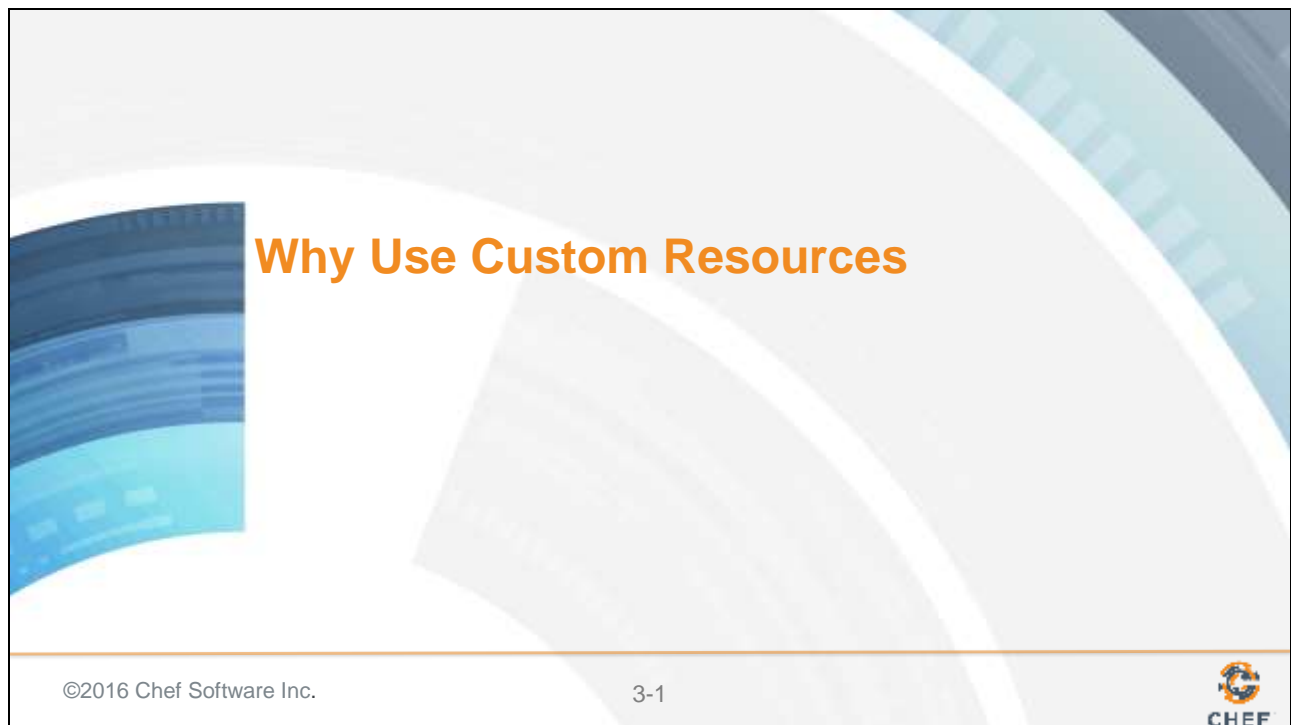


### 3: Why Use Custom Resources



As you can see there are more than a few ways to extend Chef and create a resource or resource-like implementation within your recipes. But before we do that, it is important to understand the value that a custom resource brings to a recipes.

Slide 2

## Objectives


After completing this module, you should be able to:

- Determine when a Custom Resource would be beneficial for clarity and reusability

After completing this module you will be able to describe when a Custom Resource would be beneficial for clarity and reusability.

## Slide 3

# EXERCISE




## Evaluation Before Pursuit

*Just because I can does not mean I should. It is important to implement solutions that are arguably better software design.*

**Objective:**

- ☐ Define the judgment criteria
- ☐ Evaluate a code sample

---

©2016 Chef Software Inc. 3-3 

As an group exercise we are going to look at a series of resources and discuss their quality. Quality can be rather variable unless we select a criteria for which to judge it.

## Slide 4

# CONCEPT



## Software Quality Standards

When defining resources within our recipes we are writing software. Software has a number of quality characteristics that have already been defined. ISO/IEC 9126 is an international standard for evaluation of software quality.

