CHEF™

Intro to Chef Delivery

Nathen Harvey - @nathenharvey

Agenda

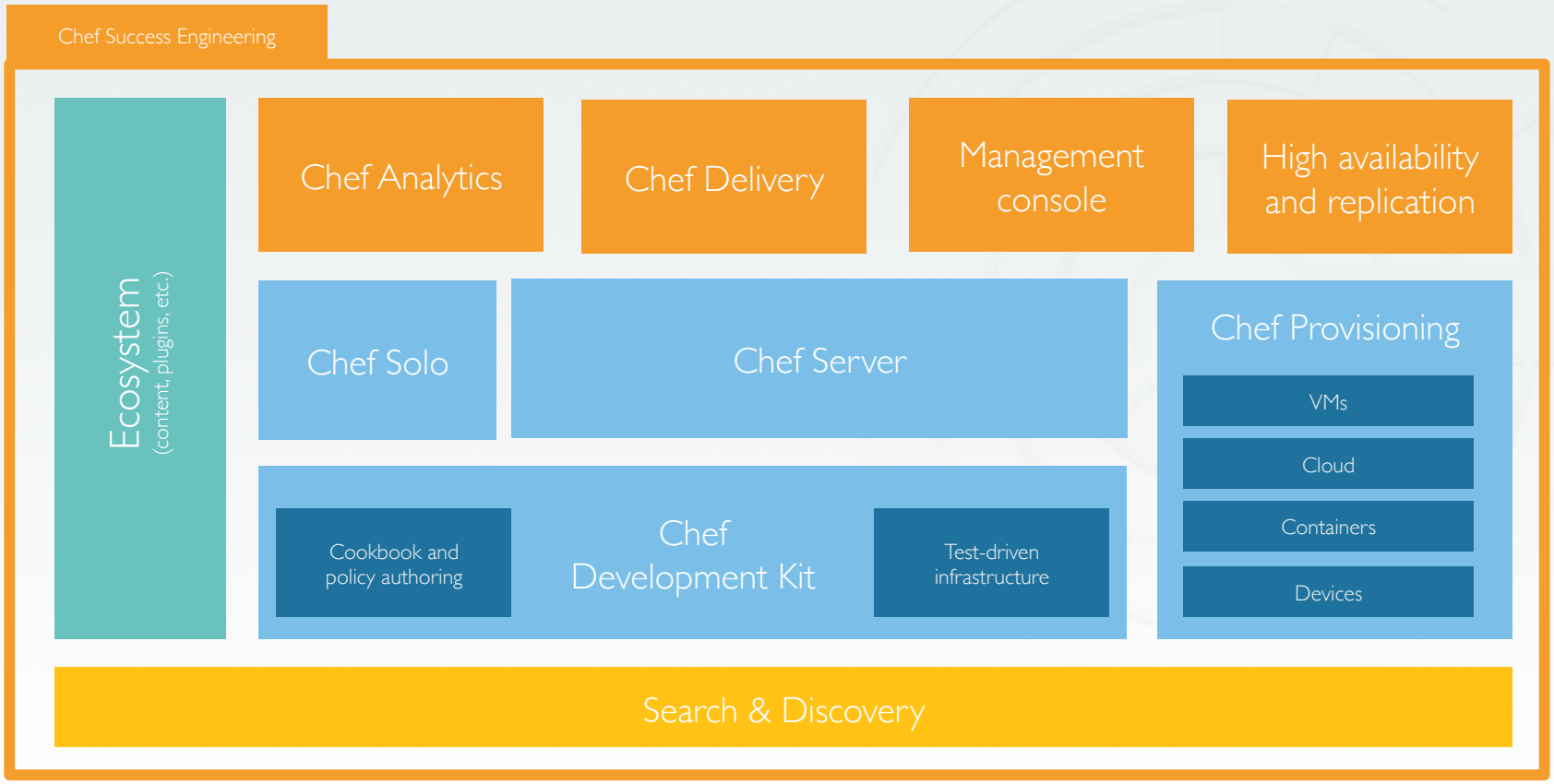
- Overview
- Chef Software Platform
- Building Blocks
- Automation Patterns
- Chef Delivery

CHEF

Overview

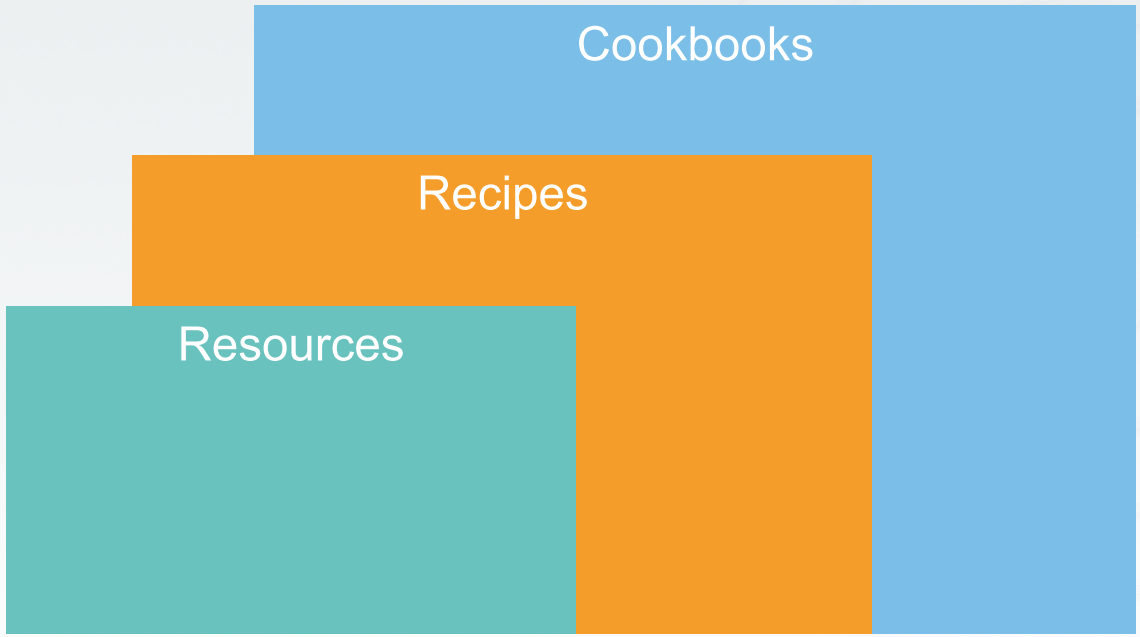
- Chef is an automation framework that enables Infrastructure as Code
- Chef leverages reusable definitions to automate desired state
- Chef is API driven
- Chef supports Linux variants, Unix variants, AIX and Windows, all as first class citizens.

