



Understanding Code Management in a Multi-Vendor Environment

**Examples of code management in a
multi-team environment**



About this Presentation



- ▶ **This presentation was prepared as part of the support materials for the IAE Industry Day on Transparency on May 13, 2014.**
- ▶ **Purpose is to provide a very basic outline of how code management will work within the new IAE environment where there will be multiple concurrent development teams operating.**
- ▶ **Assumes sprints are aligned across development teams.**

Over and above the basic need to provide a versioned, secure, managed code repository, the particular goal of Code Management within IAE are to ensure that:

- ▶ Developers can can quickly and reliable access to code
- ▶ The program can respond to emergency changes while minimizing risk of introducing defects
- ▶ The public and community can see code throughout development (with some limitations)
- ▶ DevOps and Continuous Integration operations (CI) can execute to integration code from multiple independent development teams
- ▶ Deployment to production is controlled and minimizes configuration-code failures

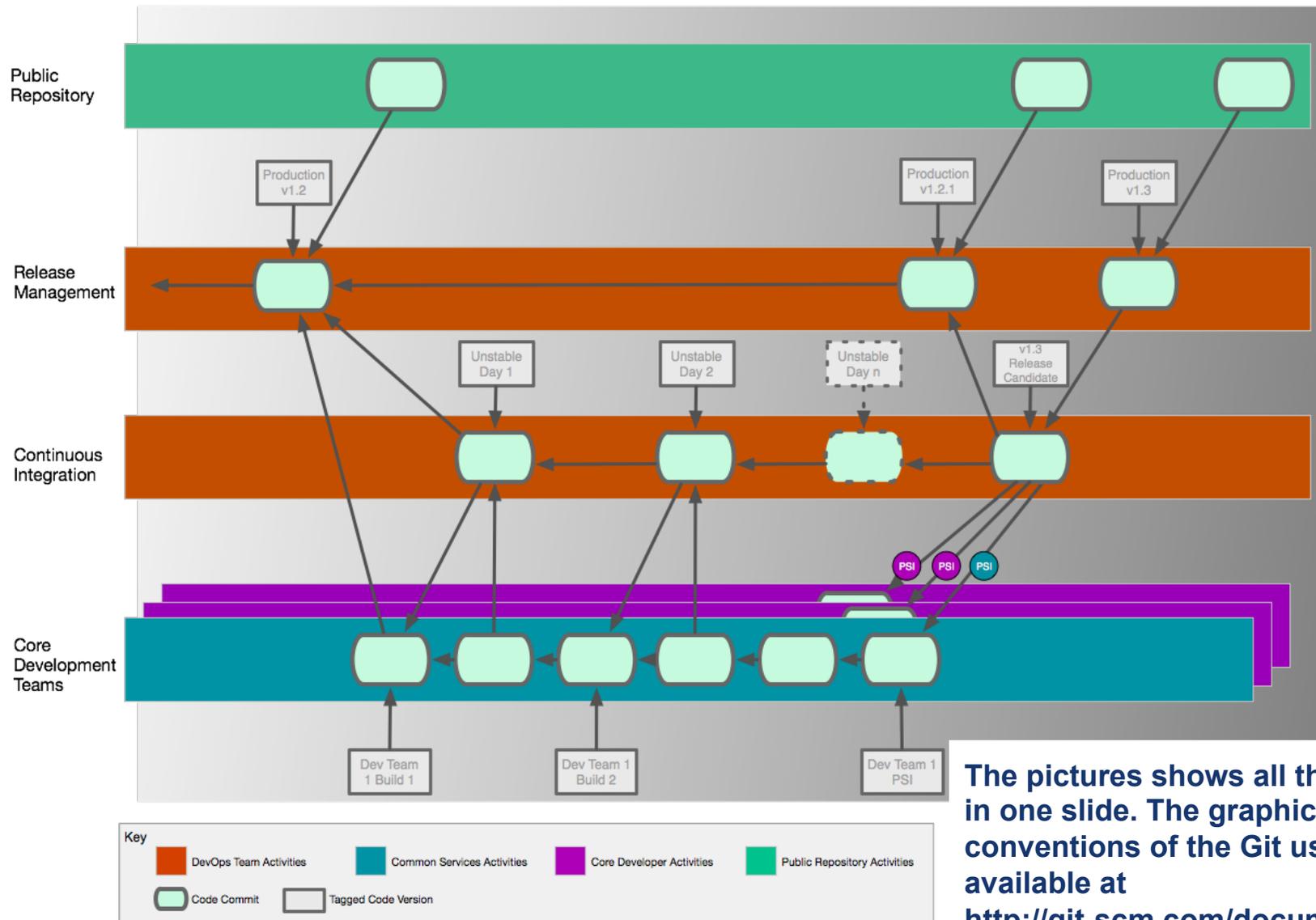
Note: we are not talking about the Continuous Integration activities around Code Management here.

