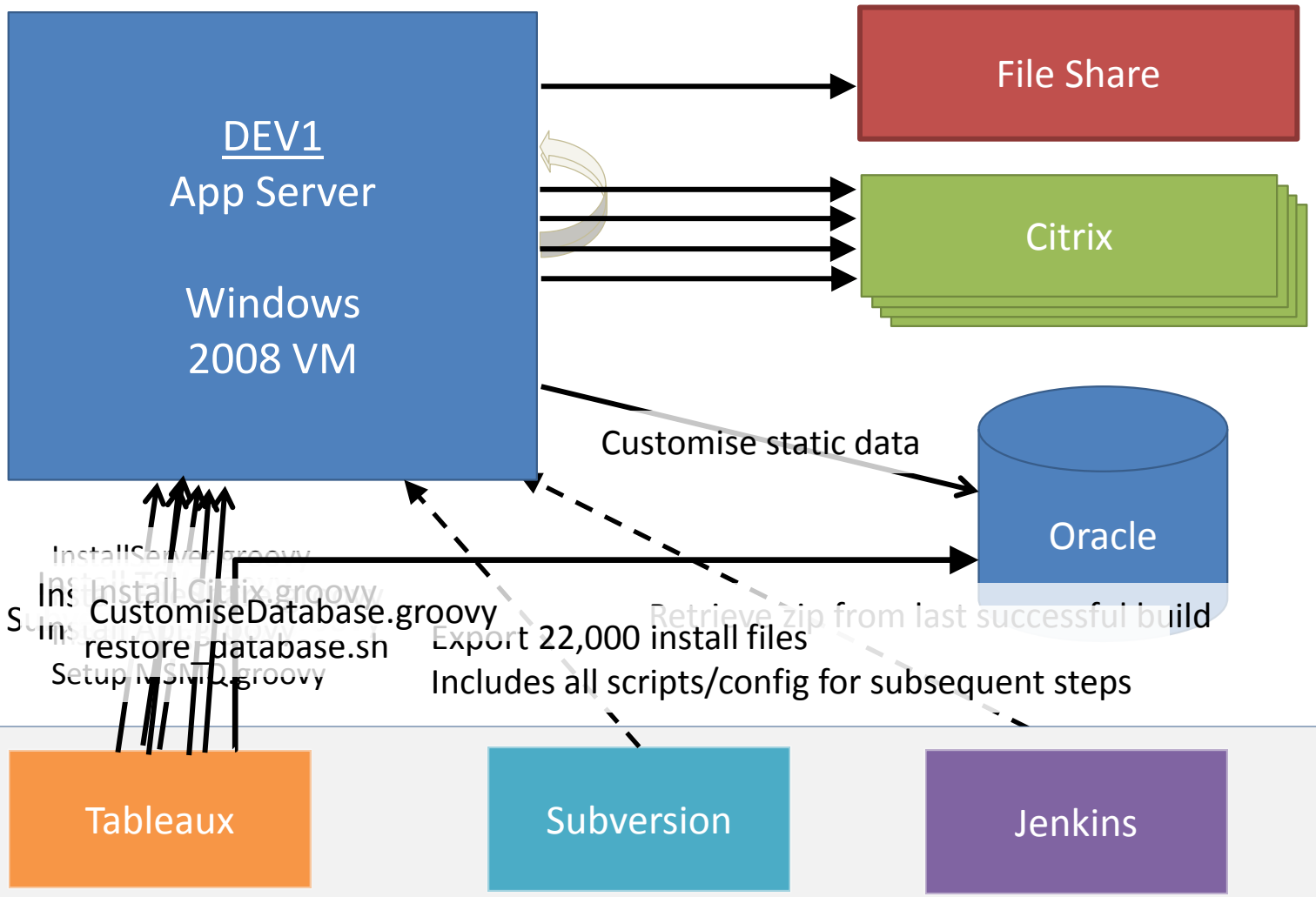


Ecosystem


DEV / TEST
PROD
Windows 2008 VM
DEV1
DEV2
DEV3
DEV4
DEV5
DEV6
DEV7
UAT1
UAT2
UAT3
PRD1
PRD2
PRD3
Oracle SID
FileShare
Citrix
Citrix1
Citrix2
Citrix3
Citrix4
Citrix1
Citrix2
Citrix3
Citrix4
Citrix1
Citrix2
Citrix3
MSMQ

Automation

Automated Deployment



Testing Your Automation

- Automation is great but does it still work?
- Wrap entire deployment process in a build and treat it like code:
  - Version control
  - Testable
  - Deployable
- “Infrastructure as code”

Continuous Integration

- Nightly build - rebuild entire test environment:
  - Deploy vendor and custom apps and configuration
  - Acceptance tests for vendor and custom apps
  - Batch run
  - Restore database ready for next day
- Approx. 4hrs

Continuous Integration

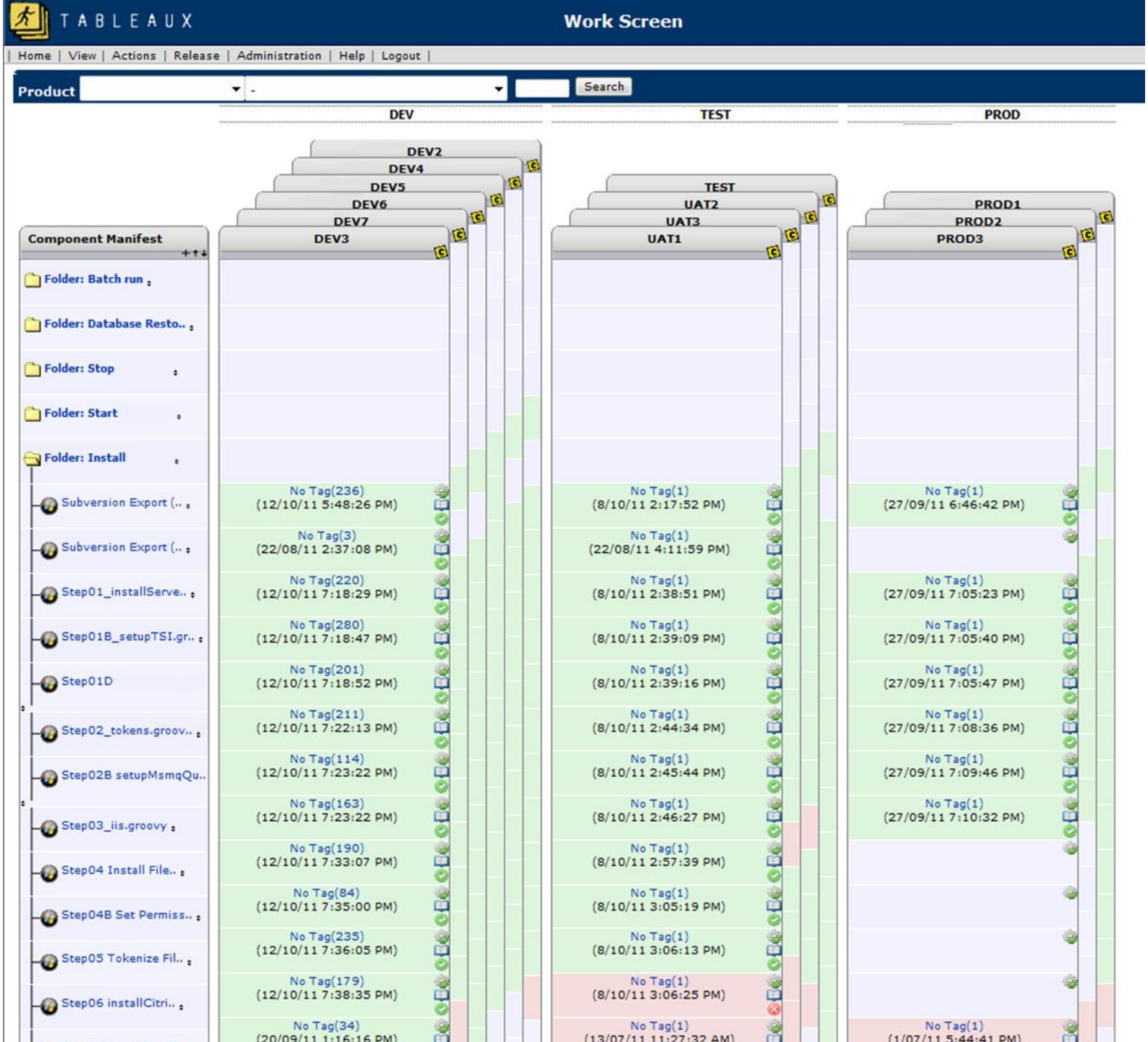
- Analyse the results in the morning…
  - anything broken from unintentional side-effects introduced day before?