The Chef Software Platform



CHEF

Building Blocks



CHEF

Building Blocks: What is a Resource?

- A Resource is a system state you define
  - Example: Package installed, state of a service, configuration file existing
- You declare what the state of the resource is
  - Chef automatically determine HOW that state is achieved

```
windows_feature "IIS-WebServerRole" do
  action :install
end
```

```
package "httpd" do
  action :install
end
```

CHEF

Building Blocks: What is a Recipe?

- A recipe is a collection of Resources
- Resources are executed in the order they are listed

```
windows_feature "IIS-WebServerRole" do
  action :install
end

template 'c:\inetpub\wwwroot\Default.htm' do
  source "Default.htm.erb"
  rights :read, "Everyone"
end

service "w3svc" do
  action [ :enable, :start ]
end
```

```
package "httpd" do
  action :install
end

template "/var/www/index.html" do
  source "index.html.erb"
  mode "0644"
end

service "httpd" do
  action [ :enable, :start ]
end
```

CHEF

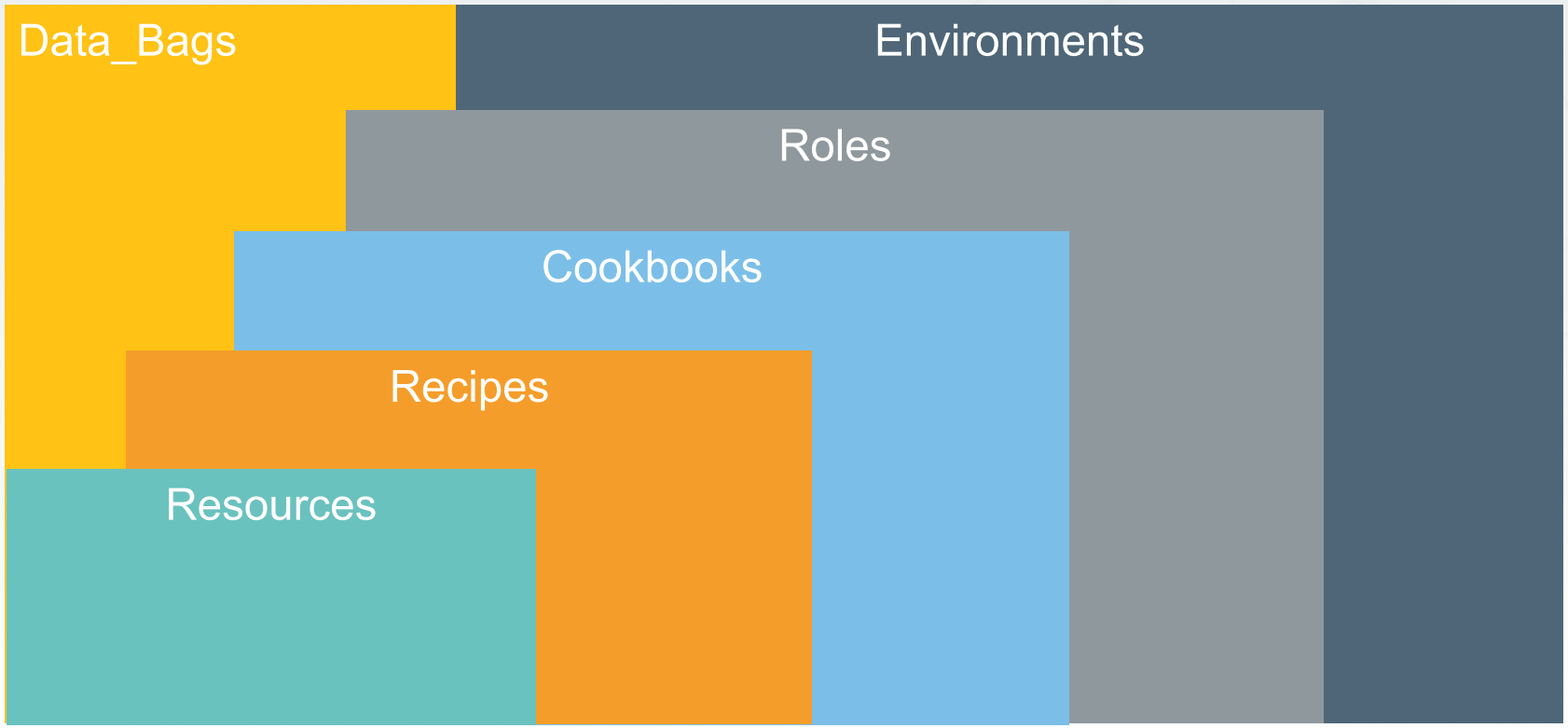
Building Blocks: What is a Cookbook?

- A cookbook is a set of recipes
- A cookbook is a defined set of items and different outcomes that you expect to address

  A cookbook could have a recipe to install apache2/httpd but also another set of recipes to activate modules required.

