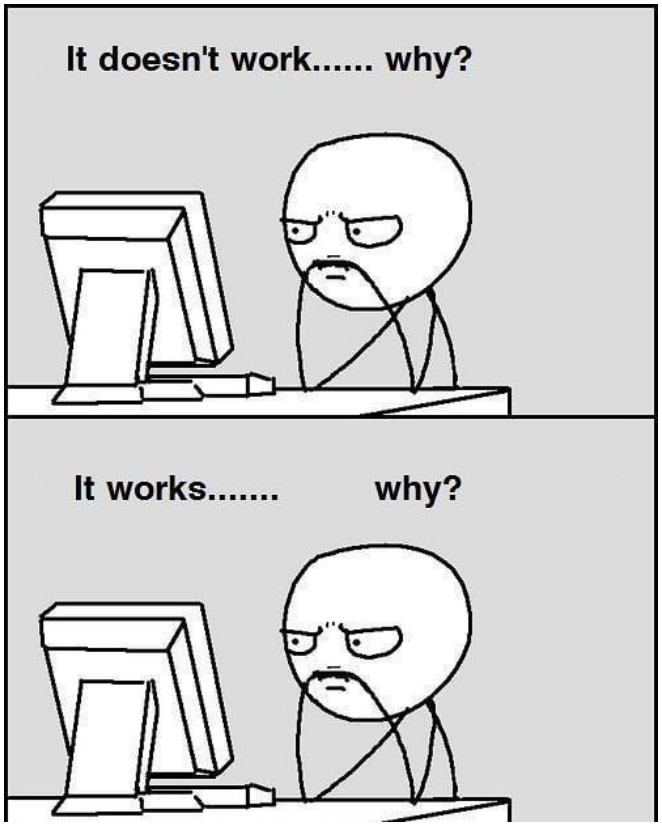
Inner Loop Development Experience and Options