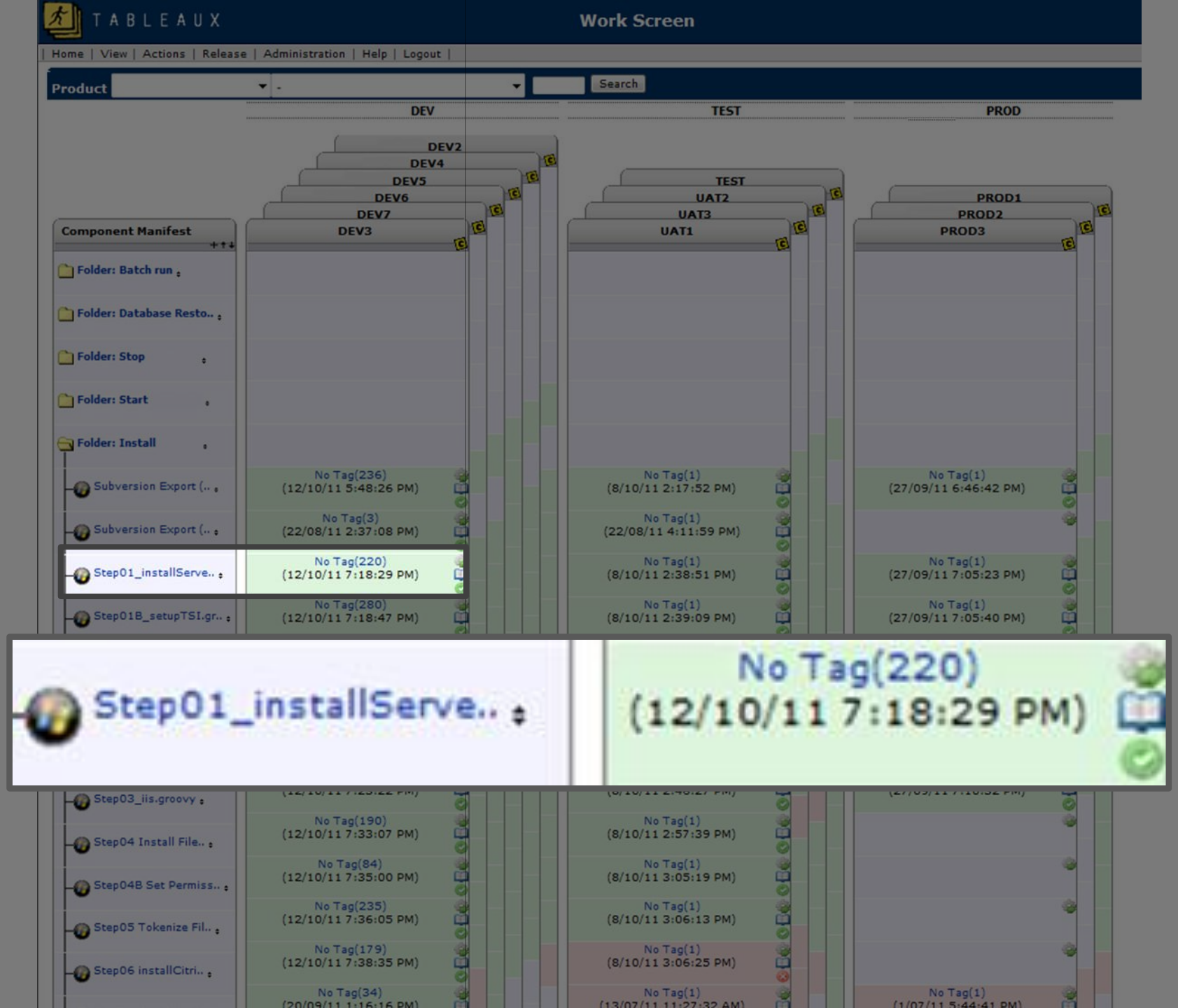
Tableaux Report

Stop Services	5:35:41 PM	50s	
Stop IIS	5:36:31 PM	5s	
Subversion Export (to latest)	5:36:37 PM	11m, 48s	
Restore all	5:48:26 PM	56m, 6s	
Generify Database	6:44:32 PM	41s	
Run treasury-tsi Build	6:45:14 PM	15m, 49s	
Run treasury-tsi-dist Build	7:01:03 PM	4m, 18s	
Run treasury-api-dist Build	7:05:21 PM	48s	
Step01_installServer.groovy	7:06:10 PM	12m, 19s	
Step01B_setupTSI.groovy	7:18:29 PM	17s	
Step01D_setupApi.groovy	7:18:47 PM	5s	

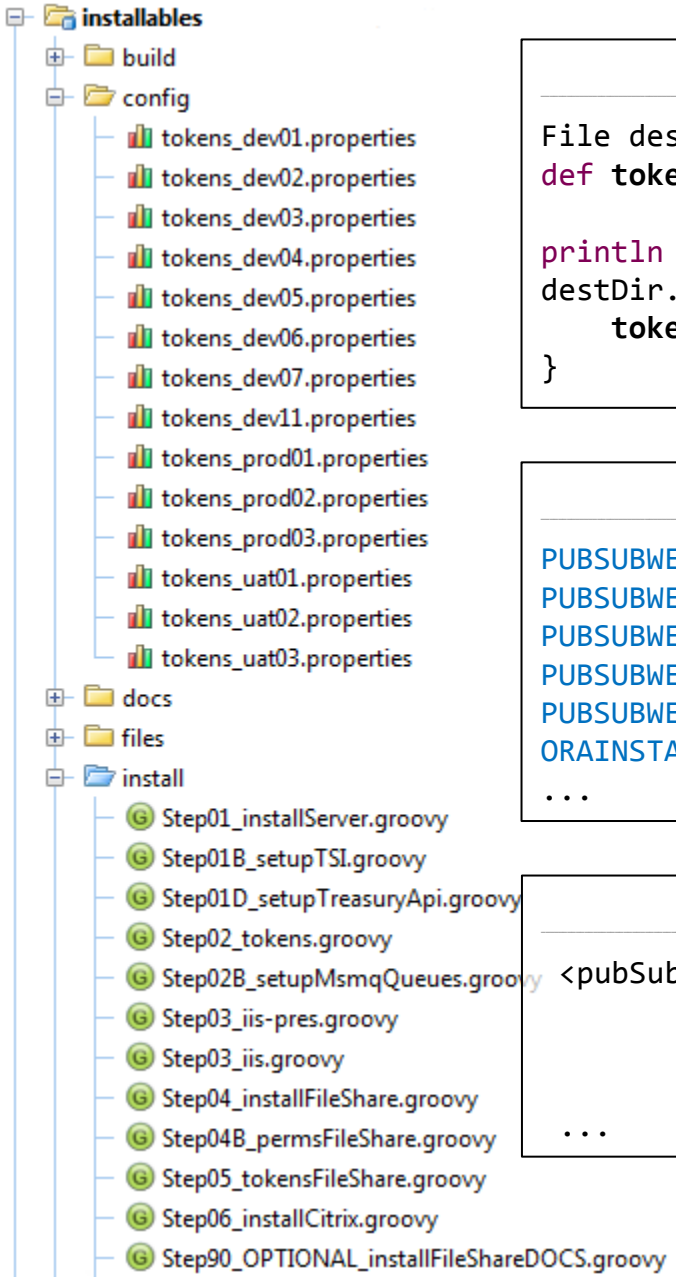
Tableaux Report

TSI Distribution Tests	7:38:35 PM	1m, 8s	✓
Start IIS	7:39:44 PM	4s	✓
Start Services	7:39:48 PM	2m, 55s	✓
EOD Batch Run	7:42:44 PM	57m, 0s	✓
Api Acceptance Tests	8:39:44 PM	11m, 27s	✓
Stop Services	8:51:12 PM	57s	✓
Stop IIS	8:52:10 PM	7s	✓
Restore all	8:52:17 PM	55m, 9s	✓
Generify Database	9:47:27 PM	36s	✓
Start Services	9:48:03 PM	31s	✓
Start IIS	9:48:35 PM	3s	✓