Scenarios:

- ▶ **The DevOps/CI team creates daily, unstable builds by merging changes from multiple development teams**
- ▶ **A development team starts from a stable release and then pulls down daily unstable releases as they progress through a sprint**
- ▶ **A release candidate is created by DevOps/CI team**
- ▶ **A production version is created from a release candidate**
- ▶ **An urgent patch is applied to the current release**

The Complete Picture

Code Management Workflow from IVV/DevOps Perspective

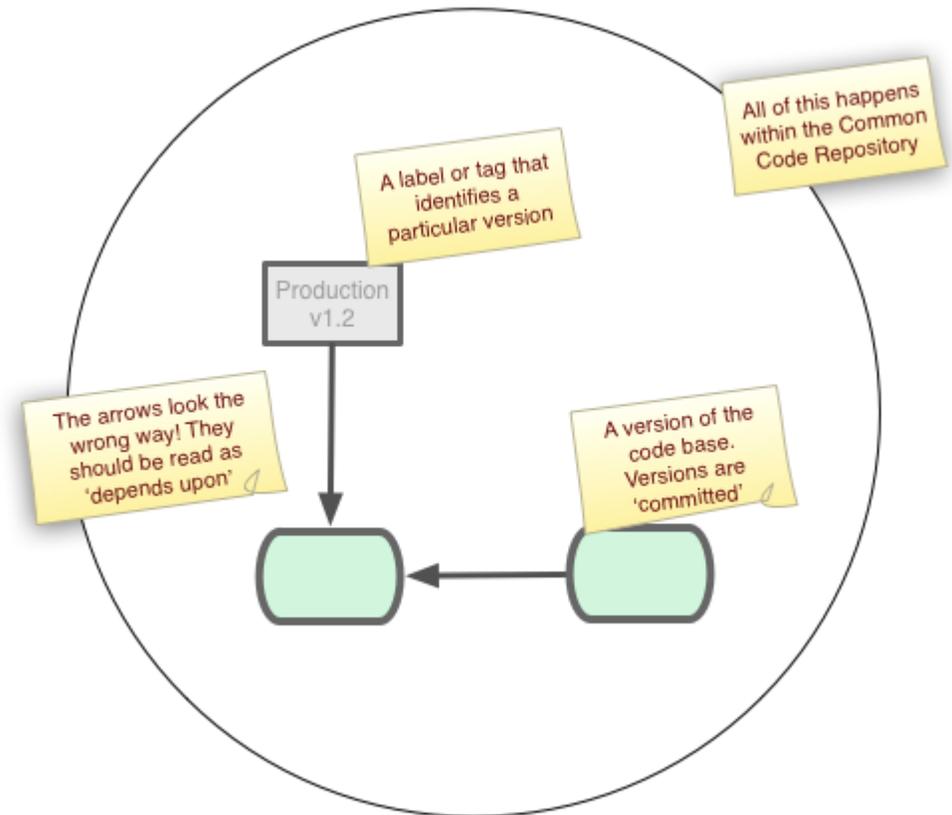


The pictures shows all the scenarios in one slide. The graphic uses the conventions of the Git user guide available at <http://git-scm.com/documentation>

Key Terms:

- ▶ Version
- ▶ Tag
- ▶ Commit
- ▶ Depends Upon

All this happens within the common code repository. Development teams and individual developers will have their own, replicated copy of the repository.

