



# Supercharged Agility with Kanban - the EAP story

Dimitris Andreadis  
Engineering Director

@dandreadis

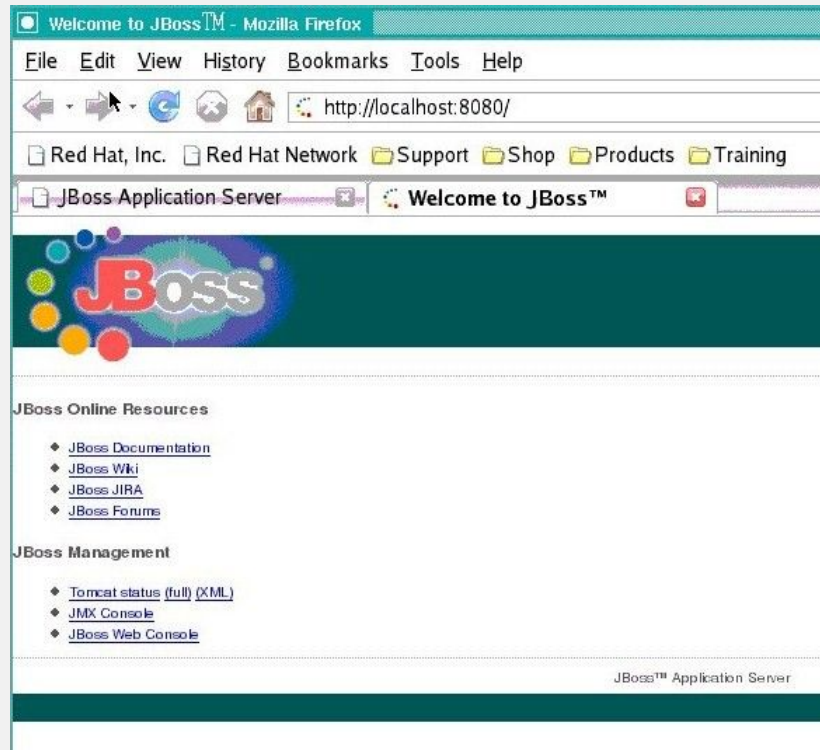
# Disclaimer

We are not an Agile Coaches, neither Kanban Experts



# Once upon a time (c. 2000) there was an Open Source project

- Code first
- Release early, release often
- Test Driven Development
- Continuous Integration
- Coding Standards
- Minimal documentation
- Peer review
- ...
- Meritocracy
- Collaboration



# Agenda

## Supercharged Agility with Kanban - The JBoss EAP Story

- Introduction
- Defining the problem
- Introducing Kanban in EAP
- Kanban Revisited
- Kanban Returns
- Tooling & Demo
- Q/A

# Defining the Problem

# A Very Complex Product

Red Hat JBoss Enterprise Application Platform - JBoss EAP

- Red Hat's 2nd biggest product after Red Hat Enterprise Linux
- Opensource Downstream for upstream WildFly community project
- Integrating 150+ components, about 50 owned by the team
- About a dozen other Red Hat products built on or depend on JBoss EAP
- Multiple delivery methods (.zip, rpm, installer, container images)
- Plus Quickstarts, Maven repos, Sources, Docs, etc.
- Multiple supported release streams, [5.x,] 6.x, 7.x
- Supported on 4 JDKs, 8 OSes, 4 Chipsets, 19 DBs
- Testing for Performance, Interoperability, Compliance, Security, etc.
- Most of team is remote, across different timezones

# EAP 7.0 Process

In a nutshell

- Pretty much Waterfall-like
  - Planning - commit on features
  - Development - do the work
  - Testing - fix bugs
- Series of builds handed off to QE every 2-3 weeks used as checkpoints
  - Developer builds (Alphas)
  - Engineering builds (Betas)
  - Release Candidate builds (potential GAs)
- Major.next releases running in parallel with previous.minor



# Defining the Problem

Sometime after EAP 7 was released, May/2016

- Long release cycles leading to insane planning times
- Misunderstood/miscommunicated requirements
- Tracking the release progress was hard
- Sticking to release schedules even harder

*We need to go Agile*



# Scrum Challenges

It's not all rosy

- How to deal with the process resourcing?
- Coordination of a large, distributed team
- Highly Specialized (veteran) Developers
- Typical lead time for feature development in the order of months

