DevOps Performance Breakout Group Results

Workshop on Challenges in Performance Methods for Software Development (WOSP-C)

Austin, TX, USA, January 31st, 2015

Results summarized by Andreas Brunnert (brunnert@fortiss.org)
Supporting DevOps Processes by Integrating SPE and APM Activities

Supporting information exchange in the direction Dev -> Ops by:
- Using performance models to describe a new software release
- Use performance unit tests / performance change detection techniques (automated & continuously)
- ...

Supporting information exchange in the direction Ops -> Dev by:
- Using monitoring data to derive representative workloads
- Using monitoring data to parameterize design-time performance models (e.g., for reused components)
- ...
Challenges for SPE/APM in DevOps

• How to realize DevOps setups from an organizational perspective? (e.g., putting Dev and Ops people into the same team, Noops)

• How to assign responsibilities for SPE/APM activities/tasks in DevOps teams? (e.g., when different organizations are involved; moving tasks from Ops to Dev)

• How to overcome the issue that Dev-Teams often deal with a specific application whereas Ops-Teams often have to deal with a huge amount of applications? (e.g., if Dev-Teams provide information for a specific application to Ops-Teams, how do they correlate this information for all applications they need to maintain)

• How to include traditional load test in rapid release cycles? (e.g., more than 10 releases per day)

• Versioning of performance metrics collected during release tests

• Is it feasible to neglect performance during release and to rather test it on a subset of users in production?
Results on the Whiteboard

- Movie tasks back and forth?
  - Tests → Ops?
- Deploying production environments?

- Organizational Perspective
  - Ops vs. Dev separate
  - "in" more teams

- Handling multiple app-performance models?
  - Dev vs. Operations
- Active vs. Design-time activities
  - Can performance be neglected during release? / developed?

- Versioning
  - The shorter the cycle, the better
- Performance testing is also missing
  - depends on the type of application

- Work and Monitoring Data
  - Loop alignment

- Dev Ops
  - Build
  - Test
  - Deploy