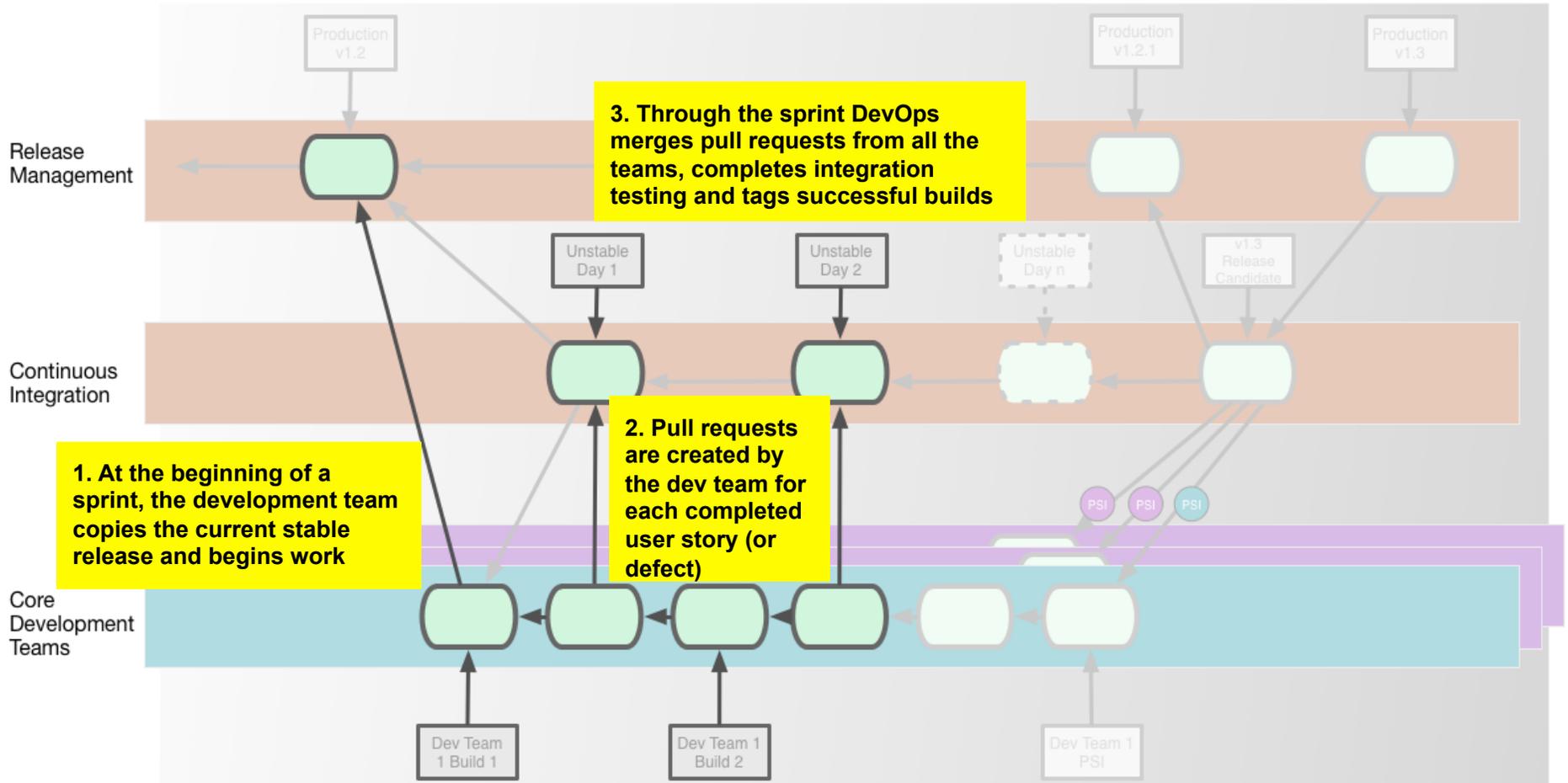