# Which problems we want to solve?

Tell me what you want, what you really really want

- Simplify planning and reduce planning time
- Reduce context switching, avoid overloading
- Be very specific about hand-off points & deliverables
- Improve requirements analysis and prioritization
- Provide enhanced visibility into the state of features

# Introducing Kanban in EAP

# Benefits of Kanban for EAP

A true Pull Model

- It can enhance an existing process
  - Respecting roles and team dynamics
  - Taking less resources to implement
- Forces you to accurately model your existing workflow
  - How you actually do the job?
  - States, inputs, outputs, handoffs, definition of Done
- Helps you visualize and track work in progress
- Iterate and continuously improve your process

# Going Agile 1.0

JBoss EAP 7.1 (WildFly 11)

- Get rid of lengthy detailed planning phase
- Agree on what will be delivered before implementing - Analysis Doc
- Promote earlier QE involvement
- Encourage developers to write community documentation

# When people think about Kanban

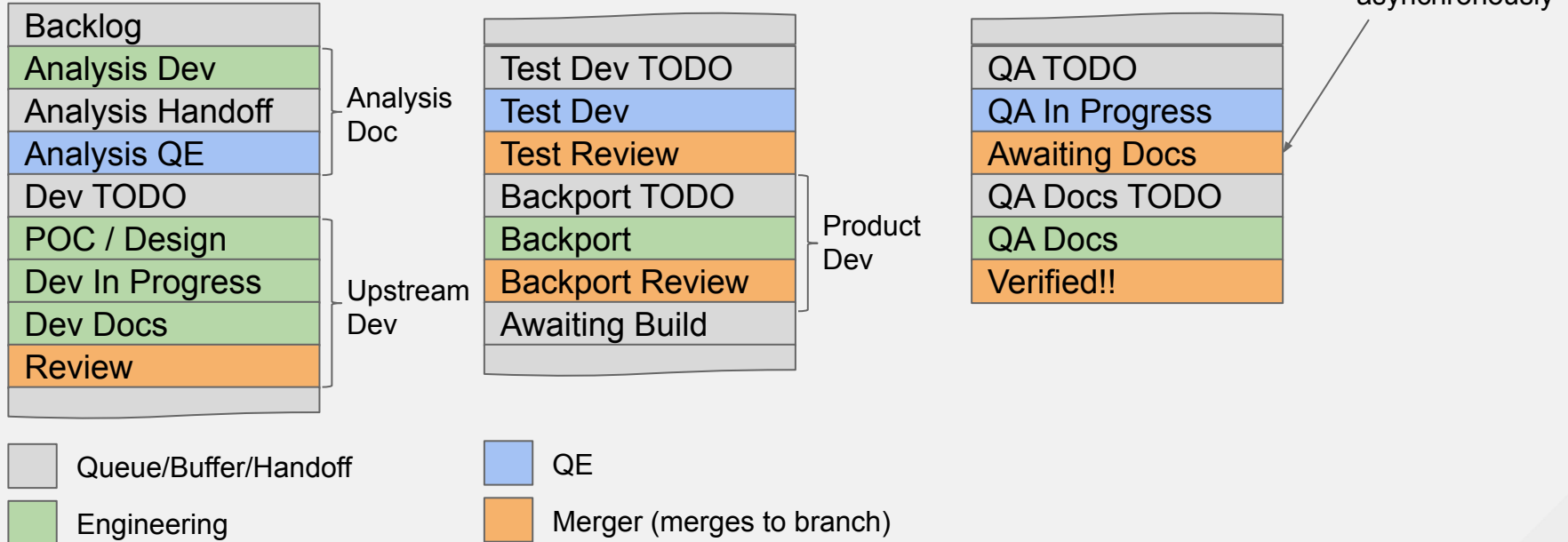
this comes to mind





# Kanban states in the real world

## Initial EAP Workflow



# Breakthrough #1 - Overbård

Make working with complex projects possible (and pleasant)

