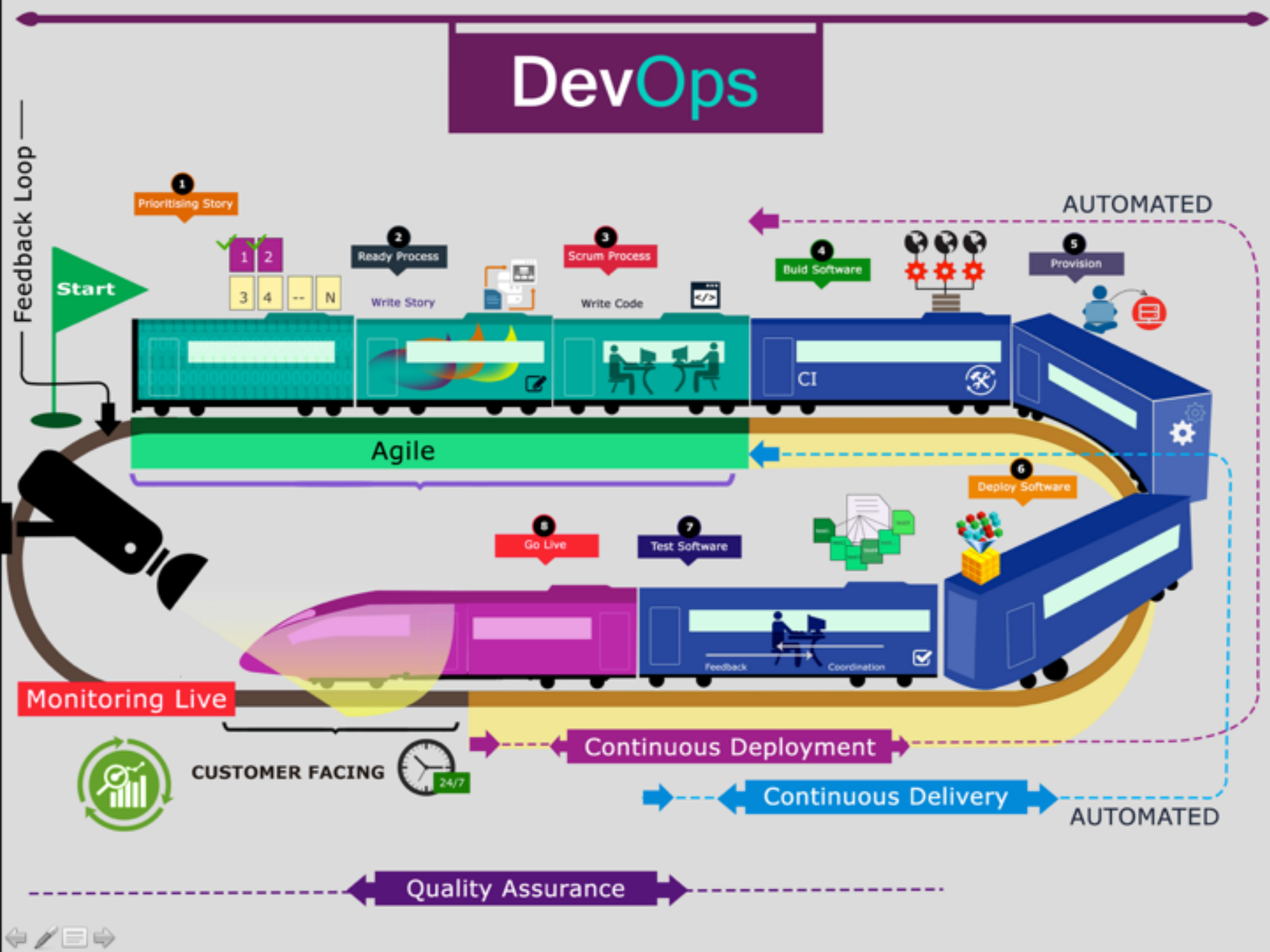
DevOps Part 3: Platform Automation

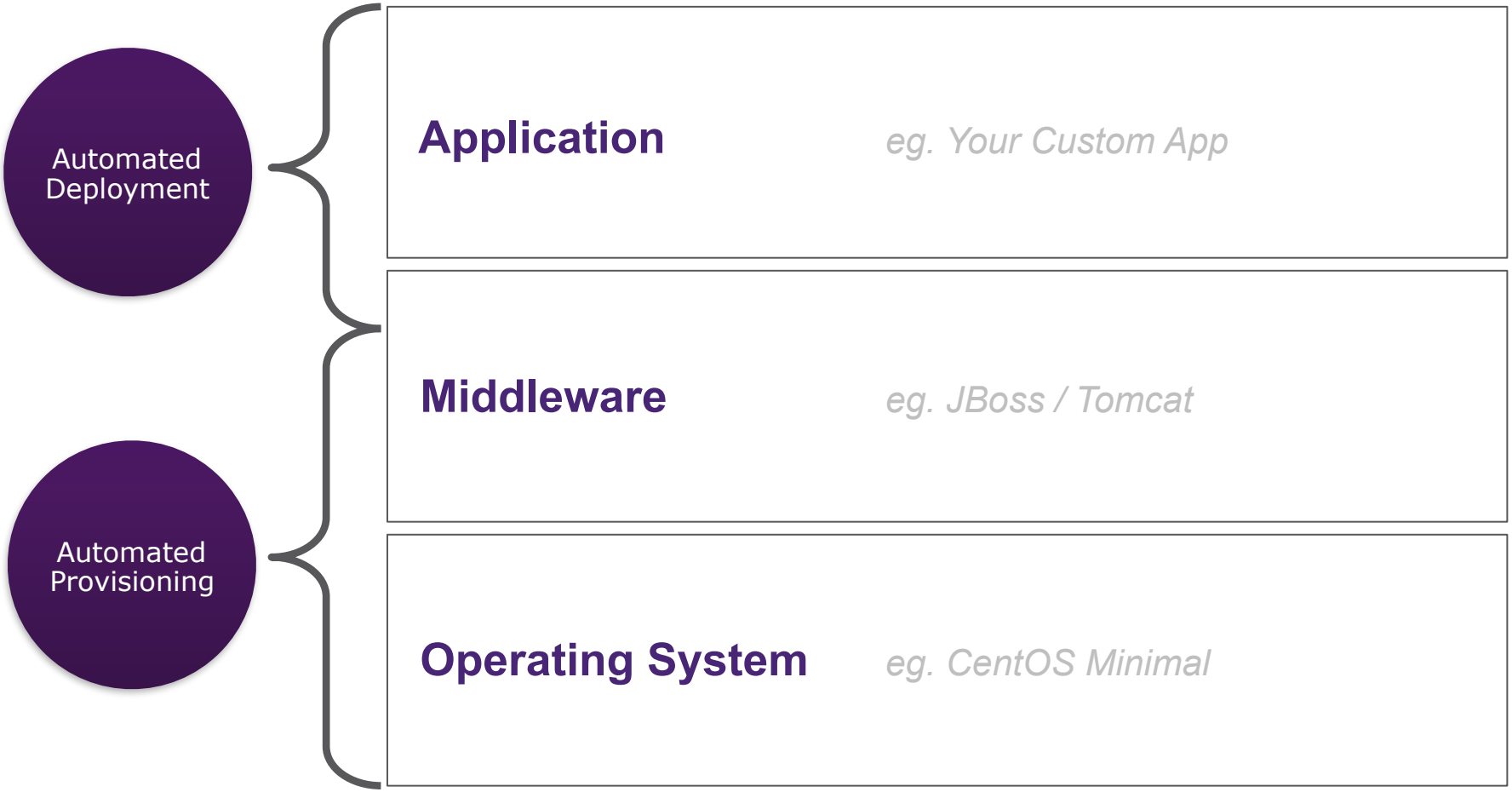
MANISH KUMAR THAKUR
MTHAKUR@XEBIA.COM

Xebia INSPIRE CHANGE **ACT ON RESULTS**