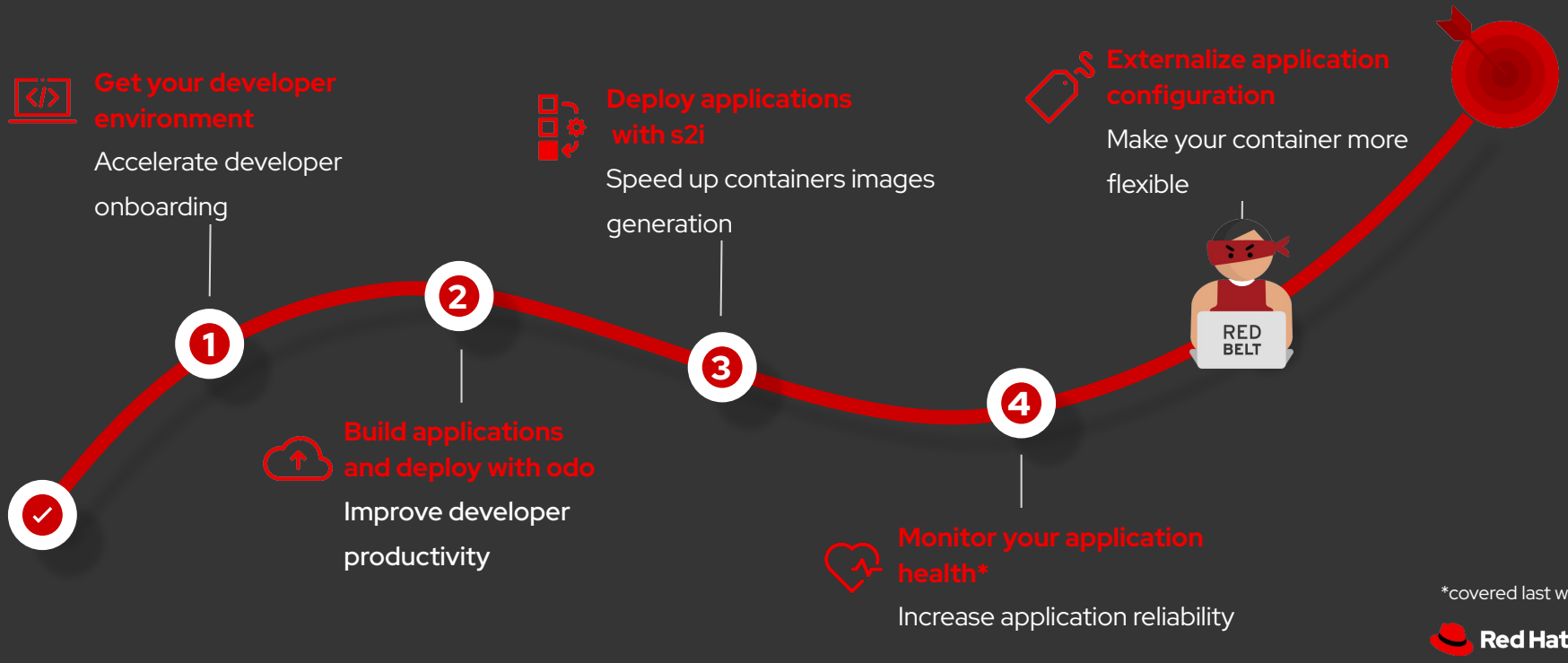
Florian Moss, Solution Architect

December 2021

Red Hat


It doesn't work..... why?
It works..... why?