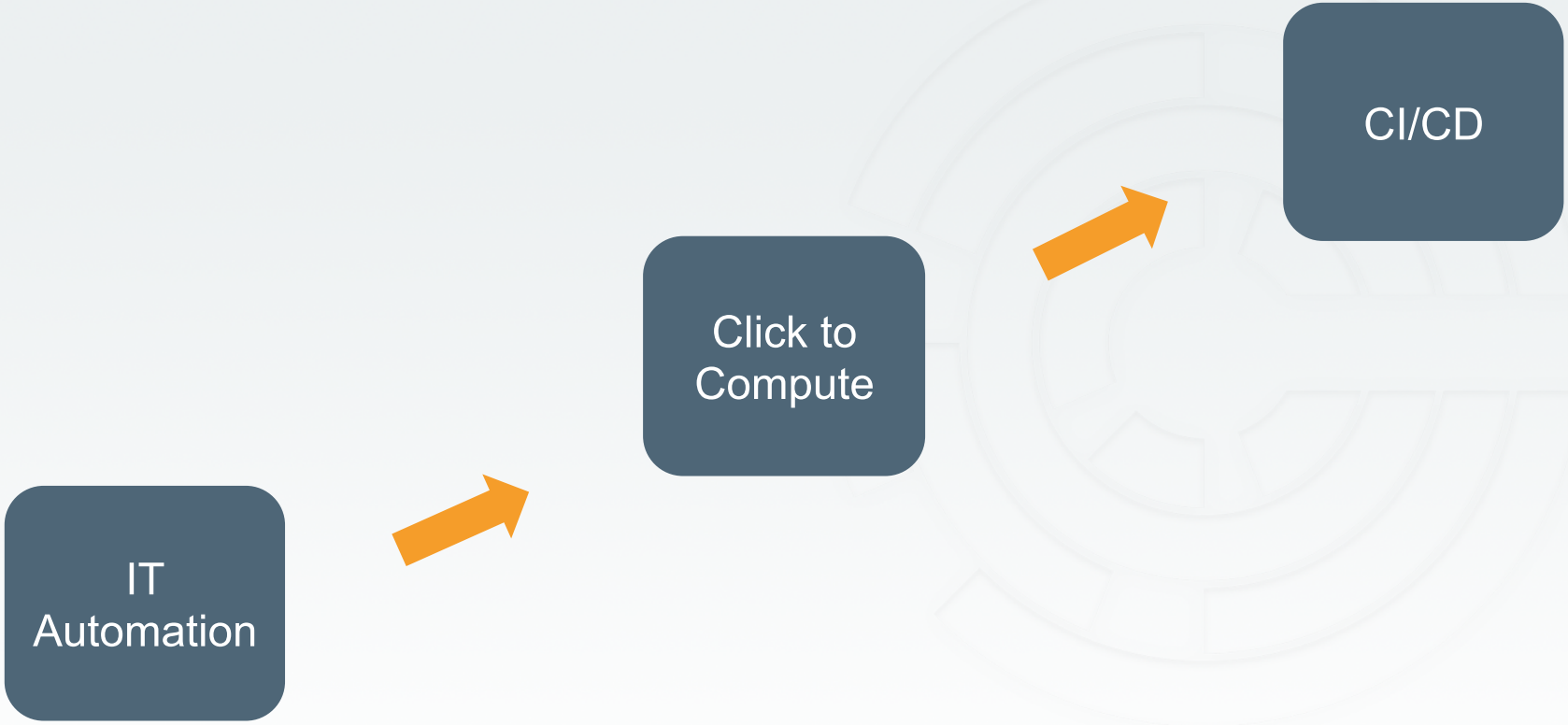
```
./attributes
./attributes/default.rb
./CHANGELOG.md
./metadata.rb
./README.md
./recipes
./recipes/application.rb
./recipes/balancer.rb
./recipes/database.rb
./recipes/default.rb
./recipes/webserver.rb
./templates
./templates/default
./templates/default/mysite.conf.erb
```

Building Blocks



CHEF

Automation Patterns



CHEF

CHEF DELIVERY

WE'VE IDENTIFIED A PROVEN PIPELINE

VALIDATED IN OUR ENGAGEMENTS WITH
ENTERPRISE AND BIG WEB CUSTOMERS.