---

©2016 Chef Software Inc. 3-4 

When defining resources within our recipes we are writing software. Software has a number of quality characteristics that have already been defined. ISO/IEC 9126 is an international standard for evaluation of software quality.

## Slide 5

# CONCEPT



## Software Quality Standards

- Functionality
- Reliability
- Usability
- Efficiency
- Maintainability
- Portability


---

©2016 Chef Software Inc. 3-5 

This standard identifies 6 main quality characteristics. Let's talk about each one of these so that we have a shared understanding of what we mean when using them in this exercise.

## Slide 6

# CONCEPT




## Software Quality Standards

<ul style="list-style-type: none"><li>➤ <b>Functionality</b></li><li>➤ Reliability</li><li>➤ Usability</li><li>➤ Efficiency</li><li>➤ Maintainability</li><li>➤ Portability</li></ul>	<p>Does the code accomplish what it is designed to accomplish?</p>
---	--

©2016 Chef Software Inc.

3-6



Functionality is the essential purpose of any product or service. Does the code accomplish what it is designed to accomplish? Functionality may also be concerned with if it does so securely and within compliance guidelines.

## Slide 7

# CONCEPT



## Software Quality Standards

- Functionality
- **Reliability**
- Usability
- Efficiency
- Maintainability
- Portability


Is the solution able to withstand fault and recover from a failure?

©2016 Chef Software Inc. 3-7 

Reliability is a judgment of whether the code accomplishes its functional goal consistently, is able to withstand fault, and recover from a failure.

## Slide 8


# CONCEPT



## Software Quality Standards

- Functionality
- Reliability
- **Usability**
- Efficiency
- Maintainability
- Portability

Is the code easy to understand?  
Is it easy to learn?


©2016 Chef Software Inc. 3-8 

Usability refers to the ease of use for the given code. Is the code easy to understand? Is it easy to learn? Does it adhere to common team standards?



## Slide 9

# CONCEPT




## Software Quality Standards

- Functionality
- Reliability
- Usability
- **Efficiency**
- Maintainability
- Portability

Does the code consume too many physical resources when it executes (e.g. CPU, memory)?

---

©2016 Chef Software Inc. 3-9 

Efficiency is concerned with the system resources required to achieve the functionality. We may consider the time, CPU, memory, network requirements, or physical space it takes to accomplish the intended operation.

## Slide 10

# CONCEPT



## Software Quality Standards

- Functionality
- Reliability
- Usability
- Efficiency
- **Maintainability**
- Portability


Are you able to easily adapt the solution? Is it testable?

©2016 Chef Software Inc. 3-10 

Maintainability measures the code to see if it is supportable. If there is a failure are you able to quickly identify the issue? Are you able to easily adapt the solution? Is it testable?

## Slide 11

# CONCEPT




## Software Quality Standards

- Functionality
- Reliability
- Usability
- Efficiency
- Maintainability
- **Portability**

Can the software adapt to changes in its environment? Or changes to its requirements?


---

©2016 Chef Software Inc. 3-11 

Portability refers to how well the software can adapt to changes in its environment or with its requirements. This may also include evaluating code for its adaptability and maybe even be easily replaced.

Slide 12

# EXERCISE




## Examine the Code Sample

*With the criteria defined we can now examine code samples...*

**Objective:**

- ✓ Define the judgment criteria
- Evaluate a code sample

---

©2016 Chef Software Inc. 3-12 

Let's examine this first example and apply the criteria that we have defined.

## Slide 13

## Resource Implementation v Custom Resource

```
directory '/srv/apache/admins/html' do
  recursive true
  mode '0755'
end

template '/etc/httpd/conf.d/admins.conf' do
  source 'conf.erb'
  mode '0644'
end

variables({document_root: '/srv/apache/admins/html',
port: 8080})
  notifies :restart, 'service[httpd]'
end

file '/srv/apache/admins/html/index.html' do
  content '<h1>Welcome admins!</h1>'
end
```

```
apache_vhost 'admins' do
  site_port 8080
end
```

