NEW HORIZON COLLEGE OF ENGINEERING

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

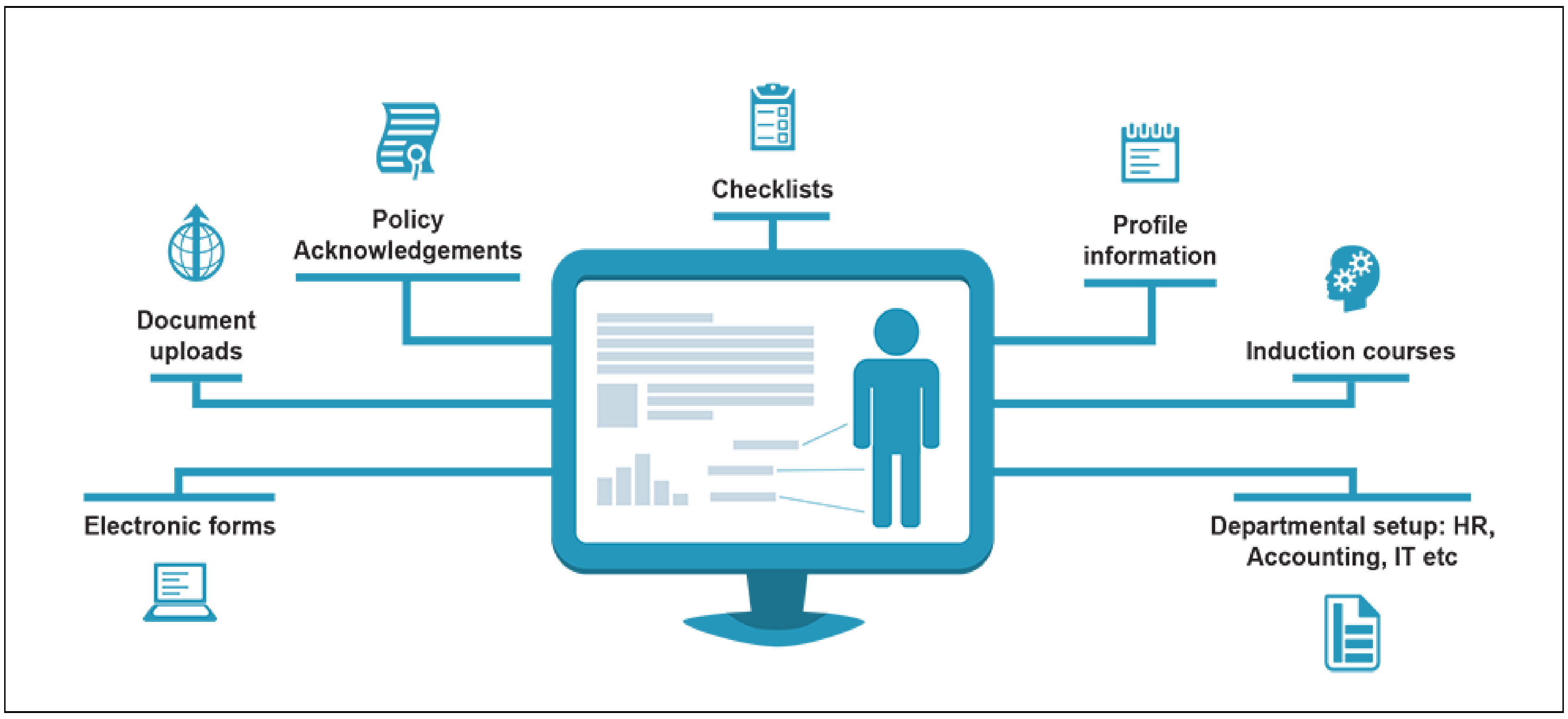
NEWSLETTER

BUSINESS AND INFORMATION TECHNOLOGY

BIT / Q4 / JUNE 2016

How IT Pros Can Automate the Employee Onboarding Process

Any growing organization hires new people; depending on size, possibly a lot of people. The employee onboarding process is typically cut-and-dried, the same across all employees. The process can be run off a checklist and followed the same way for every employee. If your organization has an IT staff, why not take advantage of their expertise, and automate as much of your onboarding process as possible?