STABLE PIPELINE

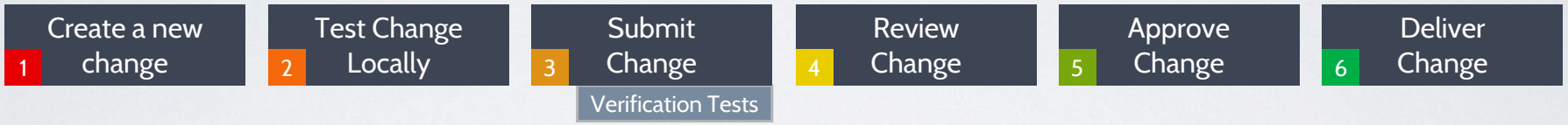
Steps

1. Create a new change
2. Test Change Locally
3. Submit Change
4. Review Change
5. Approve Change
6. Deliver Change

STABLE PIPELINE

● manual ● automated

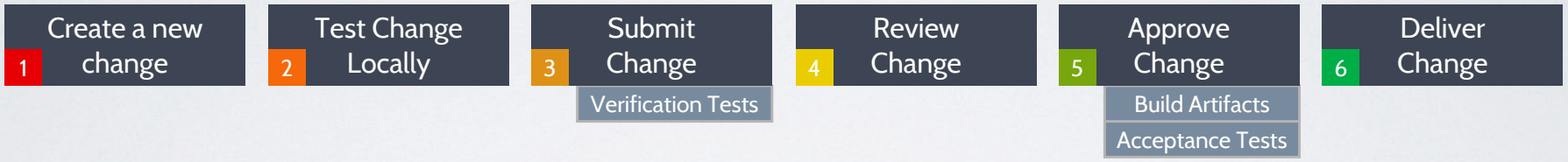
Steps



STABLE PIPELINE

● manual ● automated

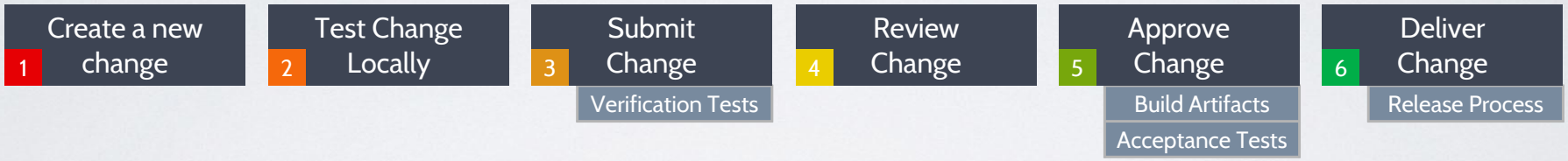
Steps



STABLE PIPELINE

● manual ● automated