TABLEAUX
Work Screen
Home | View | Actions | Release | Administration | Help | Logout
Product
Search
DEV
TEST
PROD
DEV2
DEV4
DEV5
DEV6
DEV7
DEV3
TEST
UAT2
UAT3
UAT1
PROD1
PROD2
PROD3
Component Manifest
Folder: Batch run
Folder: Database Resto..
Folder: Stop
Folder: Start
Folder: Install
Subversion Export (..
Subversion Export (..
Step01_installServe..
Step01B_setupTSI.gr..
Step01D
Step02_tokens.groov..
Step02B setupMsmqQu..
Step03_iis.groovy
Step04 Install File..
Step04B Set Permiss..
Step05 Tokenize Fil..
Step06 installCitri..
No Tag(236) (12/10/11 5:48:26 PM)
No Tag(3) (22/08/11 2:37:08 PM)
No Tag(220) (12/10/11 7:18:29 PM)
No Tag(280) (12/10/11 7:18:47 PM)
No Tag(201) (12/10/11 7:18:52 PM)
No Tag(211) (12/10/11 7:22:13 PM)
No Tag(114) (12/10/11 7:23:22 PM)
No Tag(163) (12/10/11 7:23:22 PM)
No Tag(190) (12/10/11 7:33:07 PM)
No Tag(84) (12/10/11 7:35:00 PM)
No Tag(235) (12/10/11 7:36:05 PM)
No Tag(179) (12/10/11 7:38:35 PM)
No Tag(34) (20/09/11 1:16:16 PM)
No Tag(1) (8/10/11 2:17:52 PM)
No Tag(1) (22/08/11 4:11:59 PM)
No Tag(1) (8/10/11 2:38:51 PM)
No Tag(1) (8/10/11 2:39:09 PM)
No Tag(1) (8/10/11 2:39:16 PM)
No Tag(1) (8/10/11 2:44:34 PM)
No Tag(1) (8/10/11 2:45:44 PM)
No Tag(1) (8/10/11 2:46:27 PM)
No Tag(1) (8/10/11 2:57:39 PM)
No Tag(1) (8/10/11 3:05:19 PM)
No Tag(1) (8/10/11 3:06:13 PM)
No Tag(1) (8/10/11 3:06:25 PM)
No Tag(1) (13/07/11 11:27:32 AM)
No Tag(1) (27/09/11 6:46:42 PM)
No Tag(1) (27/09/11 7:05:23 PM)
No Tag(1) (27/09/11 7:05:40 PM)
No Tag(1) (27/09/11 7:05:47 PM)
No Tag(1) (27/09/11 7:08:36 PM)
No Tag(1) (27/09/11 7:09:46 PM)
No Tag(1) (27/09/11 7:10:32 PM)
No Tag(1) (1/07/11 5:44:41 PM)


TABLEAUX
Work Screen
Home | View | Actions | Release | Administration | Help | Logout
Product
Search
DEV
TEST
PROD
DEV2
DEV4
DEV5
DEV6
DEV7
DEV3
TEST
UAT2
UAT3
UAT1
PROD1
PROD2
PROD3
Component Manifest
Folder: Batch run
Folder: Database Resto..
Folder: Stop
Folder: Start
Folder: Install
Subversion Export (..
No Tag(236) (12/10/11 5:48:26 PM)
No Tag(1) (8/10/11 2:17:52 PM)
No Tag(1) (27/09/11 6:46:42 PM)
Subversion Export (..
No Tag(3) (22/08/11 2:37:08 PM)
No Tag(1) (22/08/11 4:11:59 PM)
Step01_installServe..
No Tag(220) (12/10/11 7:18:29 PM)
No Tag(1) (8/10/11 2:38:51 PM)
No Tag(1) (27/09/11 7:05:23 PM)
Step01B_setupTSI.gr..
No Tag(280) (12/10/11 7:18:47 PM)
No Tag(1) (8/10/11 2:39:09 PM)
No Tag(1) (27/09/11 7:05:40 PM)
Step01_installServe..
No Tag(220)
(12/10/11 7:18:29 PM)
Step03_iis.groovy
No Tag(190) (12/10/11 7:33:07 PM)
No Tag(1) (8/10/11 2:57:39 PM)
Step04 Install File..
No Tag(84) (12/10/11 7:35:00 PM)
No Tag(1) (8/10/11 3:05:19 PM)
Step04B Set Permiss..
No Tag(235) (12/10/11 7:36:05 PM)
No Tag(1) (8/10/11 3:06:13 PM)
Step05 Tokenize Fil..
No Tag(179) (12/10/11 7:38:35 PM)
No Tag(1) (8/10/11 3:06:25 PM)
Step06 installCitri..
No Tag(34)
No Tag(1)

Tokenisation



Step02_tokens.groovy

```
File destDir = new File(/C:\Program Files\Server/)
def tokens = new Tokens(args)

println '----- Tokenising config files'
destDir.traverse(filter: {it.name ==~ ~/.*\.(config|Config|xml)/}) { file->
    tokens.processFile(file, tokens.tokenise)
}
```

tokens_dev03.properties

```
PUBSUBWEBSRVIP=TREAS3
PUBSUBWEBSRVPRT=47125
PUBSUBWEBSRVPRTSTART=49000
PUBSUBWEBSRVIP=TREAS3
PUBSUBWEBSRVPRT=47125
ORAINSTANCE=DEV3
...
```

Application.config

```
<pubSubHost IntervalInSecondsToCheckHeartbeat="60"
            Address="net.tcp://%PUBSUBWEBSRVIP%:%PUBSUBWEBSRVPRT%/...
            Start_Port_Range="%PUBSUBWEBSRVPRTSTART%"
            End_Port_Range="%PUBSUBWEBSRVPRTEND%"/>
 ...
```

Tokens.groovy

```groovy
class Tokens {
    @Lazy tokens = loadProperties("../config/tokens_${System.getenv('ComputerName')}.properties")

    def processFile(file, Closure... closures) {
        def result = file.readLines().collect { line ->
            def newline = line
            closures.each { newline = it(file, newline) }
            newline
        }
        file.text = result.join('\n').denormalize()
    }

    def tokenise = { file, line ->
        def newline = line
        tokens.each { k, v ->
            newline = newline.replaceAll("%$k%", v)
        }
        return newline
    }

    def propertyMissing(name) {
        tokens[name]
    }

    Map loadProperties(String name) {
        def props = new Properties()
        new File("${scriptDir}/../config/${name}.properties").withReader { r -> props.load(r) }
        props
    }
}
```

Web Server (IIS)
Install Script

1. Setup SSL certificates
2. Setup permissions
3. Reinstall application services
4. Configure services

SetupIIS.groovy

```
def tokens = new Tokens(args)
def serviceUsername = tokens.SERVICEUSERNAME
def servicePassword =  tokens.SERVICEPASSWORD

String hash = makeCertificate()
execute """ appcmd set site /site.name:"$SITENAME" /+bindings.[protocol='https',bindingInformation='*:443:'] """
execute """ netsh http add sslcert ipport=0.0.0.0:443 certhash=$hash appid={$IIS_APP_ID}"""
execute """ appcmd delete apppool -apppool.name:ApplicationPool """
execute """ appcmd add apppool -name:ApplicationPool -autoStart:true -managedPipelineMode:Integrated -processModel.

def services = [ ApplicationService:"ApplicationPool", SecureTokenService:"ApplicationPool", ServiceInterface:"Defa
services.each { name, poolname ->
  execute """ appcmd delete APP "$SITENAME/$name" """
  execute """ appcmd add APP /site.name:"$SITENAME" /path:"/$name" -physicalPath:"apppath\$name" -applicationPool:$
  execute """ appcmd set vdir "$SITENAME/$name/" -userName:$serviceUsername -password:$servicePassword """
}

execute (/ntrights -u $serviceUsername +r SeServiceLogonRight/)
execute (/SchedulerService.exe -uninstall/)
execute (/SchedulerService.exe -install/)
execute (/sc config Scheduler obj=$serviceUsername password="$servicePassword" start=demand/)

String makeCertificate() {
    def output = execute("certutil -store My Cert")
    if (output.contains('-store command FAILED')) {
        println "Making certificate"
        execute("""makecert.exe -a sha1 -n "CN=Cert" -sr LocalMachine -ss My -sky exchange -pe""")
        output = execute("certutil -store My Cert")
    }
    def hash
    output.find(/Cert Hash[^:]*: (.*)/){ hash = it[1].replaceAll(' ', '') }
    if (!hash) throw new RuntimeException("Something went wrong parsing certificate")
    println 'Using certificate: ' + hash
    hash
}
```

SetupIIS.groovy

```
String execute(command) {
  def proc = command.execute()
  def sout = new StringBuffer()
  def serr = new StringBuffer()
  proc.consumeProcessOutput(sout, serr)
  proc.waitFor()
  String result = ''
  if (sout) result += sout
  if (serr) result += 'ERROR:  ' + serr
  result
}
```

Setup new environment from scratch?