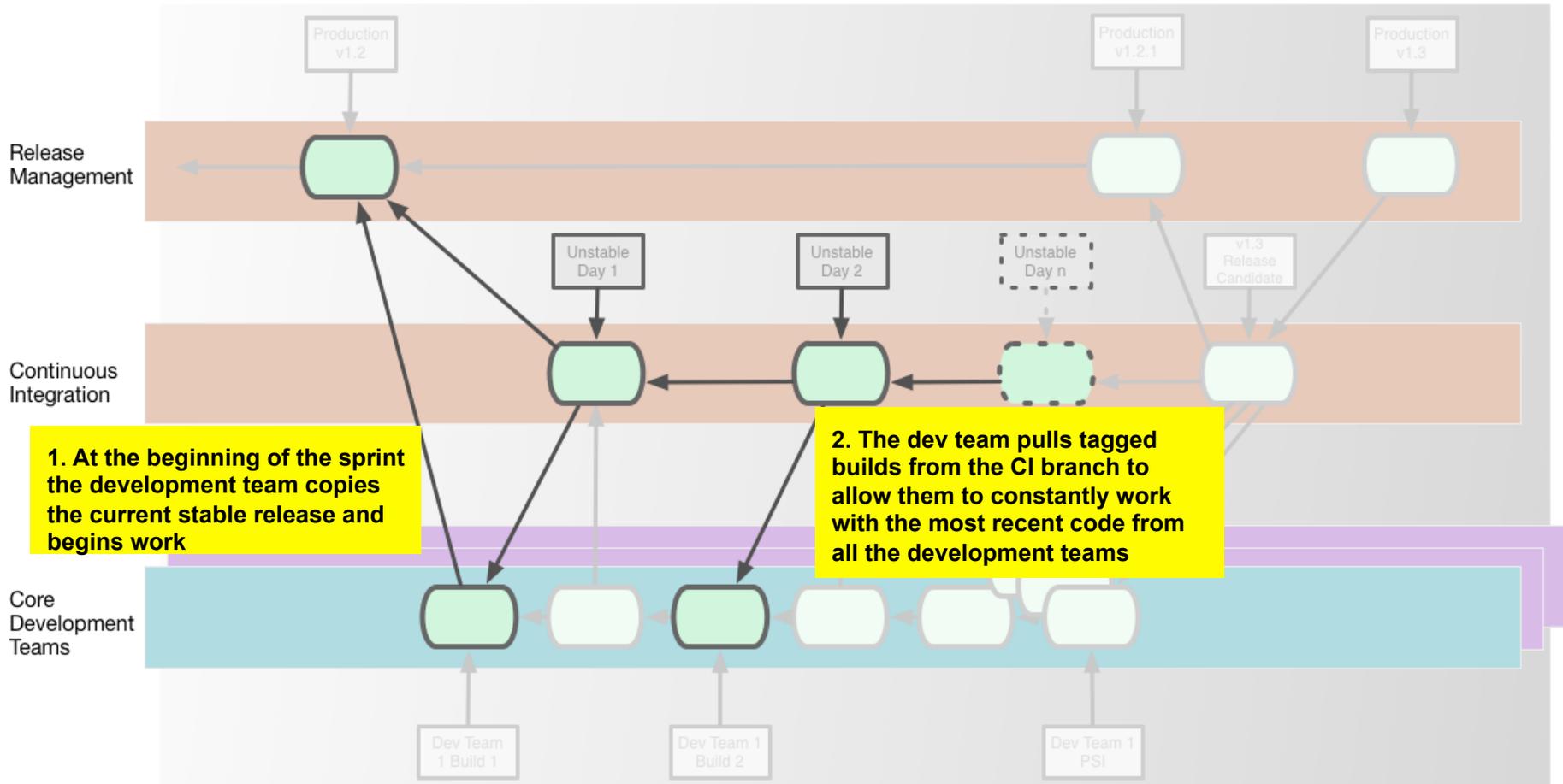
One Note...