Once employees are hired, the onboarding process usually includes the same basic needs for each employee:

- HR setup
- Employee badge
- Company key
- Computer and software
- Active Directory (AD) user account and security groups
- Email mailbox
- Home folder

And so on. Every employee's onboarding process is probably the same, making this kind of scenario ripe for automation. Unfortunately, the *entire* process cannot be automated, due to the physical nature of certain tasks, but an astute IT professional will automate wherever possible. Fully automating a process like this requires five steps:

1. Document the current manual process.
2. Eliminate tasks you can't automate.
3. Document each step in detail.
4. Script the process.
5. Document any manual processes.

Step 1: Document the Current Manual Process

When working on any automation project, we must first thoroughly understand the "moving parts" inside the current process and document everything as is. What does each function in the process do? How does each function relate to the other functions? This step is all about pounding the pavement, talking with people involved in the process (HR, IT, Operations, etc.), and gathering all the information you can about the process as a whole.

Step 2: Eliminate Tasks You Can't Automate

As we've already noted, IT professionals can't automate the entire onboarding process. For example, setting up the employee in the HR system is part of the Human Resources role. Operations probably assigns a badge/keycard or office key. But IT is likely to give new staffers their computers, get their Active Directory accounts set up, create mailboxes, and so on.

This step is all about eliminating tasks that IT staff can't automate, so they're not bogged down by details that aren't their responsibility.

Step 3: Document Each Step in Detail

Once you have a good picture of the entire process, and you've eliminated anything obvious that can't be automated, it's time to *thoroughly* document what is needed to accomplish each task in the process chain. What does *thoroughly* mean? Documenting at least these minimum questions and their answers:

- What kind of input is required for the task?
- What applications are used in the process?
- Who uses these applications?
- What screens are used in these applications? (Preferably documented with screenshots.)
- What is the desired result once the task is done? How is it presented?

I can't stress enough the level of detail required for each task if you want to succeed at automating *any* process—not just an employee onboarding process.

Step 4: Script the Process

Some people want to begin banging out code immediately, without going through the process I've outlined to this point, but I encourage you to resist that temptation. If you haven't followed steps 1–3 before you start coding, you'll be sorry. Your code will not account for numerous scenarios, and you'll have to put in Band-Aids and hacks, eventually turning your code into a giant ball of script that no one can understand. (Not even you.)

If you completed steps 1–3, however, you've defined the entire onboarding process and detailed what it currently takes for each task to happen. Great! The next step is to get scripting. But wait—what kind of scripting? Do you have Linux systems that require some kind of interaction in a task? You might have to learn Python. Perhaps your organization uses only Windows systems. In that case, Windows PowerShell is your go-to language. For this example, I'm assuming no Linux interaction, and concentrating solely on Windows PowerShell. My scripting process will consist of the following tasks:

1. Provision a new computer.
2. Install applications.
3. Create email mailboxes and Active Directory users.
4. Add user accounts to groups.
5. Create a home folder.

Task 1: Provision a New Computer

Most Windows-based organizations have Active Directory. A common task is adding a computer to the Active Directory domain and moving it into a particular organizational unit (OU). Rather than joining the computer to the domain and then manually moving it into the appropriate OU, why not pre-stage the computer account inside the correct OU before it's even joined?

With a single line in PowerShell, you can simply use the New-AdComputer cmdlet to add a computer into the HrDept OU, for example:

```
New-ADComputer -Name "HRCOMPUTER" -SamAccountName "HRCOMPUTER" -Path
"OU=HrDept,DC=domain,DC=local"
```

When the computer is delivered and joined to the domain, it will already be in the correct OU.

Task 2: Install Applications

PowerShell can automate application installations in many ways. Some modules are already in PowerShell, such as OneGet. Third-party products like Boxstarter can greatly speed up application installs. Once you have the applications documented in earlier steps, it's just a matter of deciding how the application installs will be chained together. Perhaps you have already created operating system images. In that case, install the applications onto the OS image and get it deployed.

Task 3: Create Email Mailboxes and Active Directory Users

If you're using a product like Microsoft Exchange, you can easily automate just about every part of the mailbox provisioning process. By using the Windows PowerShell New-Mailbox cmdlet, you can easily create a new Exchange mailbox *and* the corresponding new Active Directory user account at the same time. It's extremely convenient.