- Custom third-party software install package
  - Install wizard
  - Wiki page
- Fresh configuration

Installer


NSIS


Java

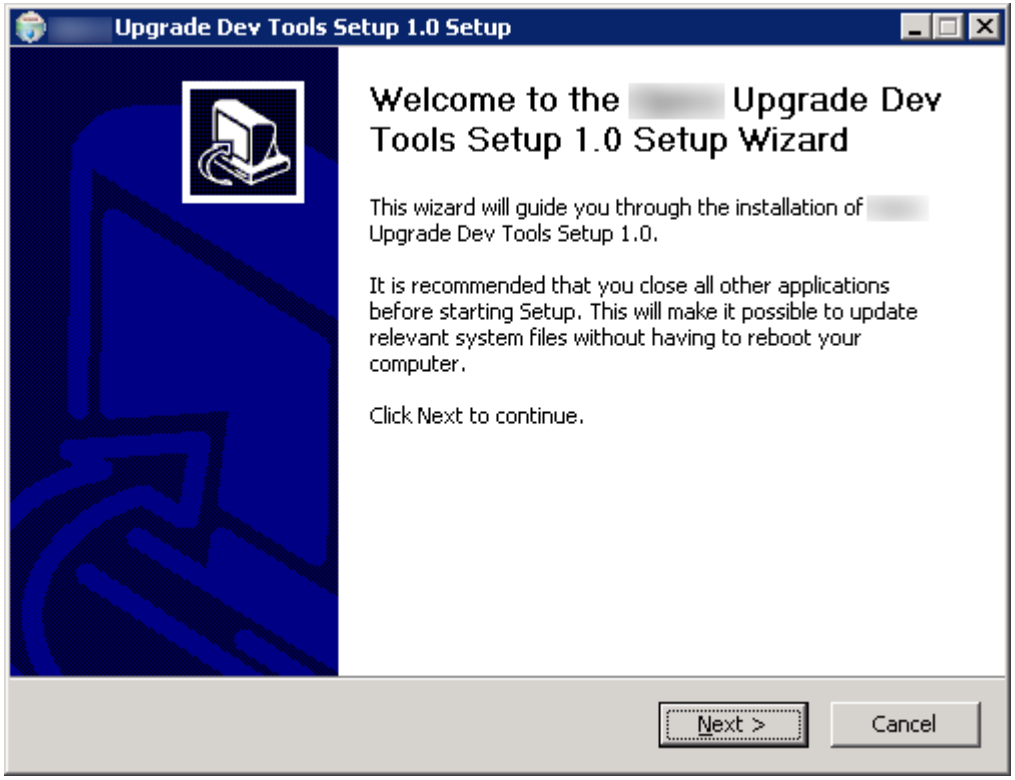

Groovy

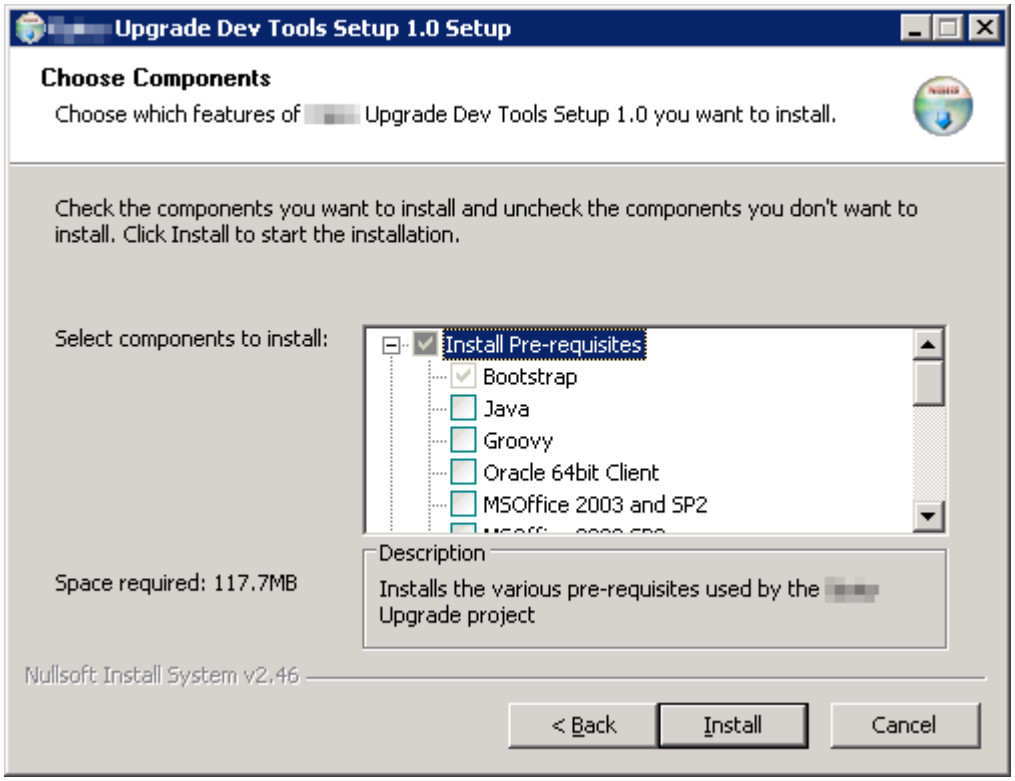

Microsoft
Office

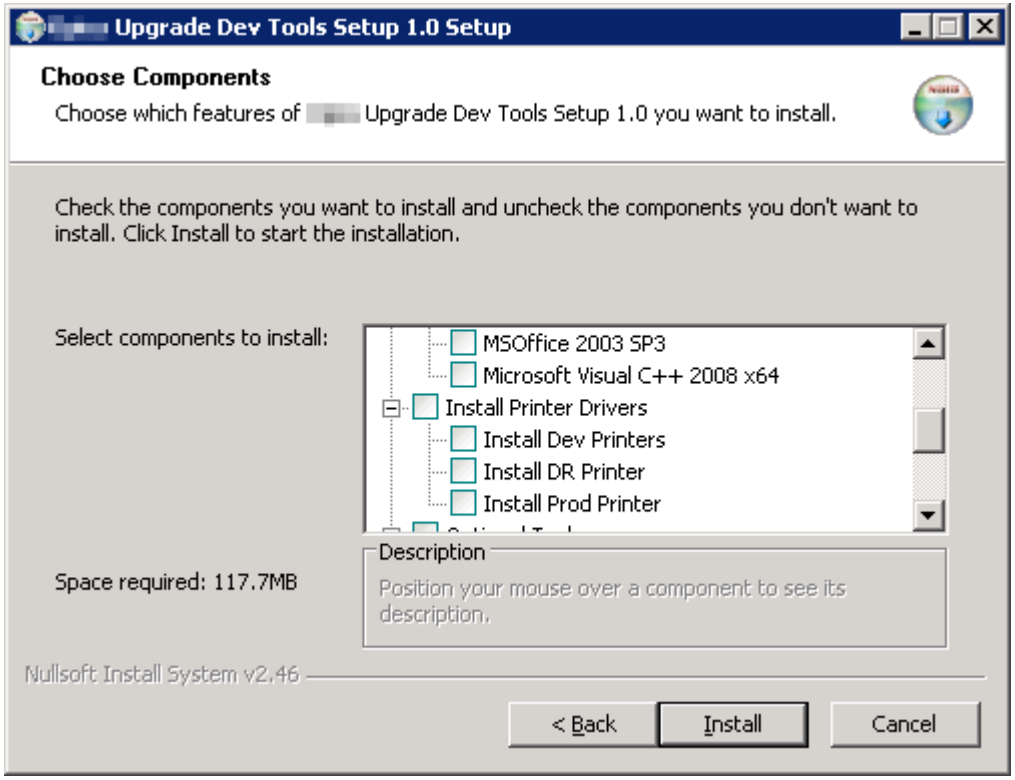

ORACLE


Microsoft
Visual Studio


LEXMARK


Upgrade Dev Tools Setup 1.0 Setup
Welcome to the Upgrade Dev Tools Setup 1.0 Setup Wizard
This wizard will guide you through the installation of Upgrade Dev Tools Setup 1.0.
It is recommended that you close all other applications before starting Setup. This will make it possible to update relevant system files without having to reboot your computer.
Click Next to continue.
Next >
Cancel