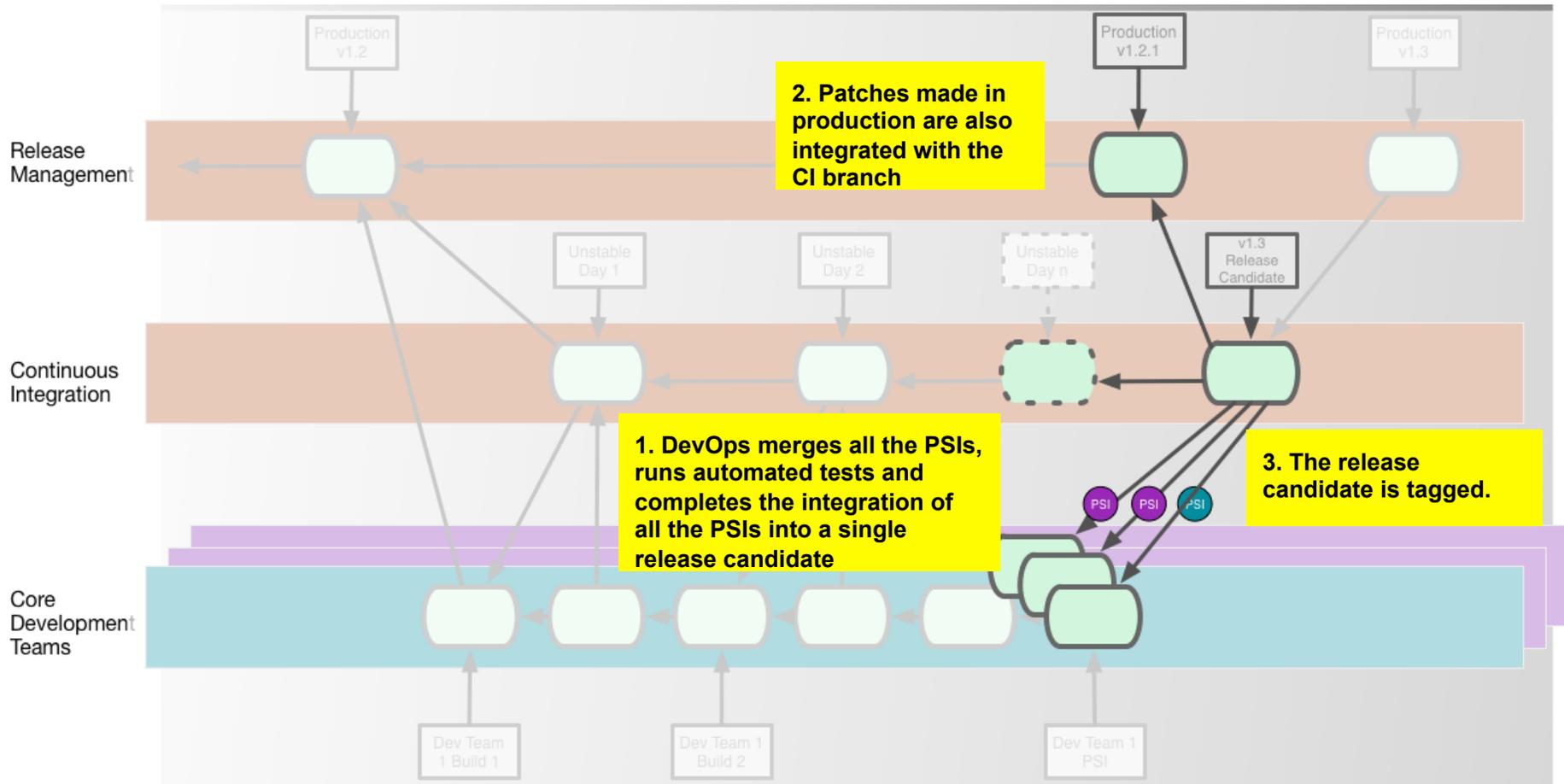


- ▶ For each of these scenarios, we are going to show you what is happening to the code.
- ▶ When we “create a new release” there are lots of activities that go with that including multiple layers of testing and acceptance; we are not showing all these activities.
- ▶ But understanding how the code will be managed helps us define the activities that are not shown and who will be responsible for them.