The screenshot displays the 'Overbård' dashboard for a 'Development' project. The dashboard is organized into columns representing different stages of the development process: Kickoff (21/21), Dev In Progress (34/34), Review (6/6), Pull Request Sent (1/1), Product Docs (7/7), and Verified (0/0). Each task card includes a title, a brief description, a progress bar, and a set of status icons (AD, CD, TP, TD, PC, DA, PD). A red gear icon is visible in the bottom right corner of the dashboard.

| Development   |  |   |  |  |              |          |
|---|--|---|--|--|--------------|----------|
| 62/62   |  |   |  |  |              |          |
| B   | Kickoff  | Dev In Progress   | Review   | Pull Request Sent  | Product Docs | Verified |
|   | 21/21  | 34/34   | 6/6  | 1/1  | 7/7          | 0/0      |
| EAP7-1114 SM :<br>Jakarta EE 8 Certification  | EAP7-830 EH :<br>Add setting to control global memory usage for messaging journal                            | EAP7-1028 JP :<br>Expose EAP Logging Data so that it can be consumed by Kiali (istio) Console | EAP7-1165 TH :<br>Provide a way to configure journal-file-open-timeout attribute of Artemis server | EAP7-1026 JM :<br>Expose EAP Metrics so that it can be consumed by Kiali (istio) Console |              |          |
| EAP7-720 FR :<br>Multiple delivery group support for MDBs in EAP-7  | EAP7-488 DL :<br>Fine-grained control on elp3 thread-pool and reuse idle threads in pool before creating new | EAP7-675 MM :<br>Automatic detection of file based KeyStore types                             |  | EAP7-726 SD :<br>SNI support for https-listeners   |              |          |
| EAP7-474 DL :<br>Ability to add a server-side configured interceptor for the applications without changing the deployments  | EAP7-1006 EH :<br>Support JMS Java EE resources definition for remote Artemis-based broker                   | EAP7-993 DL :<br>Add support for JSR-375 EE Security as required for Java EE 8 (Elytron)      |  | EAP7-1096 DL :<br>Registration of JVM wide default SSLContext (Server side)              |              |          |
| EAP7-475 DL :<br>Ability to add a global client side interceptor for deployed applications without changing the deployments | EAP7-1088 PF :<br>Determine the primary provider of Singleton Service using CLI                              | EAP7-1060 HP :<br>Make "Expose EAP Metrics" configurable                                      |  | EAP7-649 DL :<br>JASPIC ServletProfile Integration with Elytron                          |              |          |
| EAP7-892 BS :<br>Allow for custom CLI scripts as part of the S2I build process.   | EAP7-1154 DV :<br>HTTP Basic Authentication - Silent Operation   | EAP7-1087 RH :<br>Integrate mod_cluster ramp-up mechanism when starting new nodes.            |  | EAP7-1067 DL :<br>Audit logging by default has a significant impact on performance       |              |          |

# Kanban Revisited

# Going Agile 1.1

Still during EAP 7.1 (WildFly 11)

- Kanban is very "Serial"
  - Reality is more "Parallel"
- Introduce **Parallel Tasks**, for example:
  - Upstream, Downstream  
Test Development, Community Docs
- Each Parallel Task has its own states, like:
  - TODO, In Progress,  
Review, Merged / Done



# Going Agile 1.1

## Revised Workflow

|                 |
|-----------------|
| POC / Design    |
| Dev In Progress |
| Dev Docs        |
| Review          |
| Test Dev TODO   |
| Test Dev        |
| Test Review     |
| Backport TODO   |
| Backport        |
| Backport Review |
| Awaiting Build  |

Collapsed -  
Detail is in the  
Parallel Tasks

|                        |
|------------------------|
| Backlog                |
| Analysis Dev           |
| Analysis Handoff       |
| Analysis QE            |
| Dev TODO               |
| POC / Design           |
| <b>Dev In Progress</b> |
| Review                 |
| Awaiting Build         |
| Verification TODO      |
| QA In Progress         |
| Awaiting Docs          |
| QA Docs TODO           |
| QA Docs                |
| Verified!!             |

Flexibility in which PTs must  
be done. E.g test dev does  
not **need** to delay merging.

Chance for PTs to catch up.  
Once all done QA can start.