Upgrade Dev Tools Setup 1.0 Setup
Choose Components
Choose which features of Upgrade Dev Tools Setup 1.0 you want to install.
Check the components you want to install and uncheck the components you don't want to install. Click Install to start the installation.
Select components to install:
Install Pre-requisites
Bootstrap
Java
Groovy
Oracle 64bit Client
MSOffice 2003 and SP2
Description
Installs the various pre-requisites used by the Upgrade project
Space required: 117.7MB
Nullsoft Install System v2.46
< Back
Install
Cancel


Upgrade Dev Tools Setup 1.0 Setup
Choose Components
Choose which features of Upgrade Dev Tools Setup 1.0 you want to install.
Check the components you want to install and uncheck the components you don't want to install. Click Install to start the installation.
Select components to install:
MSOffice 2003 SP3
Microsoft Visual C++ 2008 x64
Install Printer Drivers
Install Dev Printers
Install DR Printer
Install Prod Printer
Description
Position your mouse over a component to see its description.
Space required: 117.7MB
Nullsoft Install System v2.46
< Back
Install
Cancel

NSIS Script

```
SectionGroup "Install Pre-requisites" SEC_GRP_INSTALL
  ...
  Section /o "Java" SEC_JAVA
    ClearErrors
    SetOutPath "${JAVA_INSTALL_DIR}"
    !insertmacro execAndCheck 'Java' '"${BOOTSTRAP}\${JAVA_EXE_NAME}" ADDLOCAL=ALL /log c:\temp\install-jc
    ; Set JAVA_HOME environment variable
    WriteRegExpandStr ${ENV_REGKEY} JAVA_HOME ${JAVA_INSTALL_DIR}
    ${EnvVarUpdate} $0 "PATH" "R" "HKLM"  "${JAVA_INSTALL_DIR}\bin"
    ${EnvVarUpdate} $0 "PATH" "P" "HKLM"  "${JAVA_INSTALL_DIR}\bin"
    Call notify_all_windows_of_environment_change
  SectionEnd
  ...
SectionGroupEnd

SectionGroup "Install Printer Drivers" SEC_GRP_PRINTER
  Section /o "Install Dev Printers" SEC_DEV_PRINTER
    ClearErrors
    !insertmacro svnExport "${FILES_REPO}/PrinterInstall" "${FILEDIR}/PrinterInstall"
    !insertmacro unzipAndCheck "${FILEDIR}\PrinterInstall\${PRINTER_DRIVER_ZIP}" "${TEMP_DIR}"
    !insertmacro execAndCheck 'Dev' '"groovy.exe" "installLexmarkPrinter.groovy" "LX656_TEXT" "c:\temp\LX
    !insertmacro execAndCheck 'Dev' '"groovy.exe" "installLexmarkPrinter.groovy" "LX656" "10.10.10.10"'
  SectionEnd
  ...
SectionGroupEnd

!macro svnExport Source Dest
    ExecWait '"${GROOVY_INSTALL_DIR}\bin\groovy.exe" "${BOOTSTRAP}\${SVN_EXPORTER}" ${Source} ${Dest}'
!macroend

!macro execAndCheck Description Executable
    DetailPrint 'Executing ${Description} (${Executable})'
    ExecWait '${Executable}'
!macroend

!macro unzipAndCheck Zipfile Dest
    !insertmacro ZIPDLL_EXTRACT "${Zipfile}" "${Dest}" "<ALL>"
!macroend
```

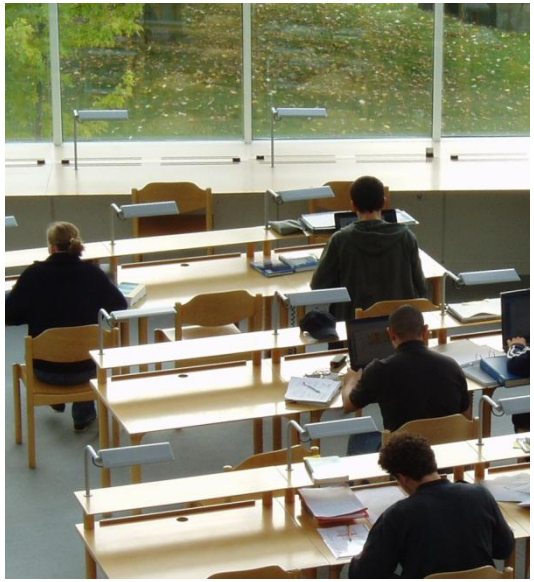
installLexmarkPrinter.groovy

```
printerName = args[0]
printerLocation = args[1]
checkDriverExists()
createPort()
installPrinter()
makeDefault()

def createPort() {
    printerLocationIsIpAddress() ? createTcpIpPort() : createLocalFilePort()
}

def createTcpIpPort() {
    def wmi = new ActiveXObject(/winmgmts:{impersonationLevel=impersonate}!\\.\root\CIMV2/)
    def newPort = wmi.Get("Win32_TCPIPPrinterPort").SpawnInstance_()
    newPort.Name = printerLocation
    newPort.Protocol = 1
    newPort.HostAddress = printerLocation
    newPort.PortNumber = "9100"
    newPort.SNMPEnabled = true
    newPort.Put_()
}

def createLocalFilePort() {
    "net stop spooler".execute()
    """reg add "HKEY_LOCAL_MACHINE\\SOFTWARE\\Microsoft\\Windows NT\\CurrentVersion\\Ports" /v "$printerL
    "net start spooler".execute()
}

def installPrinter() {
    "rundll32 printui.dll,PrintUIEntry /if /b \"$printerName\" /f C:\\temp\\LexmarkUniversalDrivers\\Prin
}

...
```

Lessons Learnt

Lessons Learnt

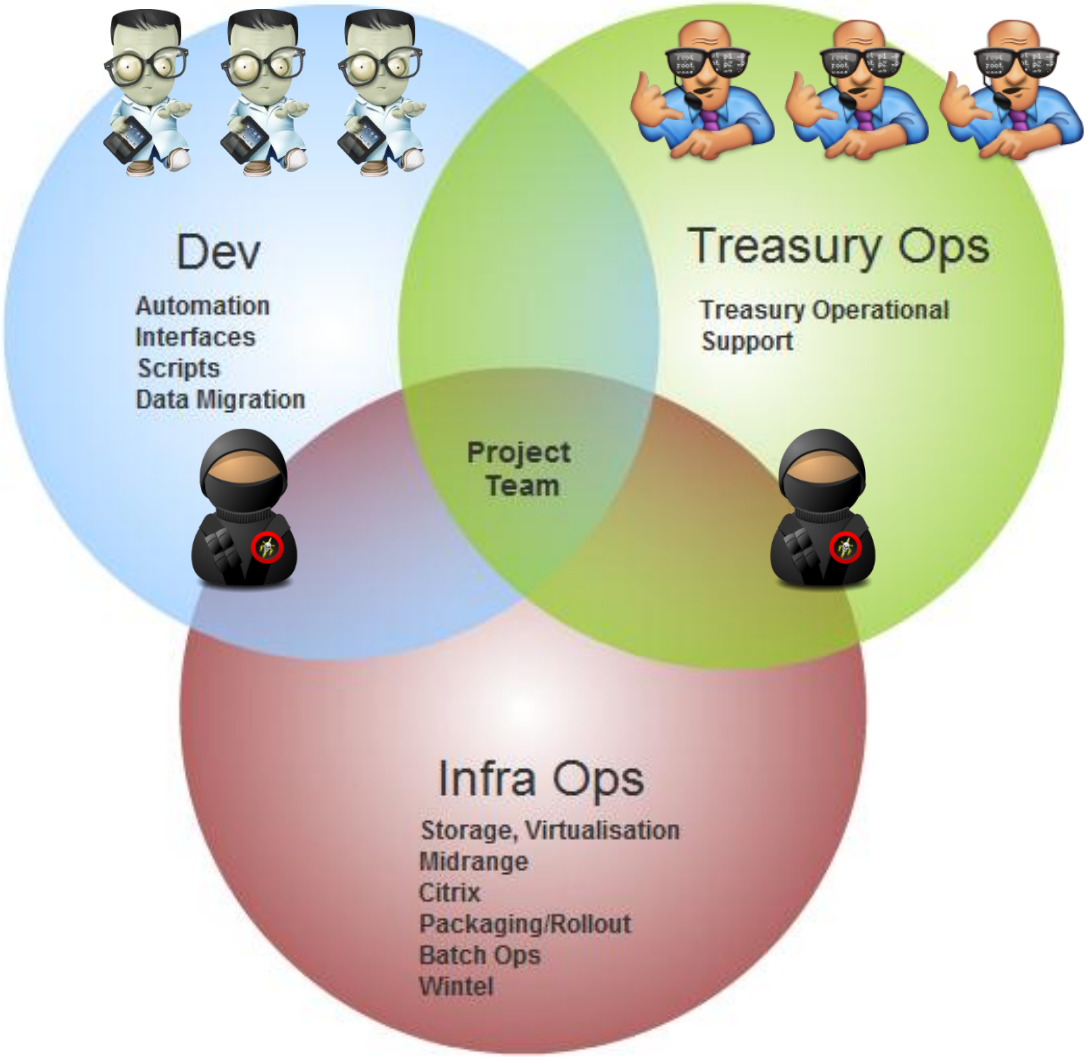
1. Get skills mix right
2. Co-locate where possible
3. Automation provides leverage
4. Fail early, fail often
5. Don’t need latest tools
6. Keep automation options open
7. Know when to automate