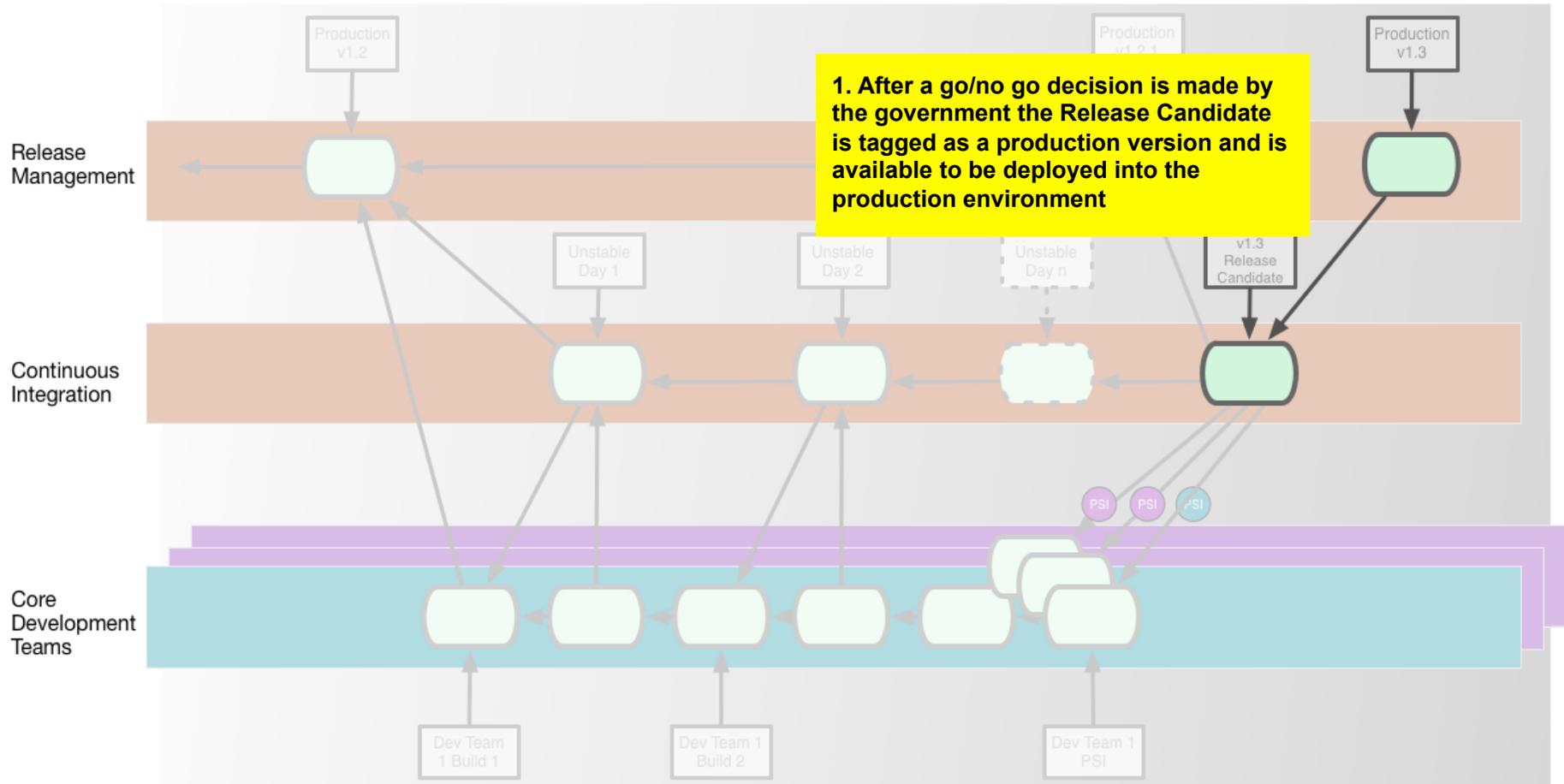
Providing Daily Builds to Application Development Teams

