Performance Management Group

fortiss

Bergamo, 01/09/2015

Model-Based Performance Evaluations in Continuous Delivery Pipelines

1st International Workshop on Quality-aware DevOps (QUDOS 2015)

Markus Dlugi

Andreas Brunnert

Helmut Krcmar

fortiss GmbH
An-Institut Technische Universität München

Agenda

- Motivation
- Architecture
- Live Demo

Motivation

Continuous Integration, Delivery, Deployment

- **Continuous Integration (CI)** ensures a **usable** application during all stages of the development process

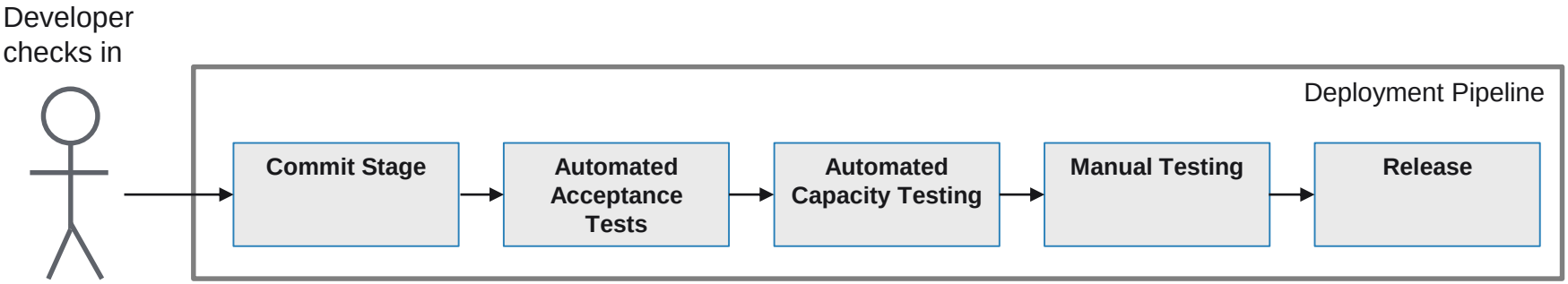
- **Continuous Delivery (CD)** demands that the application is production **deployable** for every successful release candidate

- **Continuous Deployment** is the practice of **actually deploying** every application version that has passed the necessary tests to production

Humble & Farley (2010)