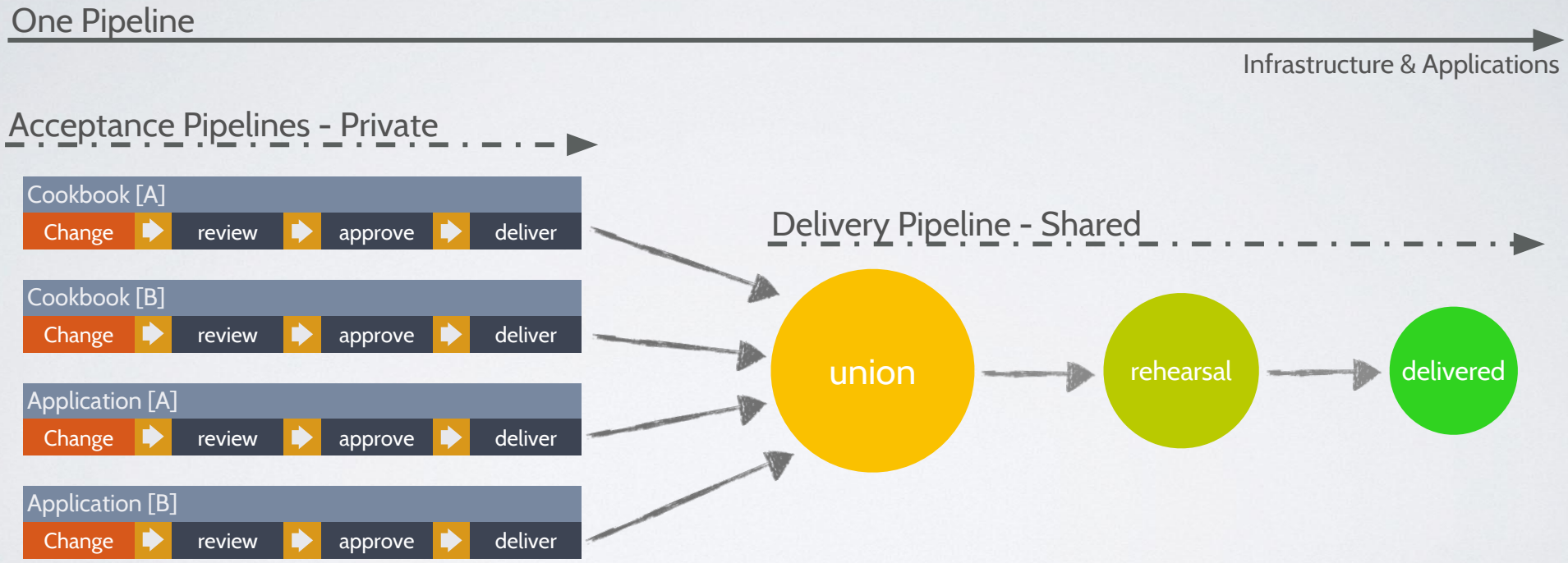
Steps



A SHARED PIPELINE

ACROSS PROJECTS AND TEAMS

COMMON PIPELINE



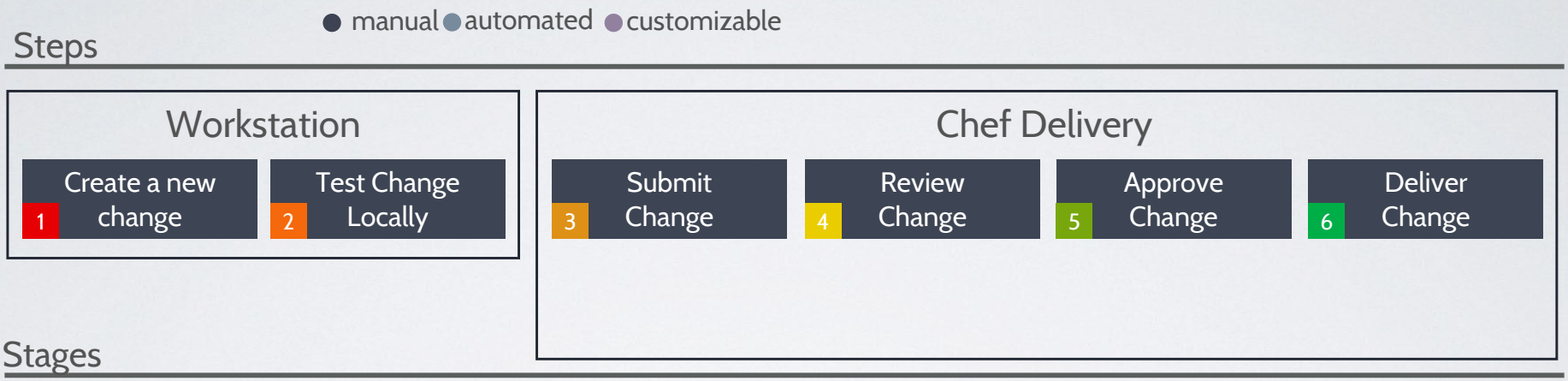
U-071982-C

A UNIFIED PIPELINE SHAPE

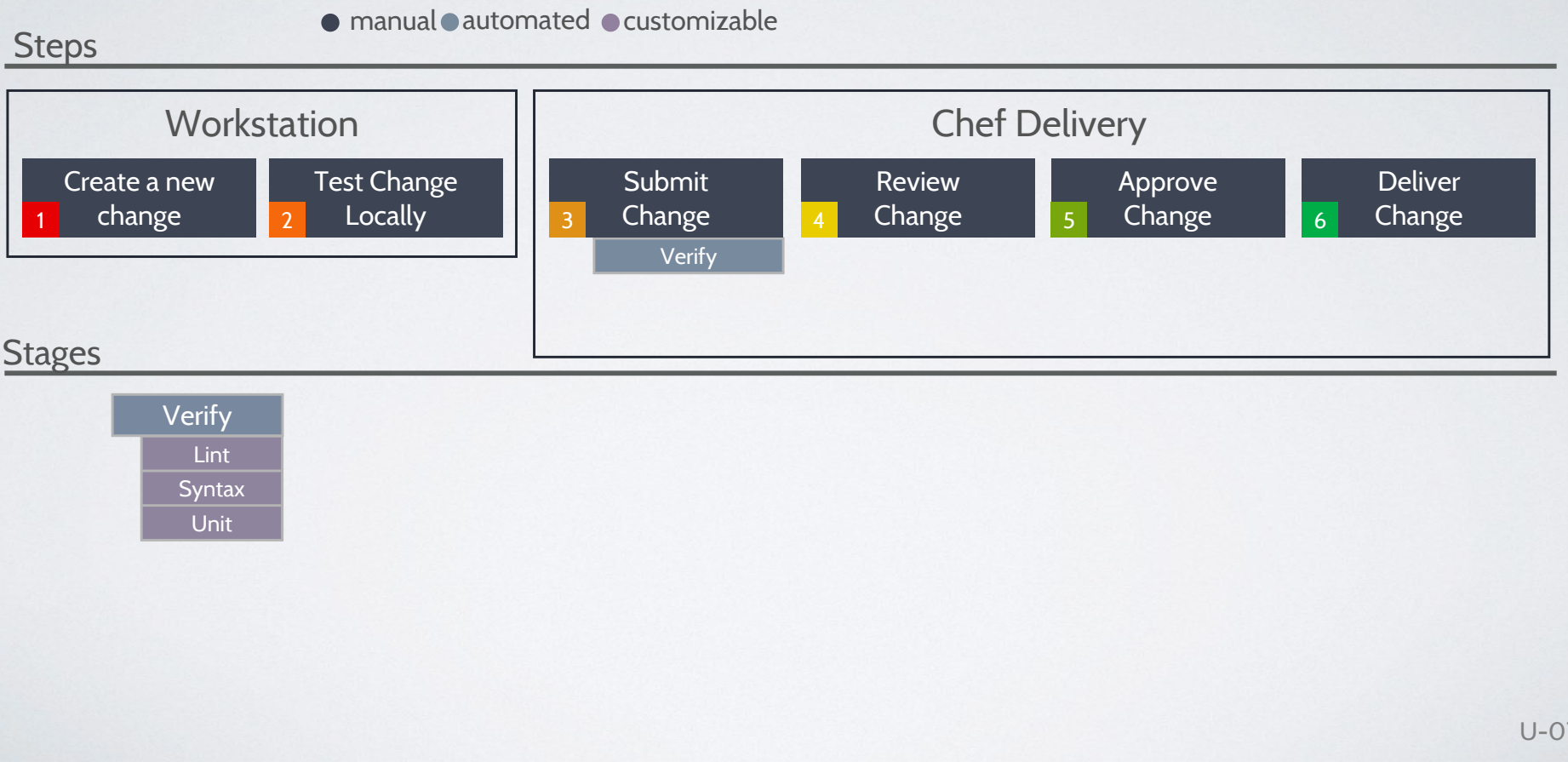
THE STAGES ARE FIXED

EACH STAGE HAS A FIXED SET OF PHASES

COMMON PIPELINE

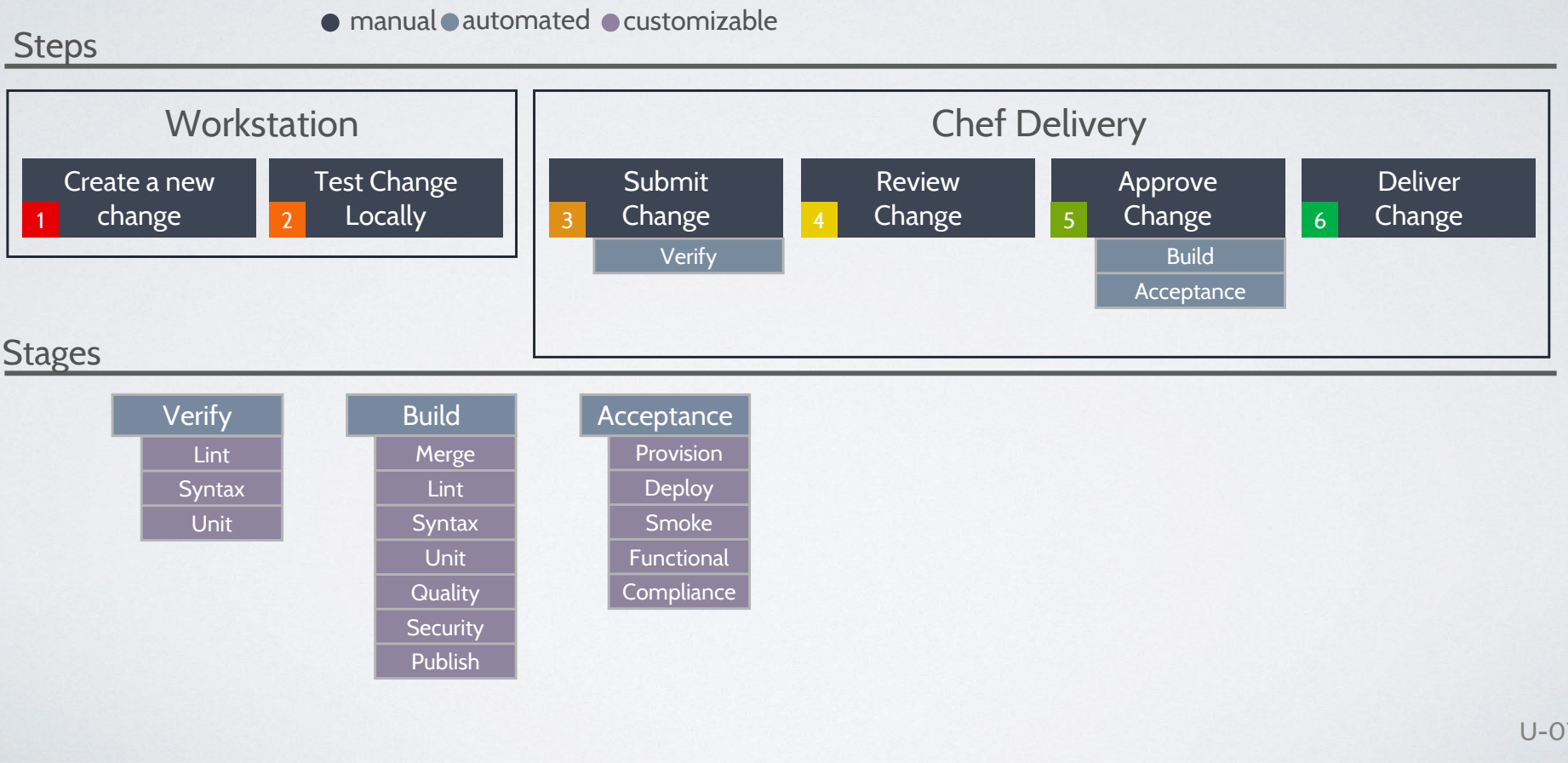


COMMON PIPELINE

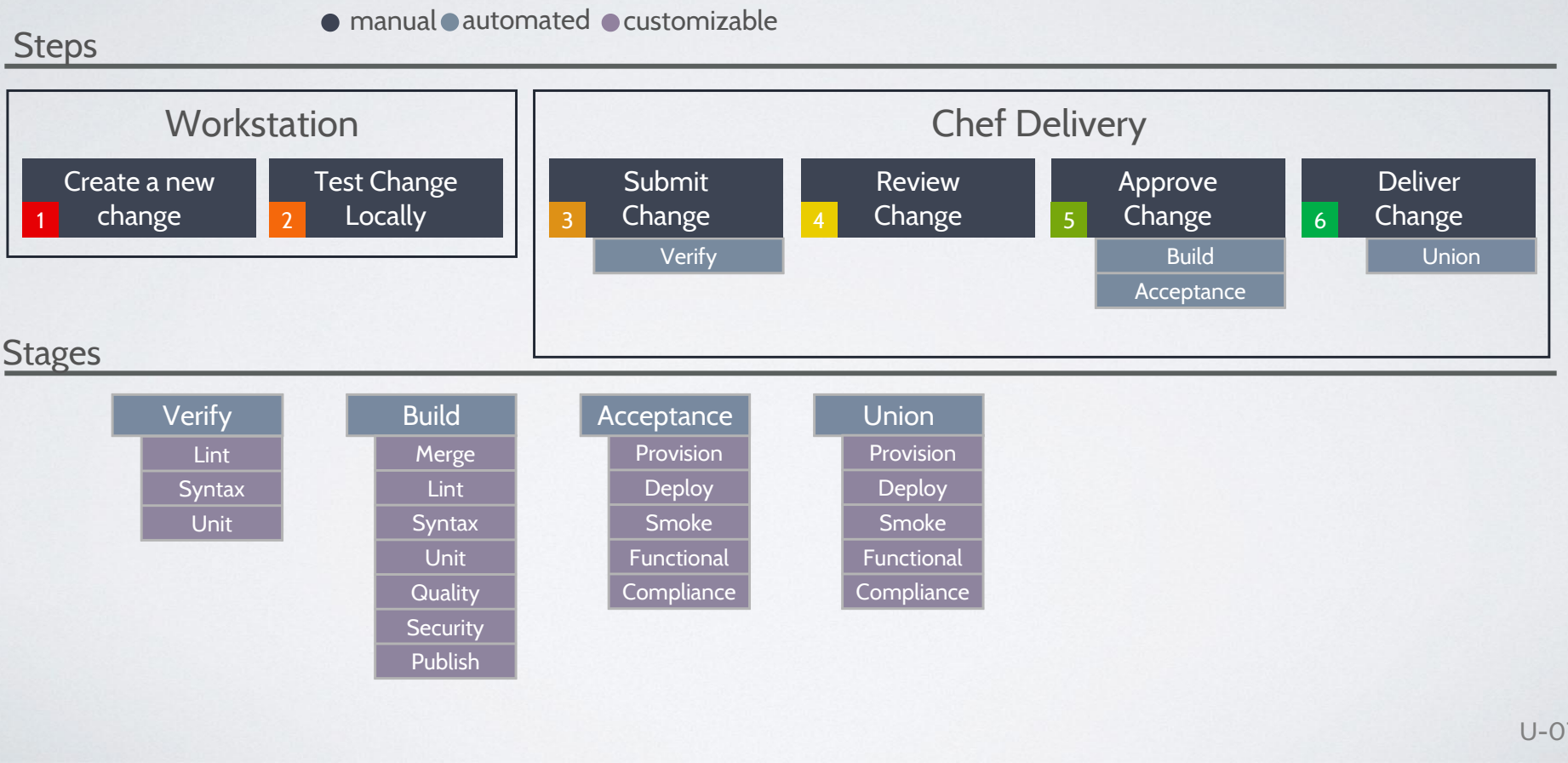