# Breakthrough #2 - Parallel Tasks

Introduce a 3rd dimension to the board

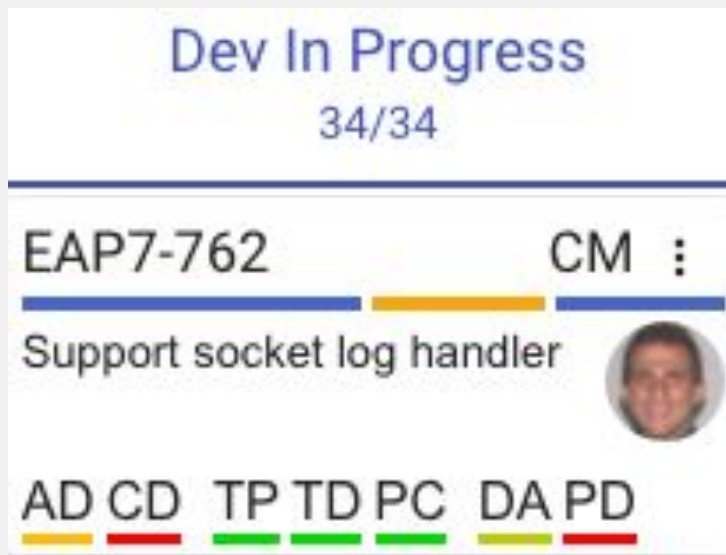
Dev In Progress  
34/34

---

EAP7-762 CM :

Support socket log handler 

AD CD TP TD PC DA PD



↑

Test Dev



# EAP 7.1 Retrospective

Improvements in some areas

- Planning went smoother
- Analysis phase/document helped
- More visibility into what is being worked on
- Move to Parallel Tasks gave a truer representation of the state of features
- In many cases test development happened earlier

# EAP 7.1 Retrospective

Still not perfect

- Dev cycle still quite short compared to testing
- Once something is merged, very hard to remove
- Testing still came late for some features
- Still too many handoffs in the process causing delays



# Kanban Returns

# JBoss EAP continuous delivery (EAP-CD)

Cloud first!

- New EAP delivery stream based off of WildFly quarterly releases
- Targeting primarily OpenShift Cloud based deployments
- Every 3rd or 4th release rolls into a traditional EAP release

# Timeboxed releases

Timeboxed **means** timeboxed

- WildFly gets released every three months
- Work happens on **feature branches** by **feature teams**
  - Developer (Engineering), Tester (QE) and a Writer (Docs)
- WildFly master needs to be **stable** and **always releasable**
  - Pre-merge requirements
  - Final components only
  - On-going regression tests of master