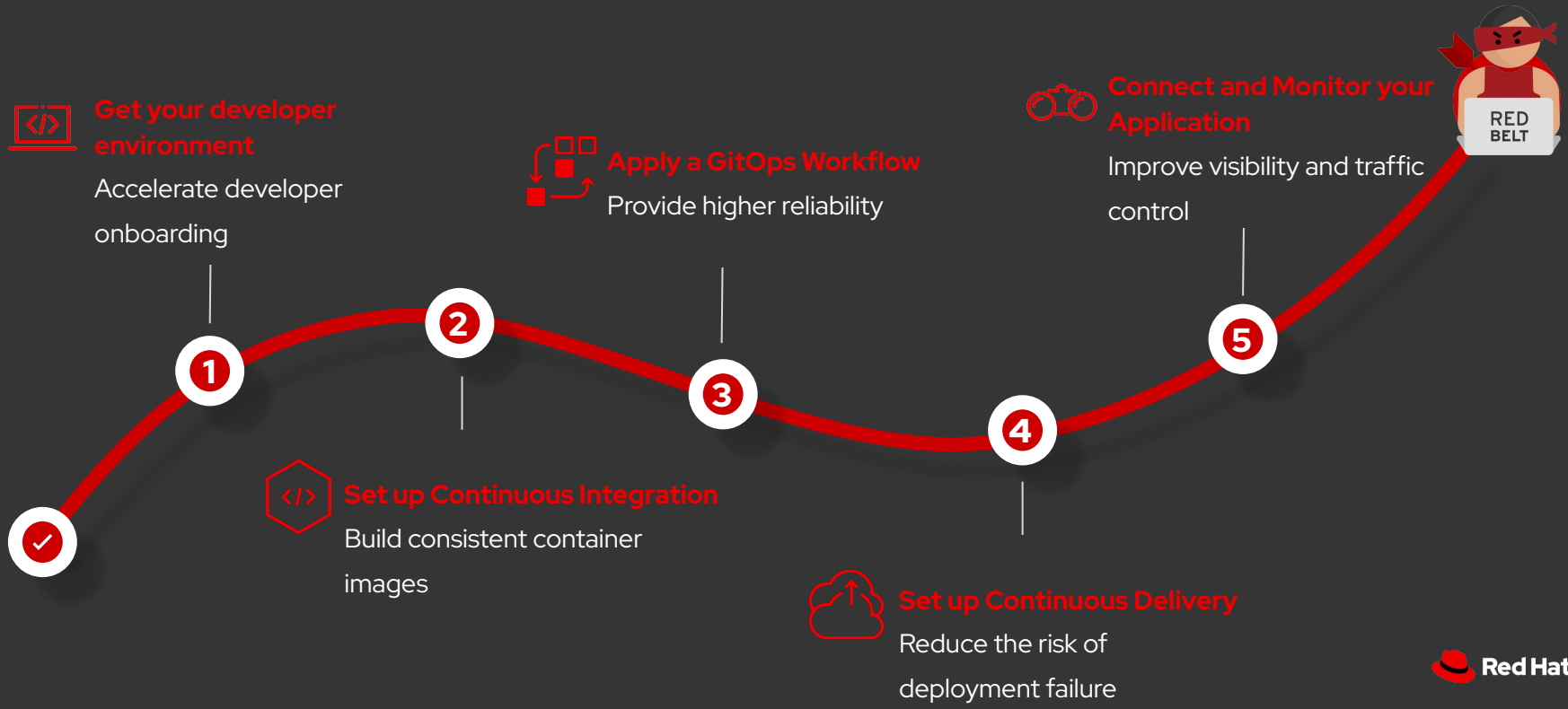
Inner Loop



*covered last week

Red Hat

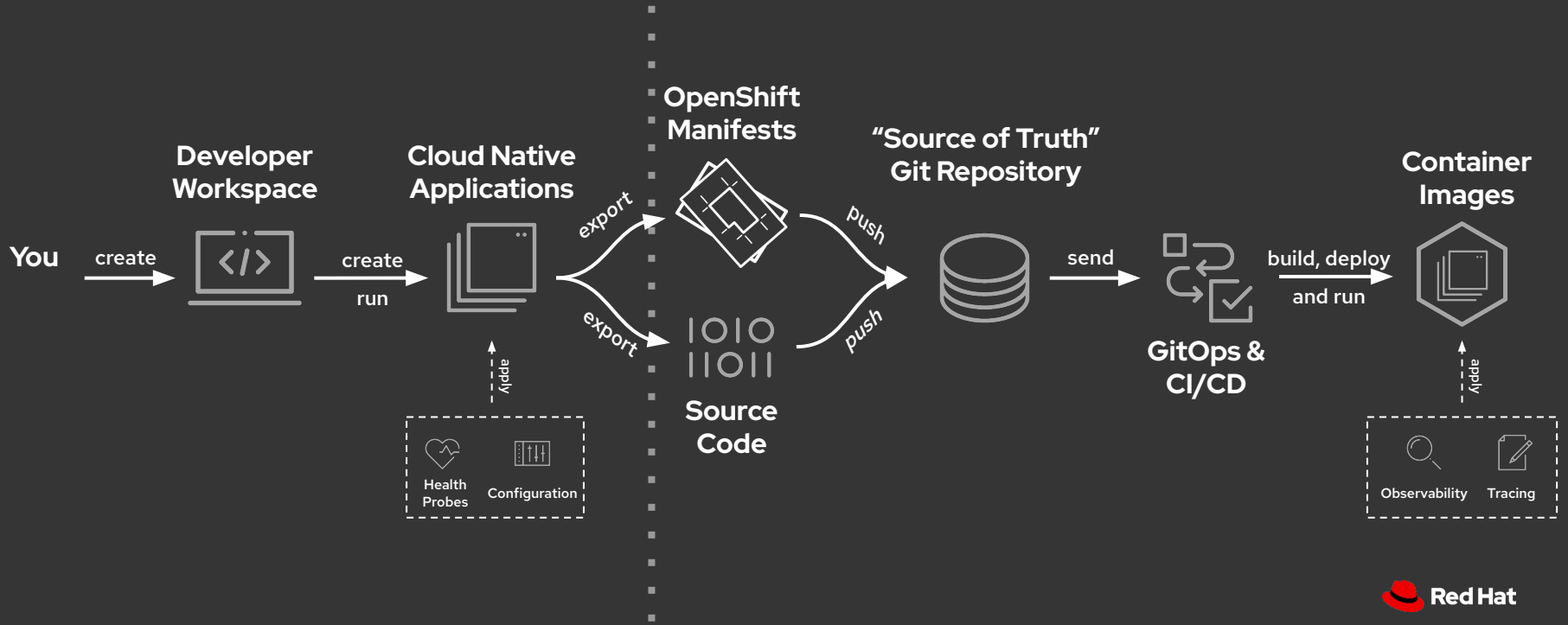
Outer Loop



Red Hat

This is the End-To-End Developer Workshop

"Inner Loop" and "Outer Loop"