COMMON PIPELINE

COMMON PIPELINE



COMMON PIPELINE



COMMON PIPELINE

● manual ● automated ● customizable

Steps

Workstation

1. Create a new change
2. Test Change Locally

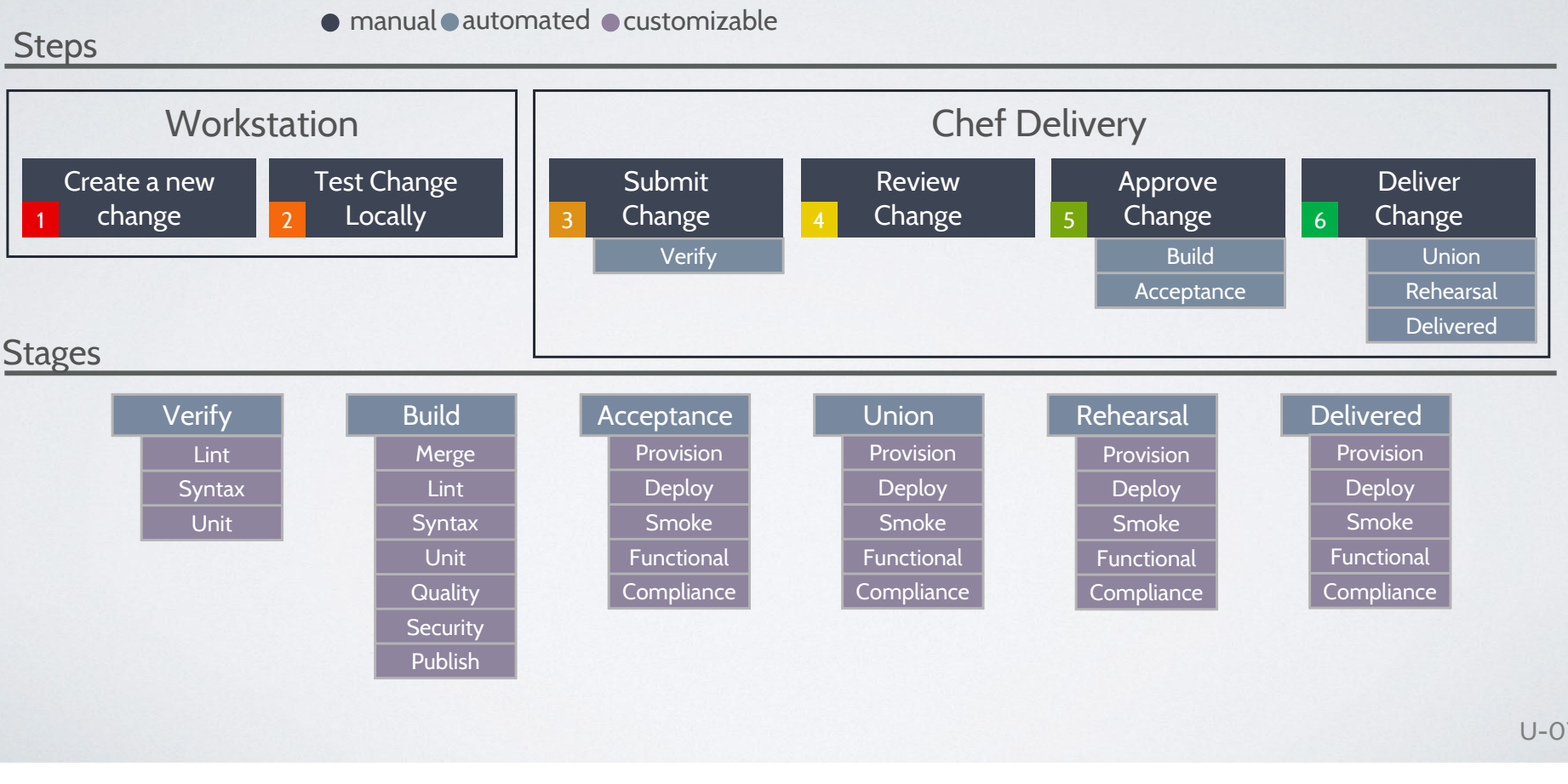
Chef Delivery

3. Submit Change
   - Verify
4. Review Change
5. Approve Change
   - Build
   - Acceptance
6. Deliver Change
   - Union
   - Rehearsal