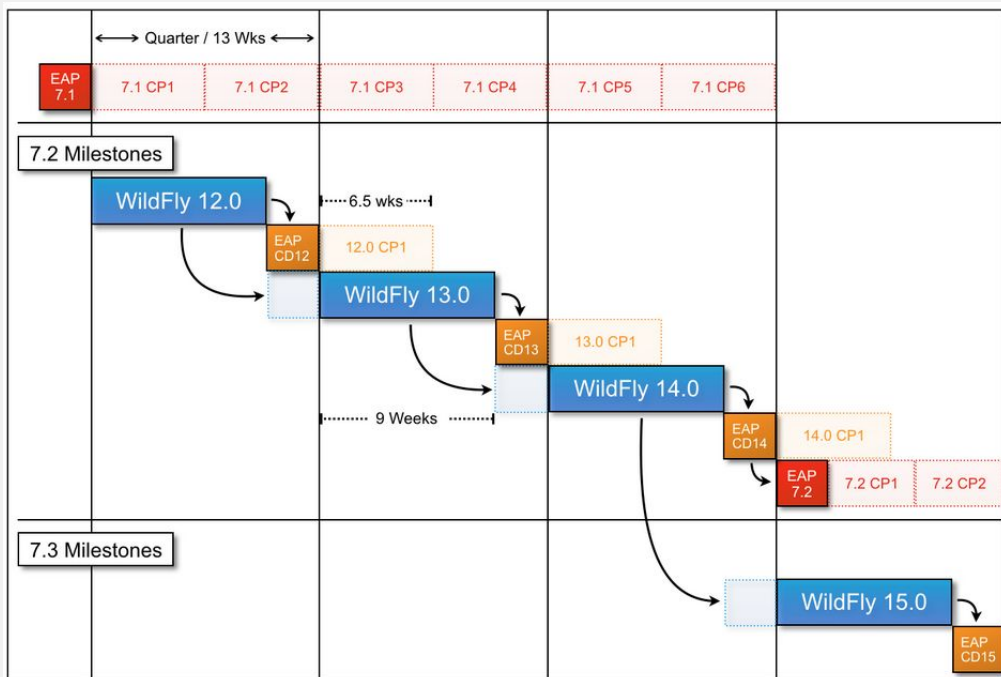
# Stable releases

## pre-checks

- To keep master always releasable, nothing gets merged until all the **pre-merge requirements** are met
  - Analysis Document
  - Implementation
  - Community Documentation
  - Test Plan
  - Test Implementation
  - Pre-Verification
- Despite our efforts if something breaks
  - Nothing is reverted
  - Treat it like a normal bug

# Breakthrough #3 - Timeboxed Releases,

Pre-test as much as possible & Merge when Done - Really Done



EAP7-1029

JM :

Implement EAP Health

Check so that it can be

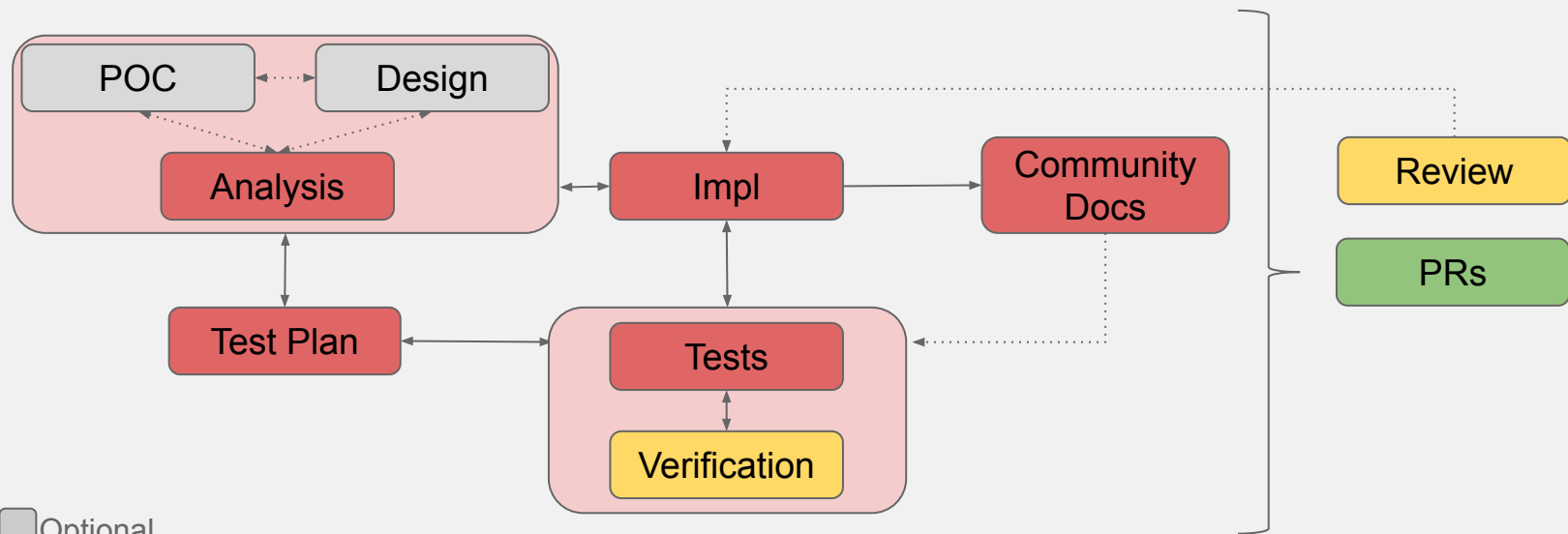
consumed by Kiali (Istio) Console

AD CD TP TD PC DA PD



# Feature delivery

Development is iterative & parallel, not sequential



Not really a linear process!