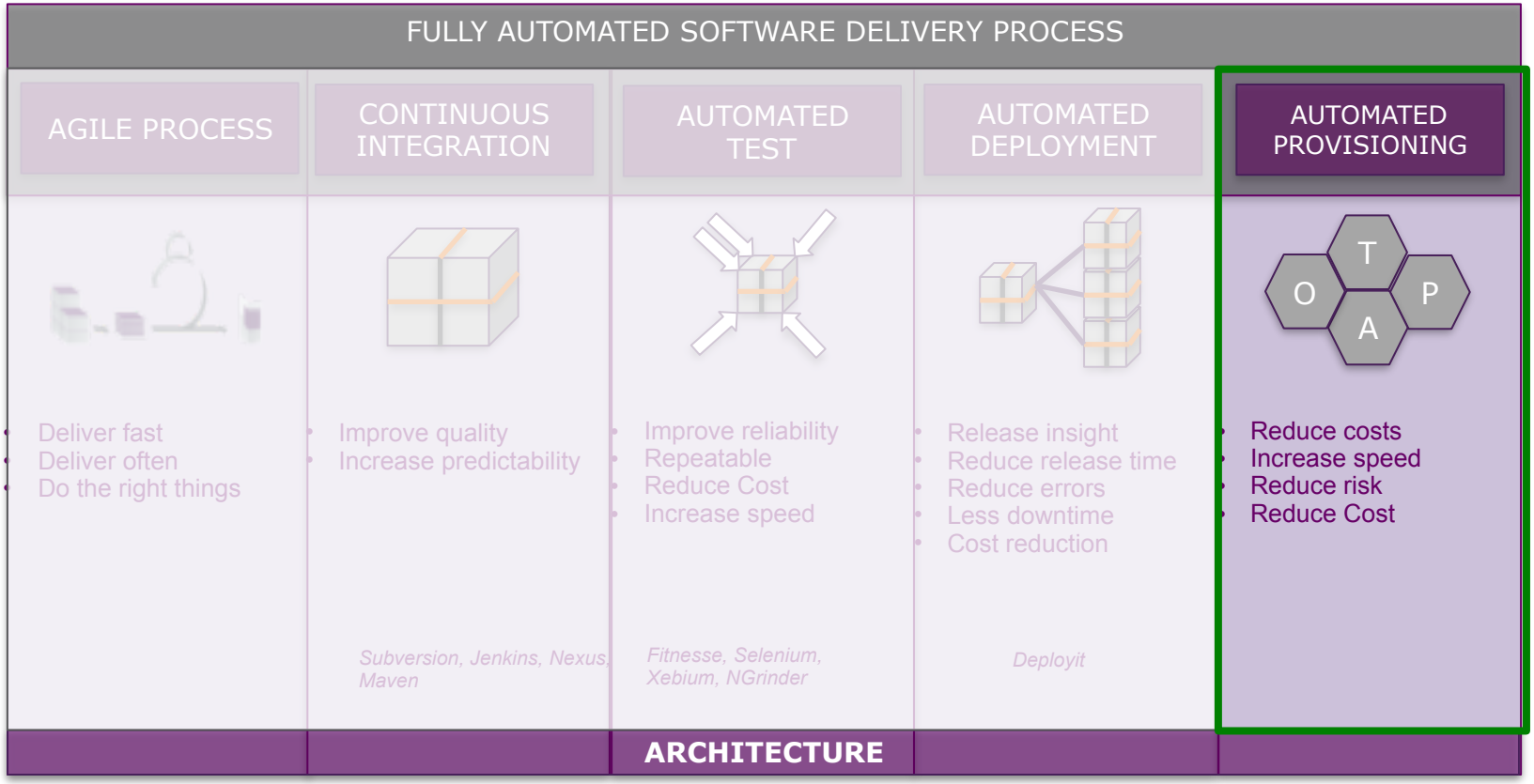
DevOps


Feedback Loop
1 Prioritising Story
2 Ready Process
3 Scrum Process
4 Buid Software
5 Provision
AUTOMATED
Start
1 2
3 4 -- N
Write Story
Write Code
CI
Agile
6 Deploy Software
8 Go Live
7 Test Software
Feedback
Coordination
Monitoring Live
CUSTOMER FACING
24/7
Continuous Deployment
Continuous Delivery
AUTOMATED
Quality Assurance