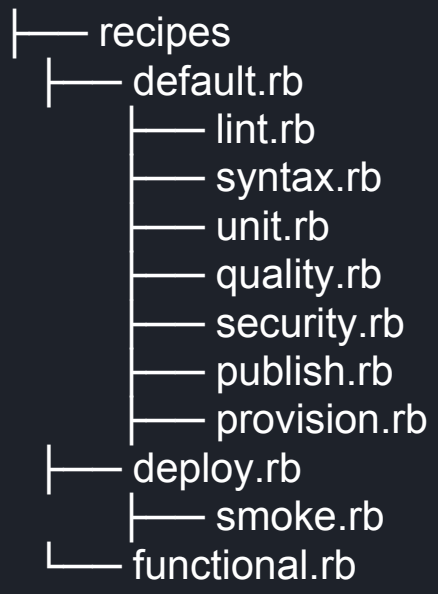
Stages

- **Verify**
  - Lint
  - Syntax
  - Unit
- **Build**
  - Merge
  - Lint
  - Syntax
  - Unit
  - Quality
  - Security
  - Publish
- **Acceptance**
  - Provision
  - Deploy
  - Smoke
  - Functional
  - Compliance
- **Union**
  - Provision
  - Deploy
  - Smoke
  - Functional
  - Compliance
- **Rehearsal**
  - Provision
  - Deploy
  - Smoke
  - Functional
  - Compliance