#1: Get Skills Mix Right

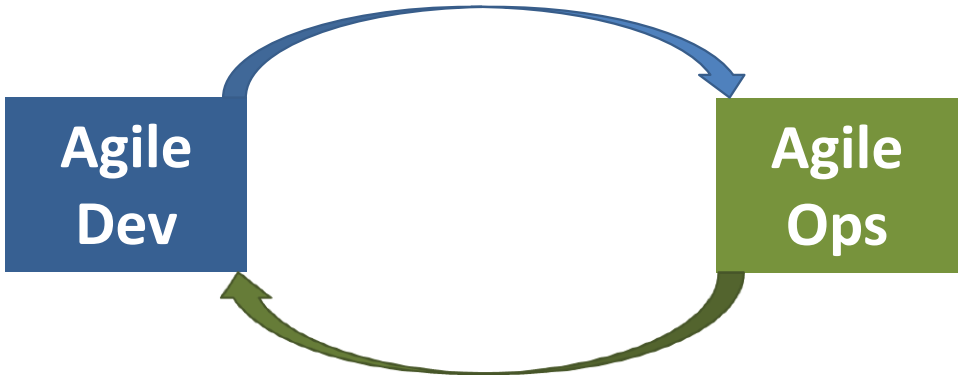
- Devs + Ops + Treasury experts + Environment experts
- Need experts from different disciplines
  - Brings balance to the force
  - A developer with a hammer sees every problem as a nail
- Pairing is not just for programming…
  - cross-skilling

Collaborative Team



DevOps Composition

- There's no one right way to compose your organization.



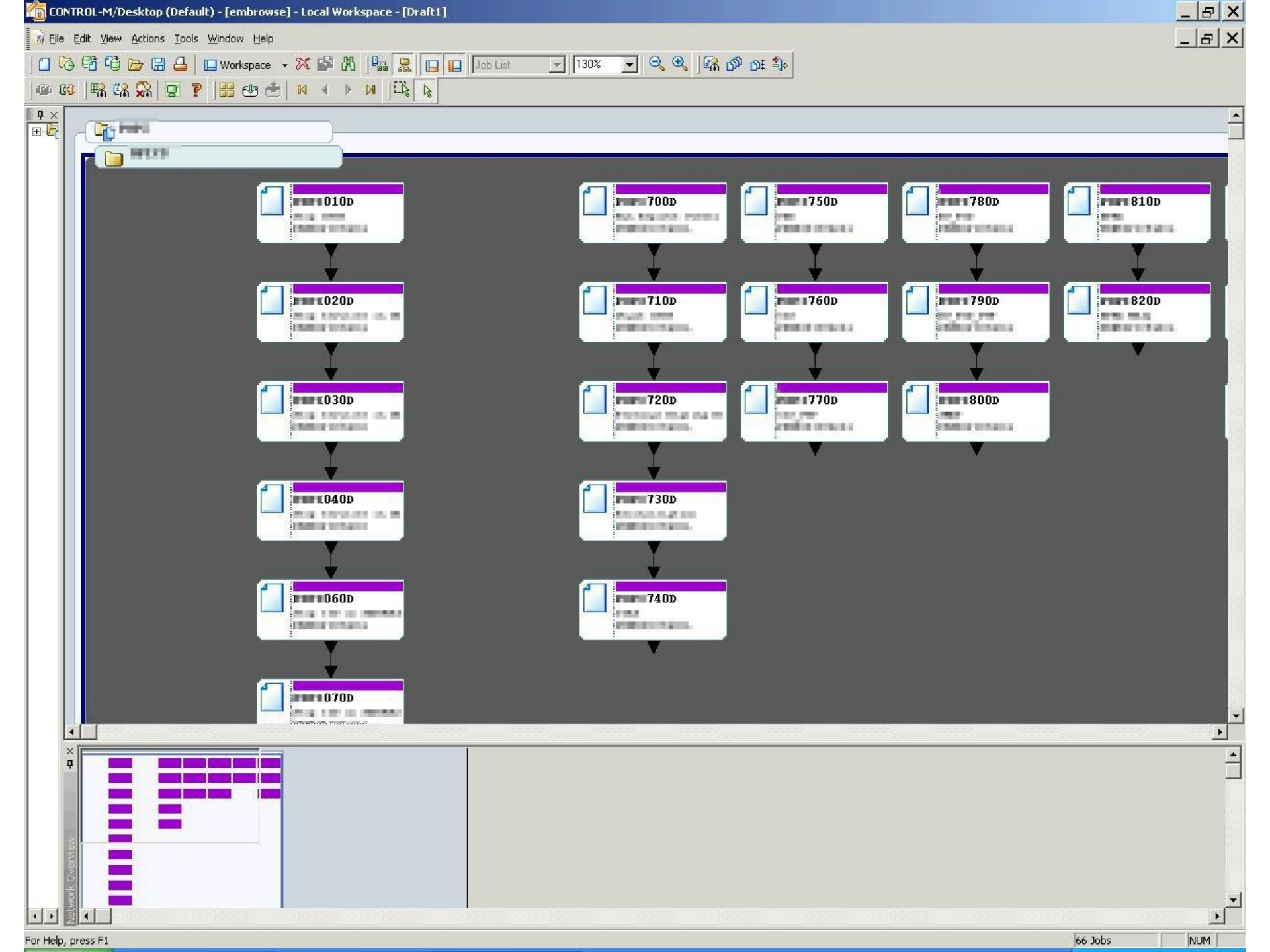
- DevOps Team?
- Continually seek to improve collaboration
  - whether in same team or different countries

#2: Co-locate as often as possible

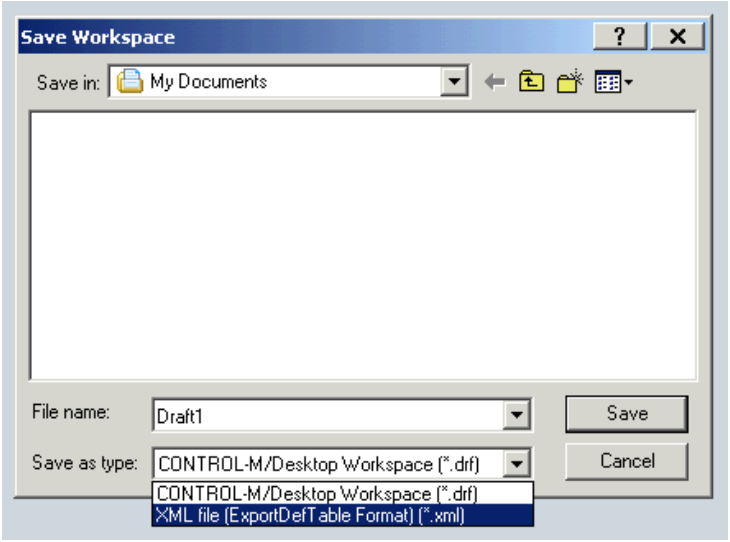
- Permanent co-location was not an option
- Need to find a way to break into silos
- Open invitation to infra guys:
  - Sit with team for ad-hoc problem solving
- Pick their brains
- Impress them with automation capabilities
- Gain their trust

Control-M Example

- Scheduling and monitoring tool
  - keeps production running 24/7
  - tedious manual GUI entry
- Manual setup would take 3-4 weeks
- Would “Huggy” embrace an automated solution?