# Tooling & Demo

# Tooling

Beating Jira into shape

- Continuing to use Jira a hard requirement
- Some external tools integrate with Jira, but not satisfactory enough
- Evaluated Jira Agile - main problems:
  - Jira Agile cannot visualize too many states into the viewport
  - Filter and swimlane views are too static
  - It doesn't understand our parallel tasks



# Overbård

To the rescue

- In-house developed Jira plugin -
  - Current: <https://github.com/overbaard/overbaard>
    - For 7.1: <https://github.com/jirban/jirban-jira>
- Designed to solve problems in Jira Agile
- Apart from the board's config, uses only data from the Jira issues
- Deals with a lot of states, supports horizontal scrolling & collapsible states
- Supports arbitrary dynamic filtering with zero config
- Understands our parallel tasks extensions
- Caches data in the client so it's super fast!

Demo....

# Recap

# Is it really working?

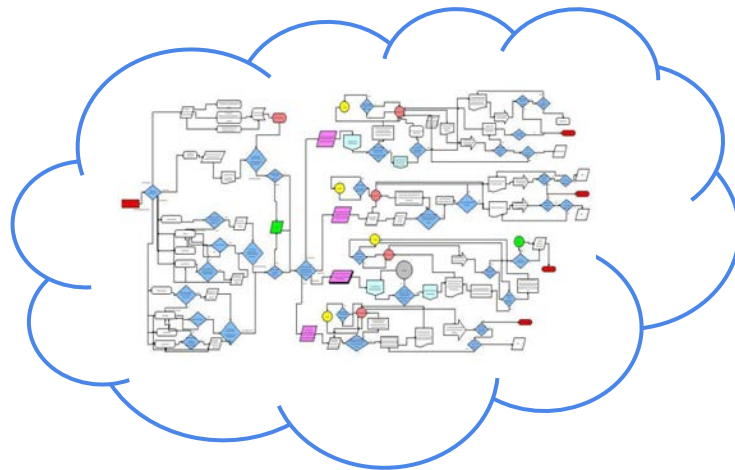
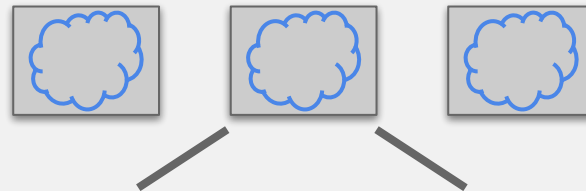
Let's look at the results

- Community project (WildFly) released **Quarterly**
- New **EAP-Continuous Delivery** stream for OpenShift (EAP-CD)
- Improvements for traditional **bare metal** EAP releases
- Our Agile Process was **run by ONE person** (who did the tooling, too)

# Wrap up

Pull model FTW!

- **Kanban can work with very complex projects** (with some tweaks)
- Kanban helps focusing on **improving how you actually work**
- **Tooling is necessary** to make it work (we'd happily share ours - Overbård)



# Call To Action: try Overbård!

<https://github.com/overbaard/overbaard>

## Documentation Home

'Bård' is a Norwegian name pronounced 'board' (alternative spelling: 'baard').

Overbård is a lightning fast Kanban board integrating with Jira. It is shipped as a Jira add-on, and was written to address some of Jira Software's short-comings when it comes to how it displays the boards. Effectively this means:

- horizontally scrollable boards, with collapsible columns.
- making setup of filters and swimlanes less static, minimising the need for configuration. Since we are using a 'fat' browser client, all changes to the view purely happen on the client side with no need for extra round trips to the server.

Apart from the configuration, all the data and permissions are driven by your Jira instance.

Read the guides to see more of our innovative features such as:

- setting up custom fields to filter and create swimlanes for
- parallel tasks
- displaying linked issues from other projects
- the ability to configure independent workflows based on issue type
- displaying issues from several issues on one board

## Guides

These guides are listed in order of 'complexity'.

- [User Guide](#) - for end users of Overbård.
- [Admin Guide](#) - for administrators to configure new boards.
- [Developer Guide](#) - how to set up the environment for contributors to the project.

# Q&A

# THANK YOU



[plus.google.com/+RedHat](https://plus.google.com/+RedHat)



[facebook.com/redhatinc](https://facebook.com/redhatinc)



[linkedin.com/company/red-hat](https://linkedin.com/company/red-hat)



[twitter.com/RedHatNews](https://twitter.com/RedHatNews)



[youtube.com/user/RedHatVideos](https://youtube.com/user/RedHatVideos)