By using various PowerShell modules that come with products like Exchange, you can generally automate just about any manual task. For example:

```
$password = Read-Host "Enter password" -AsSecureString
New-Mailbox -UserPrincipalName chris@contoso.com -Alias chris
-Database "Mailbox Database 1" -Name ChrisAshton -OrganizationalUnit Users
-Password $password -FirstName Chris -LastName Ashton -DisplayName "Chris Ashton"
-ResetPasswordOnNextLogon $true
```

The PowerShell cmdlets that come with Microsoft Exchange are wonderful. Because Exchange was the first Microsoft product team to introduce PowerShell integration, these cmdlets are the most mature.

Task 4: Add User Accounts to Groups

Once an Active Directory account is created, a common task is to add that account to one or more Active Directory security groups. This process can sometimes be a pain in Active Directory Users and Computers (ADUC) because you have to find the user account, click around a little bit, find the right tab, and finally add one or more groups to the user account. Major waste of time. With PowerShell, you can use the Add-AdGroupMember cmdlet to add a user to multiple groups. The following example, adds newemployee to four different groups:

```
$Groups = 'group1','group2','group3','group4'
$Username = 'newemployee'
foreach ($Group in $Groups) {
Add-AdGroupMember -Identity $Group -Members $Username
}
```

Using simple PowerShell constructs is an easy way to add a user account to multiple different groups.

Task 5: Create a Home Folder

Finally, you can use PowerShell to create a home folder. Home folders are typically folders (located on a file server) that contain personal files for the employee. This common task can be performed easily with a scripting language like PowerShell; it's trivial, requiring just a few lines of PowerShell code.

In addition to creating the folder automatically, we can add various checks and error-control routines with the Test-Path cmdlet; for example, ensuring that the parent folder is available, the file server has enough space to accommodate the new employee's home folder, and so on:

```
$Username = 'newemployee'
$BaseHomeFolderPath = '\\fileserver\homefolders'

$HomeFolder = "$BaseHomeFolderPath\$Username"
if (!(Test-Path -Path $HomeFolder)) {
mkdir $HomeFolder | Out-Null
}

Set-AdUser -Identity $Username -HomeDrive 'H' -HomeDirectory $HomeFolder
```

This example checks whether a user's home folder exists. If not, the script creates it. It then ensures that the specified username is set for that home folder and maps to the H drive.

Step 5: Document Any Manual Processes

The rest is up to you. Since we can't automate every piece of the onboarding process, the next best thing is thoroughly documenting everything else that needs to happen. This list might include giving new employees office supplies, providing training, or whatever else is specific to your organization.

Final Thoughts

A set of dependent tasks with lots of "moving parts," like an employee onboarding process, can seem daunting. As an IT professional, you can't automate the entire process, but you can take responsibility for automating your part of the process. Follow the steps discussed here, and take these examples to heart. Take the time to fully understand all the "moving parts" and then automate everything you can. Automating not only prevents you from performing the same mundane processes all the time, but also helps your organization by saving your time. We all know that time equals money, and every organization likes saving money!

ACHIEVERS OF THE YEAR

USN	NAME	SEMESTER	EVENTS	COLLEGE	PRIZE
1NH12CS004 1NH12CS008 1NH12CS033	Abhishek YSS Amar Thapa Dinesh	VII	Paper Presentation	NHCE,Bangalore	I
1NH13CS083	Priyanka H	VI	Paper Presentation	RRIT,Bangalore	NA
1NH12CS015	Arvind Iyer	VII	Rock Jam	Christ University,Bangalore	I
1NH13CS048	Hema Chitrapu	V	Code 2 win	CBIT,Bangalore	II
1NH12CS037	Ganashree N	VII	Tech Quiz	NIT, Hyderabad	II
1NH13CS039	Divya Somarajan	V	Tech Quiz	TJIT,Bangalore	III
1NH13CS103	Sarah Biju	VI	Paper Presentation	RRIT,Bangalore	NA

ADVISORY COMMITTEE

Dr. Prashanth C.S.R., Dean Academics,
Head, Dept of CSE

Prof Anjana Sharma, Dept of CSE

STUDENT EDITORS

Sumesh Mohan C

Sanya Dixit