Automated Deployment and Provisioning



AUTOMATED PROVISIONING



“INFRASTRUCTURE AS CODE”

CM Tools

Ansible	Chef	Puppet
Open source	Open source, standard and premium	Open source and commercial enterprise
YAML	Ruby	Ruby but customised DSL
SSH-based	Master-client	Master-agent
When to use: • getting up and running quickly • No need for agents on remote nodes or managed servers	When to use: • Development-focused teams and environments • Mature solution for a heterogeneous environment	When to use: • Choice if stability and maturity • Good for large enterprises with a heterogeneous environment

PUPPET LANGUAGE

A Declarative Domain Specific Language (DSL)

It defines STATES (Not procedures)

Puppet code is written in manifests (files with .pp extension)

In the code we declare resources that affect elements of the system (files, packages, services ...)

Resource Types

Resource Types are single units of configuration composed by:

A type (package, service, file, user, mount, exec ...)

A title (how is it referred)

Zero or more arguments

```
type { 'title':
  argument  => value,
  other_arg => value,
}
```

Example for a file resource type:

```
file { 'DevOps':
  path    => '/etc/DevOps',
  content => 'Welcome to DevOps',
}
```

INSTALL TOMCAT MODULE

```
puppet module install puppetlabs-tomcat
```

PUPPET APPLY

```
puppet apply tomcat.pp

tomcat::install { '/opt/tomcat':
  source_url => 'http://www-us.apache.org/dist/
tomcat/tomcat-7/v7.0.70/bin/apache-
tomcat-7.0.70.tar.gz ',
}
tomcat::instance { 'default':
  catalina_home => '/opt/tomcat',
}
```

INSTALL MYSQL MODULE

```
puppet module install puppetlabs-mysql
```

PUPPET APPLY

```
puppet apply mysql.pp

class { '::mysql::server':
  root_password           => '<<password>>',
  remove_default_accounts => true,
  override_options        => $override_options
}
mysql::db { 'DevOps':
  user     => 'mthakur',
  password => '<<password>>',
  host     => 'localhost',
  grant    => ['SELECT', 'UPDATE'],
}
```

Check in your Database Server

```
mysql

show databases
```

CENTRALIZED LOGGING:

PAPERTRAIL

PAPERTRAIL

- Create an account at:
  - https://papertrailapp.com/
- Copy your log destination
  - E.g. logs3.papertrailapp.com:<YOUR-PORT-NR>
- Type:

```
export PAPERTRAIL=logs3.papertrailapp.com:<port>

./startLogspout.sh
```

Perform a build in Jenkins

- Check the result in Papertrail
- Try filtering

```
program:default_xlr_1 err
```

CENTRALIZED MONITORING:

DATADOG

DATADOG

- Create an account at:
  - http://www.datadoghq.com/

CHECK THE DASHBOARD AS IS

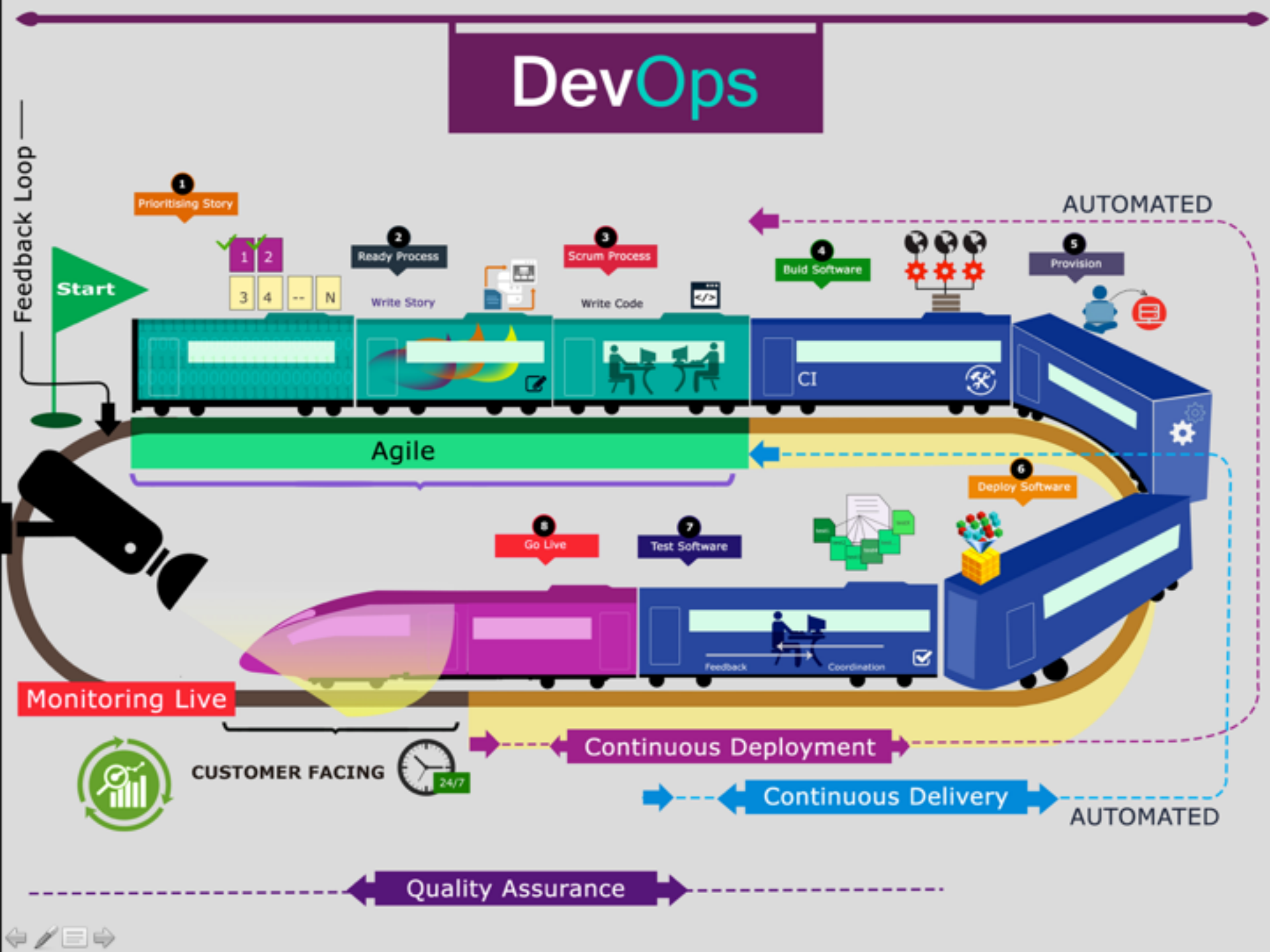
DataDog

- Open Integrations -> APIs
  - Create an API key
  - Copy it
- Go back to your terminal and type

```
export DATADOG=<your-API-key>
./startDatadog.sh
```

CHECK THE DASHBOARD WITH INFO

DevOps


Feedback Loop
1 Prioritising Story
2 Ready Process
3 Scrum Process
4 Buid Software
5 Provision
AUTOMATED
Start
1 2
3 4 -- N
Write Story
Write Code
CI
Agile
6 Deploy Software
8 Go Live
7 Test Software
Feedback
Coordination
Monitoring Live
CUSTOMER FACING
24/7
Continuous Deployment
Continuous Delivery
AUTOMATED
Quality Assurance