CONTROL-M/Desktop (Default) - [embrowse] - Local Workspace - [Draft1]
File Edit View Actions Tools Window Help
Workspace
Job List
130%
010D
020D
030D
040D
060D
070D
700D
710D
720D
730D
740D
750D
760D
770D
780D
790D
800D
810D
820D
Network Overview
For Help, press F1
66 Jobs
NUM

Sample Control-M Job XML




```
<?xml version='1.0' encoding='ISO-8859-1' ?>
<!DOCTYPE DEFTABLE SYSTEM "deftable.dtd">
<DEFTABLE >
 <SCHED_TABLE DATACENTER="MIDRANGE_TEST">
  <JOB APPLICATION="TREAS" APR="1" AUG="1" ...
   CMDLINE="c:\Progra~1\..." CONFIRM="0" CREATION ...
   GROUP="TREAS" IND_CYCLIC="START" INTERVAL="00000M" JAN="1" JOBNAME="TREAS010
   PRIORITY="AA" RETRO="0" SEP="1" SHIFT="IGNOREJOB" SHIFTNUM="+00" SYSDB="0"
   <QUANTITATIVE NAME="AVAIL" QUANT="1"/>
   <SHOUT DEST="ECS" MESSAGE="%%JOBNAME ABENDED" URGENCY="R" WHEN="NOTOK"/>
   <ON CODE="NOTOK" STMT="*">
    <DOCOND NAME="P-FAILED" ODATE="ODAT" SIGN="ADD"/>
   </ON>
  </JOB>
 </SCHED_TABLE>
</DEFTABLE>
```

Data-driven Automation: Excel

Name	JOBNAME	SUFFIX	NODEID	OWNER	TIMEF	INCOND
SCHEDULE: Business Days|WEEKDAYS="1,2,3,4,5"						
Stop OSYS	TRES010	D	%LOCAL_SERVER%	%SERVICEUSERNAME2%	1815	
Stop services on SERVER1 before	TRES020	D	%LOCAL_SERVER%	%SERVICEUSERNAME2%		TRES010D-TRE
Stop services on SERVER2 before	TRES030	D	%PAIRED_SERVER_1%	%SERVICEUSERNAME2%		TRES020D-TRE
Stop services on SERVER3 before	TRES040	D	%PAIRED_SERVER_2%	%SERVICEUSERNAME2%		TRES030D-TRE
Stop IIS on SERVER1 before backu	TRES060	D	%LOCAL_SERVER%	%SERVICEUSERNAME2%		TRES040D-TRE
Stop IIS on SERVER2 before backu	TRES070	D	%PAIRED_SERVER_1%	%SERVICEUSERNAME2%		TRES060D-TRE
Stop IIS on SERVER3 before backu	TRES080	D	%PAIRED_SERVER_2%	%SERVICEUSERNAME2%		TRES070D-TRE
Database backup before batch	TRES100	D	%ORASERVER%	%ORAUSER%		TRES080D-TRE
Start IIS on SERVER1 after backup	TRES110	D	%LOCAL_SERVER%	%SERVICEUSERNAME2%		TRES100D-TRE
Start IIS on SERVER2 after backup	TRES120	D	%PAIRED_SERVER_1%	%SERVICEUSERNAME2%		TRES110D-TRE
Start IIS on SERVER3 after backup	TRES130	D	%PAIRED_SERVER_2%	%SERVICEUSERNAME2%		TRES120D-TRE
Start services on SERVER1 after b	TRES150	D	%LOCAL_SERVER%	%SERVICEUSERNAME2%		TRES130D-TRE
Start services on SERVER2 after b	TRES160	D	%PAIRED_SERVER_1%	%SERVICEUSERNAME2%		TRES150D-TRE
Start services on SERVER3 after b	TRES170	D	%PAIRED_SERVER_2%	%SERVICEUSERNAME2%		TRES160D-TRE
Kick off the EOD batch run	TRES190	D	%LOCAL_SERVER%	%SERVICEUSERNAME2%		TRES170D-TRE
Run Generate …	TRES200	D	%LOCAL_SERVER%	%SERVICEUSERNAME2%		TRES190D-TRE
Stop services on SERVER1 before	TRES210	D	%LOCAL_SERVER%	%SERVICEUSERNAME2%		TRES200D-TRE
Stop services on SERVER2 before	TRES220	D	%PAIRED_SERVER_1%	%SERVICEUSERNAME2%		TRES210D-TRE
Stop services on SERVER3 before	TRES230	D	%PAIRED_SERVER_2%	%SERVICEUSERNAME2%		TRES220D-TRE
Stop IIS on SERVER1 before backu	TRES250	D	%LOCAL_SERVER%	%SERVICEUSERNAME2%		TRES230D-TRE
Stop IIS on SERVER2 before backu	TRES260	D	%PAIRED_SERVER_1%	%SERVICEUSERNAME2%		TRES250D-TRE
Stop IIS on SERVER3 before backu	TRES270	D	%PAIRED_SERVER_2%	%SERVICEUSERNAME2%		TRES260D-TRE
Database backup after batch	TRES290	D	%ORASERVER%	%ORAUSER%		TRES270D-TRE

Control-M Job Generator: Groovy

```groovy
private String convert(File csvFile, String env) {
    def headerXml = """<?xml version='1.0' encoding='ISO-8859-1' ?><!DOCTYPE DEFTABLE SYSTEM
    def footerXml = """  </SCHED_TABLE></DEFTABLE>"""
    def templateXml = '''
    <JOB APPLICATION="TREAS" CMDLINE="${cmdLine}" ${confCal} DESCRIPTION="${description}"
       <ON CODE="NOTOK" STMT="*"><DOCOND NAME="P@%%JOBNAME-%%NODEID" ODATE="ODAT" SIGN="ADD"
    </JOB>'''
    Template template = new groovy.text.GStringTemplateEngine().createTemplate(templateXml)
    CSVReader reader = new au.com.bytecode.opencsv.CSVReader(new StringReader(csvFile.text),
    String[] cells
    output = headerXml
    while ((cells = reader.readNext()) != null)
        output += cellsToJobXml(template, cells, rowNum++, schedule, region)
    output += footerXml
}

private String cellsToJobXml(Template template, String[] cells, int rowNum, String schedule,
    parameters = [
            region: region, description: cells[0], jobName: cells[1] + cells[2],
            nodeId: cells[3], theOwner: cells[4], timeFrom: cells[5],
            inCond: cells[6] ? """<INCOND AND_OR="AND" NAME="${cells[6]}" ODATE="ODAT" />"""
            quantResource: """<QUANTITATIVE NAME="%%NODEID._AVAIL" QUANT="1"/>""" + "\n",
            outCond: reverseInCond(cells[6]) + buildOutConds(cells[7]),
            cmdLine: cells[8], confCal: getConfCal(cells[2]), shift: getShift(cells[2]),
            schedule: schedule
    ]
    template.make(parameters).toString()
}
```