**Functionality** | Reliability | Usability | Efficiency | Maintainability | Portability

**Does the code accomplish what it is designed to accomplish?**

## Slide 14

## Resource Implementation v Custom Resource

```
directory '/srv/apache/admins/html' do
  recursive true
  mode '0755'
end

template '/etc/httpd/conf.d/admins.conf' do
  source 'conf.erb'
  mode '0644'
end

variables(document_root: '/srv/apache/admins/html',
port: 8080)
  notifies :restart, 'service[httpd]'
end

file '/srv/apache/admins/html/index.html' do
  content '<h1>Welcome admins!</h1>'
end
```

```
apache_vhost 'admins' do
  site_port 8080
end
```

Functionality | **Reliability** | Usability | Efficiency | Maintainability | Portability

**Is the solution able to withstand fault and recover from a failure?**

## Resource Implementation v Custom Resource

```
directory '/srv/apache/admins/html' do
  recursive true
  mode '0755'
end

template '/etc/httpd/conf.d/admins.conf' do
  source 'conf.erb'
  mode '0644'
end

variables(document_root: '/srv/apache/admins/html',
port: 8080)
  notifies :restart, 'service[httpd]'
end

file '/srv/apache/admins/html/index.html' do
  content '<h1>Welcome admins!</h1>'
end
```

```
apache_vhost 'admins' do
  site_port 8080
end
```

Functionality | Reliability | **Usability** | Efficiency | Maintainability | Portability

Is the code easy to understand? Is it easy to learn?

## Resource Implementation v Custom Resource

```
directory '/srv/apache/admins/html' do
  recursive true
  mode '0755'
end

template '/etc/httpd/conf.d/admins.conf' do
  source 'conf.erb'
  mode '0644'
end

variables({document_root: '/srv/apache/admins/html',
port: 8080})
  notifies :restart, 'service[httpd]'
end

file '/srv/apache/admins/html/index.html' do
  content '<h1>Welcome admins!</h1>'
end
```

```
apache_vhost 'admins' do
  site_port 8080
end
```

Functionality | Reliability | Usability | **Efficiency** | Maintainability | Portability

**Does the code consume too many physical resources when it executes (e.g. CPU, memory)?**



## Slide 17

## Resource Implementation v Custom Resource

```
directory '/srv/apache/admins/html' do
  recursive true
  mode '0755'
end

template '/etc/httpd/conf.d/admins.conf' do
  source 'conf.erb'
  mode '0644'
end

variables({document_root: '/srv/apache/admins/html',
port: 8080})
  notifies :restart, 'service[httpd]'
end

file '/srv/apache/admins/html/index.html' do
  content '<h1>Welcome admins!</h1>'
end
```

```
apache_vhost 'admins' do
  site_port 8080
end
```

Functionality | Reliability | Usability | Efficiency | **Maintainability** | Portability

**Are you able to easily adapt the solution? Is it testable?**

## Resource Implementation v Custom Resource

```
directory '/srv/apache/admins/html' do
  recursive true
  mode '0755'
end

template '/etc/httpd/conf.d/admins.conf' do
  source 'conf.erb'
  mode '0644'
end

variables(document_root: '/srv/apache/admins/html',
port: 8080)
  notifies :restart, 'service[httpd]'
end


file '/srv/apache/admins/html/index.html' do
  content '<h1>Welcome admins!</h1>'
end
```

```
apache_vhost 'admins' do
  site_port 8080
end
```

Functionality | Reliability | Usability | Efficiency | Maintainability | **Portability**

**Can the software adapt to changes in its environment? Or changes to its requirements?**

# EXERCISE




## Evaluation Before Pursuit

*There are many ways to critically evaluate code ... if these do not suit your or your team find the ones that do; talk about them and share them.*

**Objective:**

- ✓ Define the judgment criteria
- ✓ Evaluate a code sample

---

©2016 Chef Software Inc. 3-19 

We've evaluated one code sample, let's look at a second one.

Slide 20

# DISCUSSION



## Discussion

What value does reviewing code for functionality, reliability, usability, efficiency, maintainability, portability bring?

Slide 21

# DISCUSSION



## Q&A

What questions can we answer for you?

---

©2016 Chef Software Inc. 3-21 