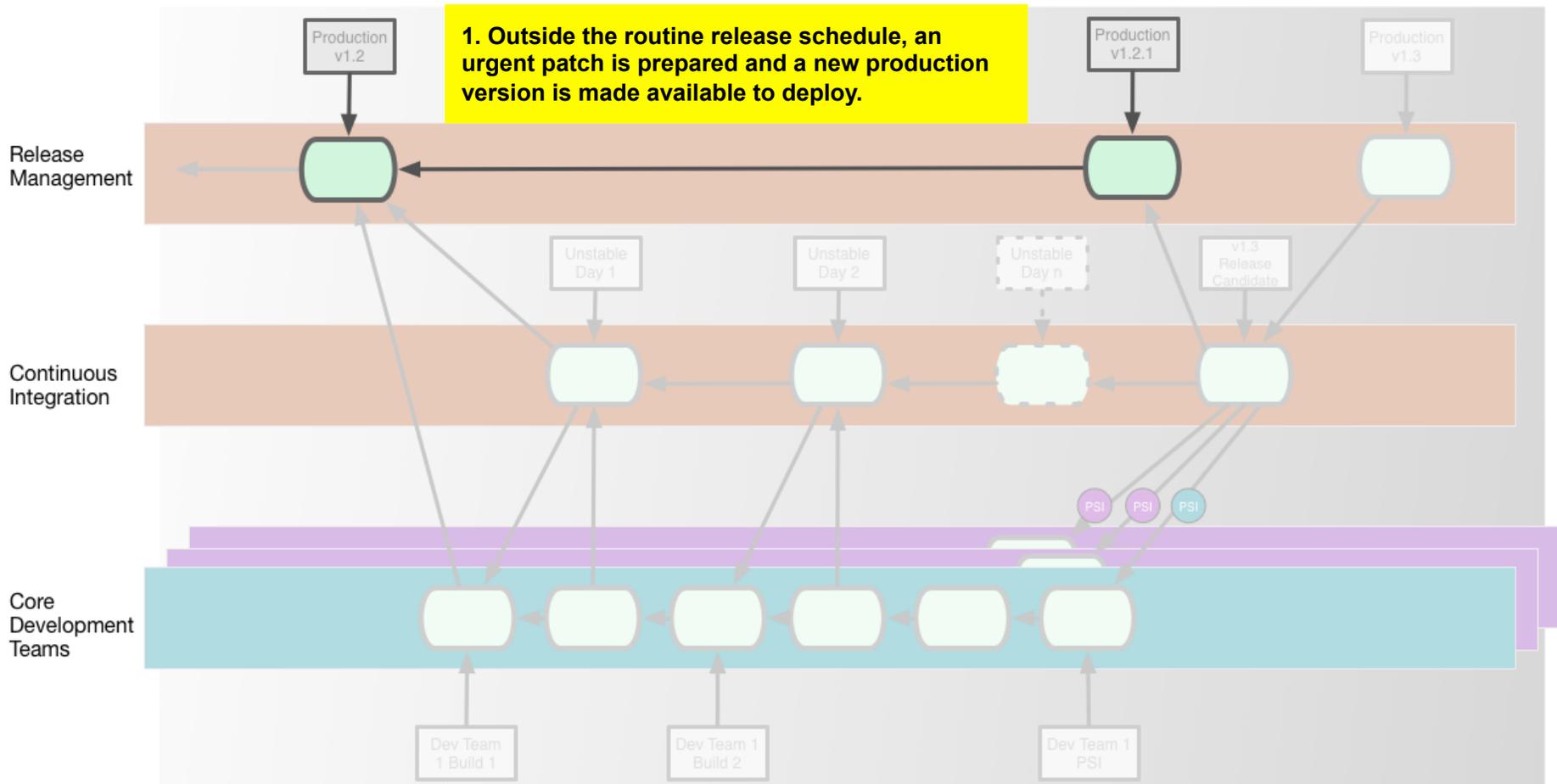




PSI: Potentially shippable increment.

Designating a Release Candidate as a New Production Version





In this particular example, the detail of how and where the patch is created that is applied to produce v1.2.1 is not shown. In reality, there would be the same commit, pull requests and integration activity for an emergency patch. The intention is to show how out-of-band changes to production can later be integrated into the main branch.