Result XML

```
<?xml version='1.0' encoding='ISO-8859-1' ?>
<!DOCTYPE DEFTABLE SYSTEM "deftable.dtd">
<DEFTABLE >
  <SCHED_TABLE DATACENTER="MIDRANGE_TEST" LAST_UPLOAD="21/04/2011 6:01:23 AM" TABLE_NAME="
    <JOB APPLICATION="TREAS" APR="1" AUG="1" AUTHOR="ctmauth" AUTOARCH="0" CMDLINE="C:\Prog
       DAYS_AND_OR="AND" DEC="1" DESCRIPTION="Stop OSYS" FEB="1" GROUP="TREAS" JAN="1" JOBN
       MAXRUNS="0" MAXWAIT="98" MAY="1" MEMNAME="TREAS010D" MULTY_AGENT="N" NODEID="TREAS"
       TASKTYPE="Command" TIMEFROM="1815" USE_INSTREAM_JCL="N" WEEKDAYS="1,2,3,4,5" WEEKSCA
      <QUANTITATIVE NAME="%%NODEID._AVAIL" QUANT="1"/>
      <OUTCOND NAME="TREAS010D-TREAS020D" ODATE="ODAT" SIGN="ADD"/>
      <SHOUT DEST="ECS" MESSAGE="%%JOBNAME FAILED on %%NODEID - Call out to Treasury Suppor
      <ON CODE="NOTOK" STMT="*">
        <DOCOND NAME="P@%%JOBNAME-%%NODEID" ODATE="ODAT" SIGN="ADD"/>
      </ON>
    </JOB>
    <JOB APPLICATION="TREAS" APR="1" AUG="1" AUTHOR="ctmauth" AUTOARCH="0" CMDLINE="c:\prog
       DAYS_AND_OR="AND" DEC="1" DESCRIPTION="Stop services on SERVER1 before backup before
       MAXRUNS="0" MAXWAIT="98" MAY="1" MEMNAME="TREAS020D" MULTY_AGENT="N" NODEID="TREAS"
       TASKTYPE="Command" TIMEFROM="" USE_INSTREAM_JCL="N" WEEKDAYS="1,2,3,4,5" WEEKSCAL="P
      <INCOND AND_OR="AND" NAME="TREAS010D-TREAS020D" ODATE="ODAT" />
      <QUANTITATIVE NAME="%%NODEID._AVAIL" QUANT="1"/>
      <OUTCOND NAME="TREAS010D-TREAS020D" ODATE="ODAT" SIGN="DEL"/>
      <OUTCOND NAME="TREAS020D-TREAS030D" ODATE="ODAT" SIGN="ADD"/>
      <SHOUT DEST="ECS" MESSAGE="%%JOBNAME FAILED on %%NODEID - Call out to Treasury Suppor
      <ON CODE="NOTOK" STMT="*">
        <DOCOND NAME="P@%%JOBNAME-%%NODEID" ODATE="ODAT" SIGN="ADD"/>
      </ON>
    </JOB>
```

Would Huggy Go For It?

- We'd already saved a huge amount of time. But … would Huggy go for it?
- Collaboration approach
  1. Initially, just ask for advice – don't be arrogant
  2. Co-locate – collaborate in-person
  3. Test your assumptions, refine
  4. Impress
  5. Agree on way forward

Keys to Enterprise Engagement

- Engage early, engage often
- Face-to-face is still the best
  - One meet-and-greet can make a huge difference
- Pick your battles
  - Every other team in the organisation is vying for limited resources
  - Know what needs you have and what compromises the business can live with
- Not what you know – who you know

#3: Automation Provides Leverage

- Serendipity
- Automating one task often enables another task
  - Example: **Api** for batch run enabled ***testing***
  - Example: **Tokenisation** enabled ***installs, Control-M***

#4: Fail Early, Fail Often

- Practice, practice, practice
- Feedback is key
  - Execute ops code often
  - CI is not just for code
  - Reinstall entire vendor app every night in test environment
- Example: exercising DBA scripts every night

#5: You don't need the latest tools to get started

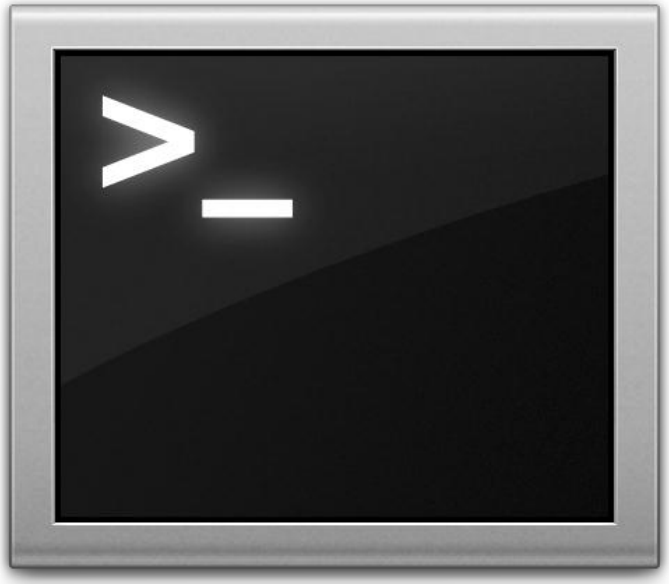
- Can get started with simple scripts and version control
  - shell, batch, powershell, groovy, ruby
- Reuse. What do you have in your hand right now?
  - IntelliJ / JUnit for Data Migration
- Tools do help…
- …but don’t procrastinate

Tool Choices

- Configuration Management (e.g. Puppet / Chef)
  - Declarative; Resource abstraction
  - Scalable; Platform independent
  - Reusable templates: PuppetForge; opscode-cookbooks
- Logging
  - Splunk , Logstash, GrayLog2, Flume
- Monitoring
  - "monitoring sucks"
  - Nagios, Sensu, Graphite, Ganglia
- IDEs
  - Puppet: Geppetto
  - Chef: RubyMine, Vim

#6: Keep automation options open

- Don’t let tool back you into a hole
  - E.g. Data migration got stuck inside IDE



- Make it reusable
  - Available from the command-line
  - No intervention

#7: Know when to automate

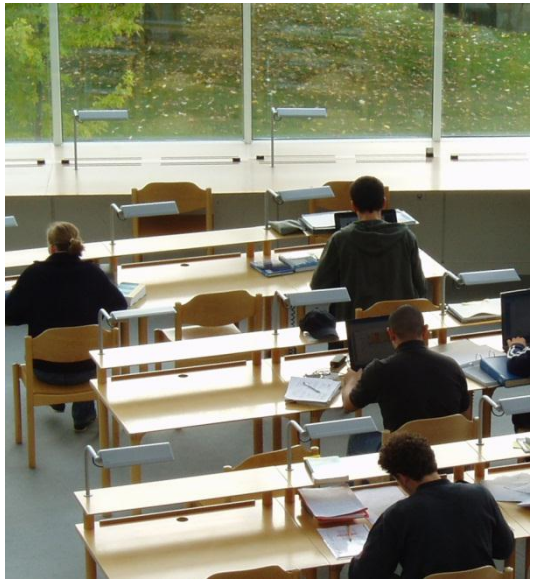
- Automation is not the goal
  - delivering business value sooner is the goal
- Gold-plated automation that's rarely used?
  - could be wasteful or could be indispensable…
  - depends on many factors: criticality / scalability / manual difficulty
- Over time, learn to trust gut-feel
  - Learning improves intuition
- If in doubt… automate
- Economies of Scale

Automation Trade-offs

- Questions to ask yourself
  - Complexity?
  - Criticality?
  - Cost?
  - Need to scale?
  - How often?

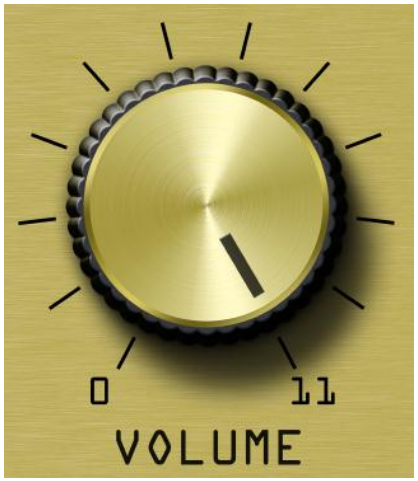
Lessons Learnt

1. Get skills mix right
2. Co-locate where possible
3. Automation provides leverage
4. Fail early, fail often
5. Don’t need latest tools
6. Keep automation options open
7. Know when to automate



DevOps = pushing Agile further

- We applied agile principles to “the last mile”
  - Culture of **collaboratively** seeking continual improvement
  - **Automating** fast feedback loops
- It’s an attitude
  - Can we get more efficient at this?
  - Can a computer do this for me?
  - How can we collaborate with that team and what can we learn from them?



Results of Journey

- Successful implementation
- Patches: 4-5days → 1day
- On-call support reduction
  - Including downstream ops
- Confidence in changes
  - Continuous integration
  - Reliable environment
- Automation / Business priorities = Win-Win

Done…
But The Journey Continues

Questions?