Motivation

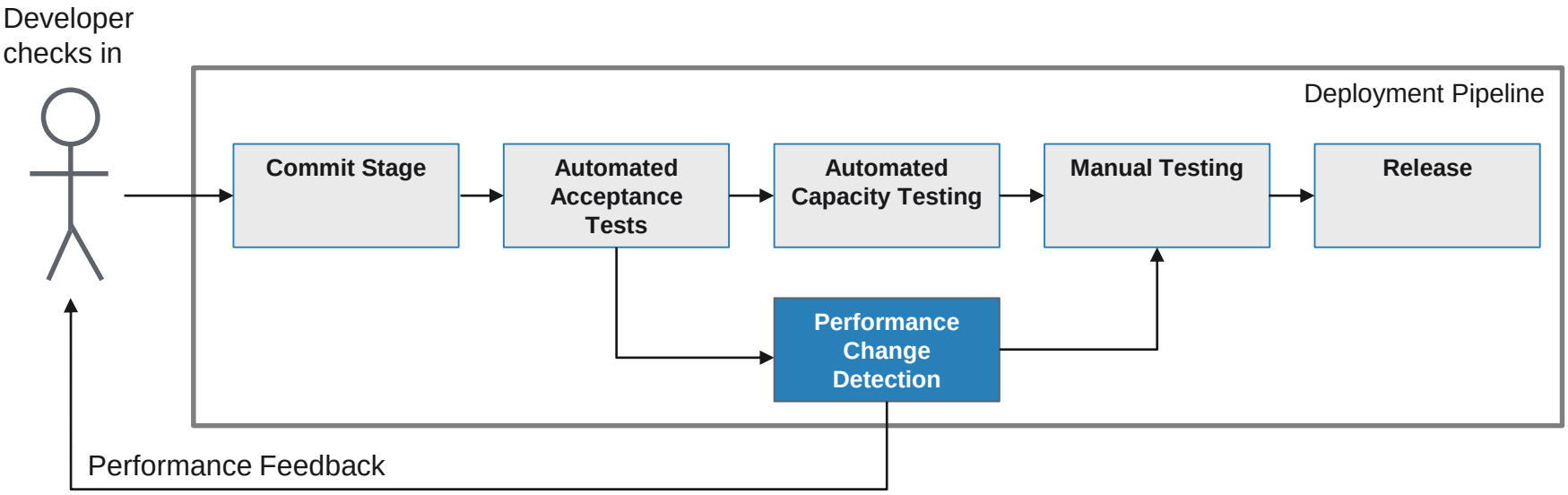
Deployment Pipeline



Humble & Farley (2010)

Motivation

Modified Deployment Pipeline



Brunnert & Krcmar (2014)

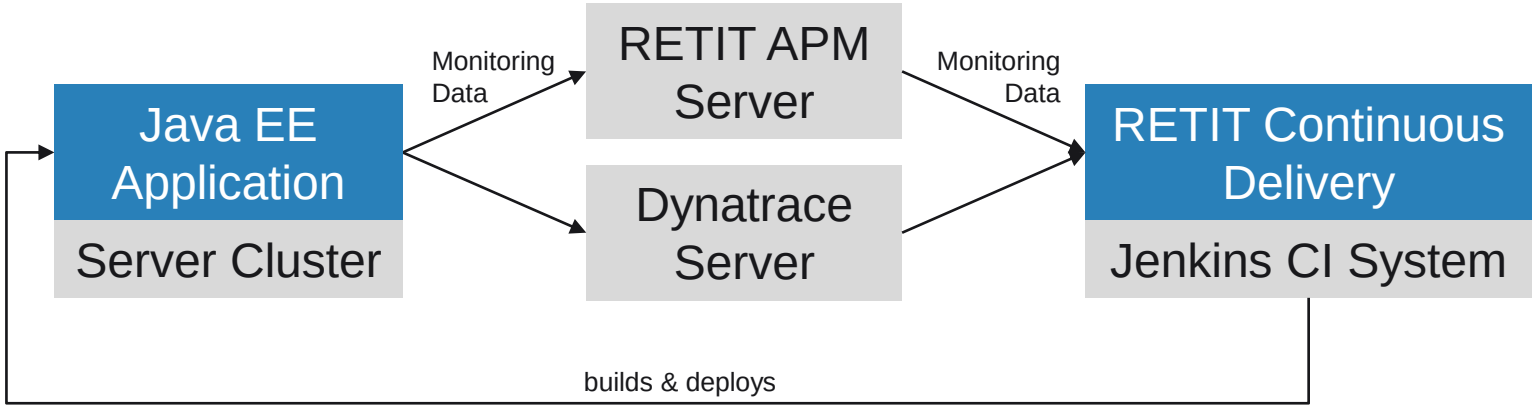
Motivation

Benefits

- Detect performance changes continuously to prevent regressions and help maintain SLAs
- Provide stable performance tracking mechanism independent of testing environments
- Improve integration and collaboration between development and operations teams