COMMON PIPELINE



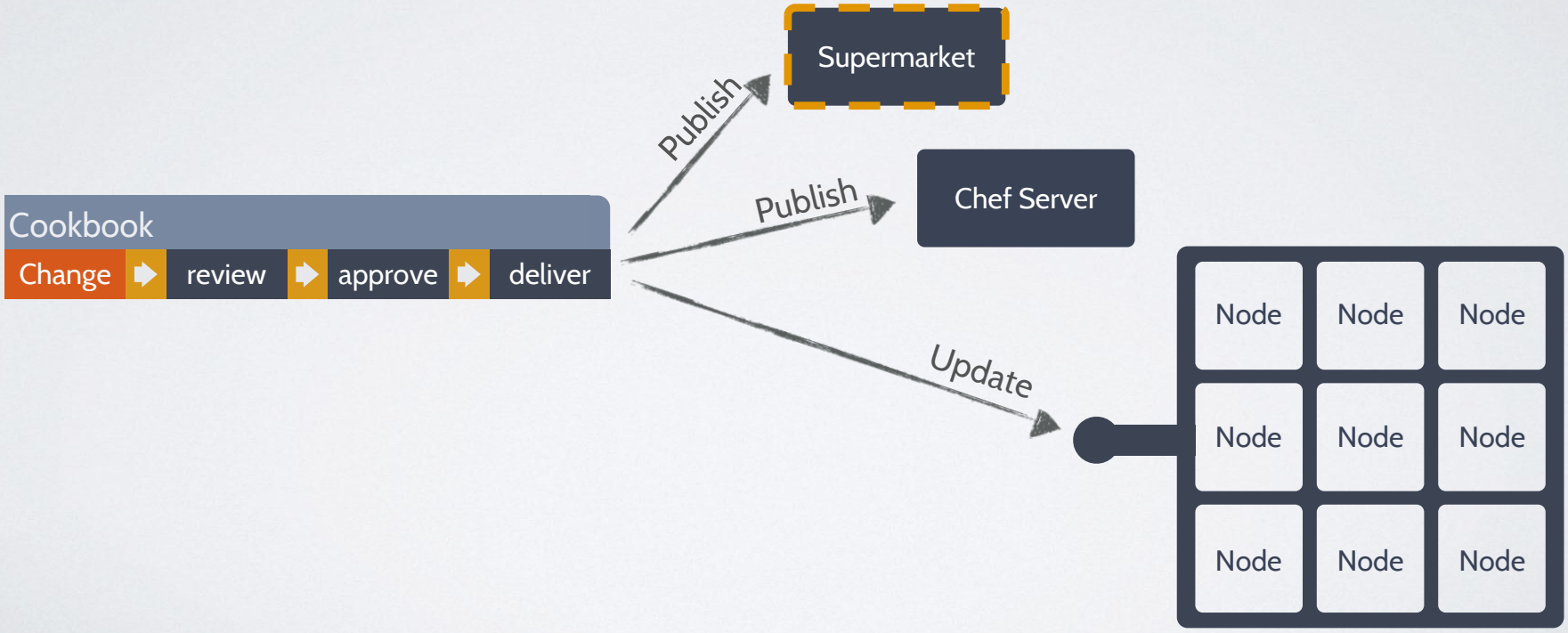
BUILD COOKBOOK



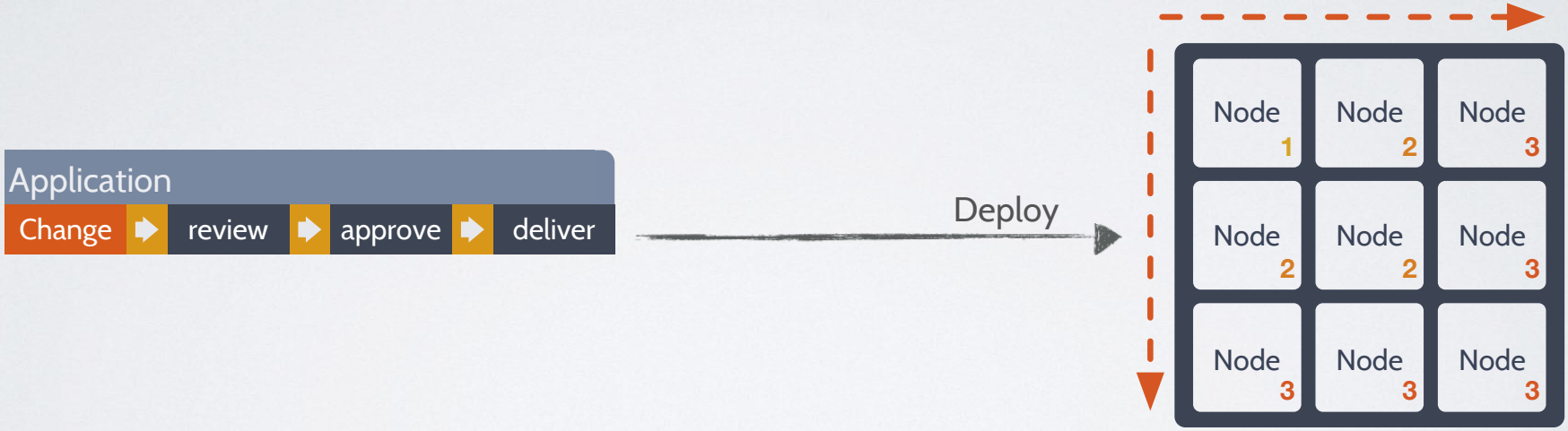
PHASE EXECUTION

```
log "Running unit"

repo = node['delivery_builder']['repo']

execute “run my junit tests" do
  command "mvn test"
  cwd repo
end
```

COOKBOOK WORKFLOW



APPLICATION WORKFLOW



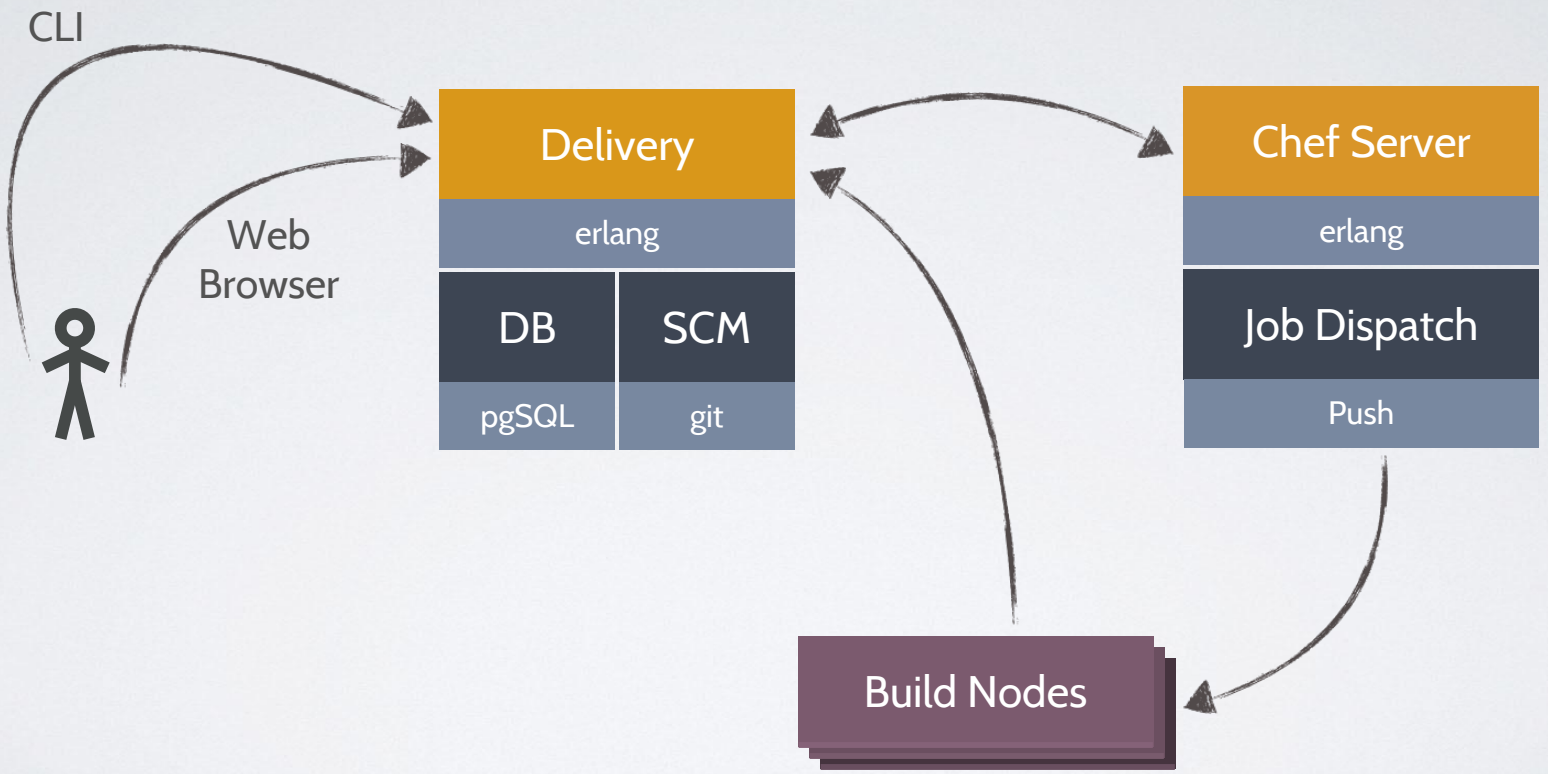
EASILY CREATE DELIVERY PIPELINES

PROJECT GIT REPO
BUILD COOKBOOK
PROJECT CONFIG FILE

CONFIG.JSON

```
{
  "version": "1",
  "build_cookbook": "./.delivery/build_cookbook",
  "build_nodes": {
    "default": [“name:*buildnode-*”],
          "unit": [“name:*-unittest*”],
          "lint": [“name:*-linttest*”],
    "build": [“name:*-builder*”]
  }
}
```

HOW IT WORKS


CLI
Web
Browser
Delivery
erlang
DB
SCM
pgSQL
git
Chef Server
erlang
Job Dispatch
Push
Build Nodes

U-071982-C