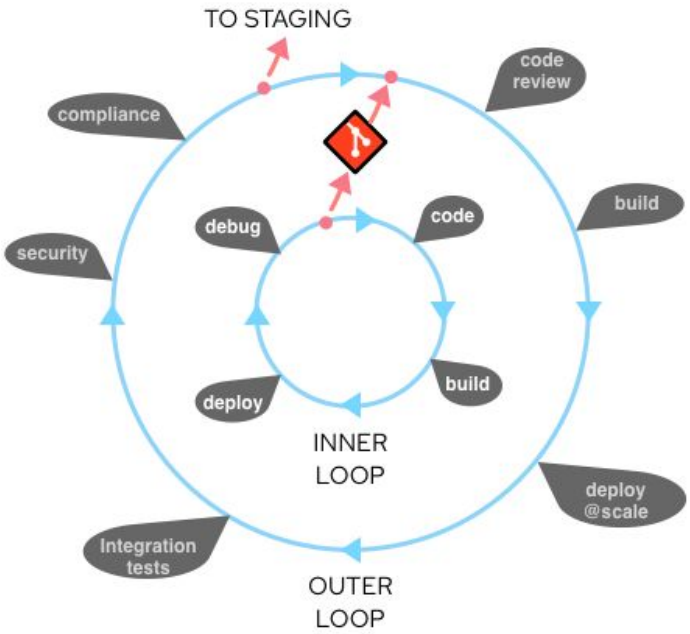


Red Hat

Developer coding cycle

How does it change with OpenShift?

How might we make the developer feel closer to the platform?



Dimensions of the problem

- How fast can we make the inner loop
- How similar to production is the run environment? I.e. how much are we actually testing?
- HW and network requirements for developer laptops.
- Standardization and compliance.

High Level Options

Option	Inner Loop Speed	Run Env	HW/network requirements	Standardization Compliance
IDE runs on laptop App runs on laptop	**secs**	**Quite different**	**High tier laptop** Low tier network	Limited
IDE runs on laptop App runs on remote OCP	**Tens of sec to few minutes**	similar	Mid tier laptop **High tier network**	Medium (the run environment can be controlled)
IDE runs on OCP App runs on OCP	Tens of second (*)	similar	Low tier laptop Mid tier network	**High**

(*) it depends on how well engineered the deployment of the IDE is.

IDE runs on laptop – App runs on laptop

Deeper Analysis

Tools such as **docker compose, podman compose**, **minikube** or Code Ready Containers can be used to bring up a local environment (*).

All of this tools require the ability to run docker/podman and or VM virtualization capability.

Windows users don’t have the greatest experience with this usually.

Free for all, things break, Kubernetes moves fast, difficult onboarding.

Laptop specs are costly.

(*) We are not considering the case where the app runs outside of a container.

IDE runs on laptop – App runs on remote OCP

Deeper Analysis

A tool watches the local file-system and on changes, builds and deploys the app on a remote developer-dedicated namespace.

These tools have features such as:

- support for complex env setup, via helm, kustomize
- hot reload,
- sometimes ability to debug

Some of these tools expect to be able to build a container image on local (still a blocker for windows user), but perhaps some of the could be configured to build remotely with something like a s2i binary build.

List of tools (non-exhaustive):

Telepresence (app actually runs locally)

Tilt

Skaffold

Odo

DevSpace

Draft

IDE runs on OCP – App runs on OCP

Deeper analysis

Code Ready Workspaces allows to run IDEs in pods and access them via browser.

One or more workspace pod per developer, typically one namespace per developer

Workspace pod defined by devfile spec.

The coding experience becomes similar to using Office 360 tools.

Considerations:

- High level of standardization and compliance (the code never leaves the datacenters, all IDE setups are controlled centrally).
- Requires high guarantees of availability before it can be approved.
- **Can face some pushback from the developers.**
- Requires significant resources on the datacenter side, while it relieves stress on the laptop side.