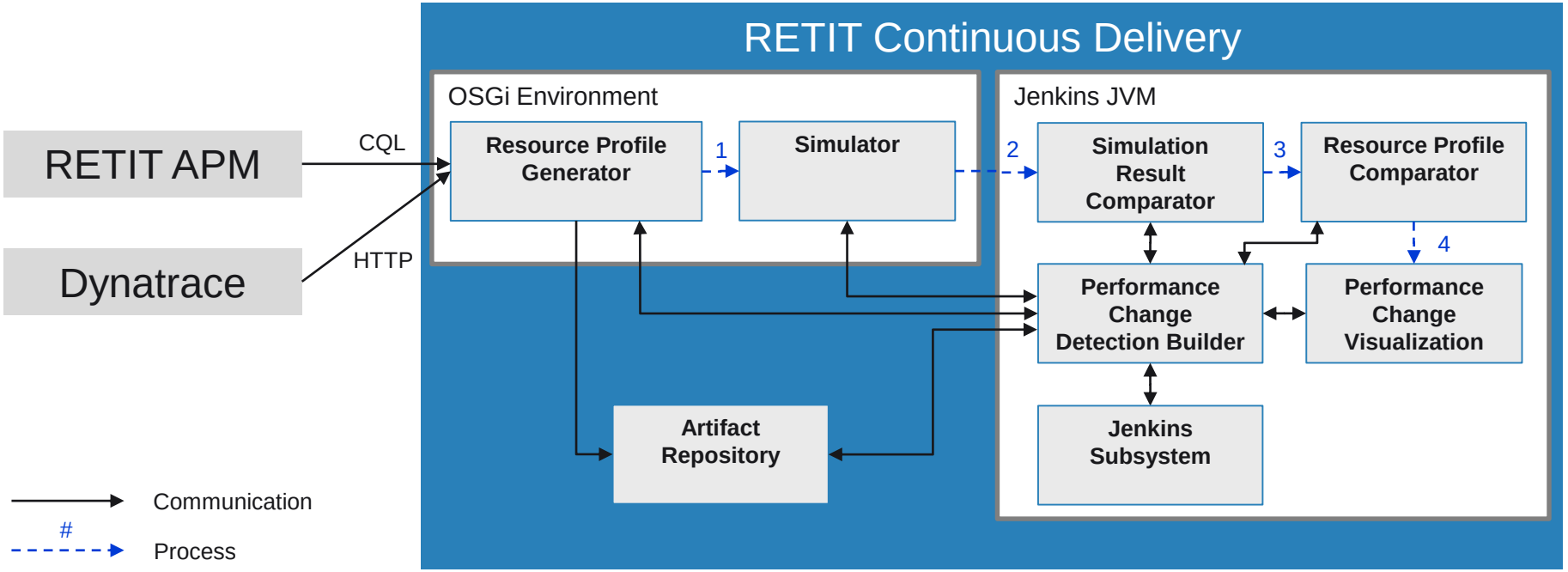
Architecture

System Context



Architecture

Architectural Overview



Demo

Method control flow

```
@WebMethod
public boolean sellInventory(Integer userID, long inventoryID, boolean
rollback){
    Customer customer = getCustomer(userID,true);
    CustomerInventory inventory = getInventoryItem(inventoryID,
    customer.getId());
    if (inventory == null)
        return false;
    customer.changeBalance(inventory.getTotalCost());
    customer.getInventories().remove(inventory);
    em.remove(inventory);
    if (rollback) {
        mySessionCtx.setRollbackOnly();
    }
    return true;
}
```

Demo

Modified method control flow

```
@WebMethod
public boolean sellInventory(Integer userID, long inventoryID, boolean
rollback){
    for(int i = 0; i < 50000; i++) {
        Math.random();
    }
    Customer customer = getCustomer(userID,true);
    CustomerInventory inventory = getInventoryItem(inventoryID,
    customer.getId());
    if (inventory == null)
        return false;
    customer.changeBalance(inventory.getTotalCost());
    customer.getInventories().remove(inventory);
    em.remove(inventory);
    if (rollback) {
        mySessionCtx.setRollbackOnly();
    }
    return true;
```

Demo

Live Demo

References

- Brunnert, A., Krcmar, H. (2015). *Continuous Performance Evaluation and Capacity Planning Using Resource Profiles for Enterprise Applications.* Journal of Systems and Software (JSS), 10.1016/j.jss.2015.08.030
- Brunnert, A., Krcmar, H. (2014). *Detecting performance change in enterprise application versions using resource profiles.* In *Proceedings of the 8th International Conference on Performance Evaluation Methodologies and Tools, VALUETOOLS ,14,* pages 165-172, ICST, Brussels, Belgium.
- Humble, J., Farley, D. (2010). *Continuous delivery: reliable software releases through build, test, and deployment automation*: Pearson Education.
- Dehling, H., Haupt, B. (2006). *Einführung in die Wahrscheinlichkeitstheorie und Statistik*: Springer-Verlag.
- Stephens, M. A. (1992). An Appreciation of Kolmogorov's 1933 Paper

Markus Dlugi
Andreas Brunnert

dlugi@fortiss.org
brunnert@fortiss.org
performancegroup@fortiss.org

pmw.fortiss.org

Questions for the audience

- How should the Simulation Result Comparator be built to ensure that significant performance changes are detected while simultaneously being robust enough to avoid false negatives?

- How can sensible thresholds for metrics like the mean/median response time be determined?