Continuous Delivery @ SOLEIL
Soleil context

- 30 beamlines 24/7 operation since 10 years
- Small Control & Acquisition team
- High user pressure to regularly deliver new features
- Continuous Integration since 5 years
- Moving to Continuous Delivery
Continuous Integration

Detecting software defects at earliest stage possible

- Version Control System
- Automated build
- Team agreement
Expected Benefits

- Reduce error-prone Manual actions
- Reduce Time to delivery
- Federate Team
- Control changes
- Reduce incidents
Our pipeline V1

CI Server → Artifact Repository → Prod environment

Prod environment:
- 150 OpenVZ containers
- 200 industrial Windows
- 100 Exotic plateform

ARCHIVE

DEPLOY
Soleil issues

- Many modules
- Wide variety of production environments
- Big packages, too few deployments
- Too few deployments
- Few tests
- Dev team not fully committed to CD
- Too heavy deployment pipeline
Pipeline V2: CI platform

- Stuck to old versions of our build tool (Maven 2). Big work to upgrade
- Try reduce module number
- Enhance Team commitment
Pipeline V2: UAT

Add User Acceptance Test (UAT) platform
Pipeline V2: deployment

• Review packaging
• Use automated tools to check for the conformity of the deployment
• Currently studying tools like Chef, Puppet or CFEngine
Pipeline V2: monitoring

• Already monitoring there but no systematic use

• Must be tuned to raise relevant alarms
DevOps

Enhance collab between Dev & Ops:
- Operationals closer to business requirements
- Developers closer to operation
Pipeline V2: Change management

• Define a clear changing workflow
• Deploy more frequently (every week?)
• Will help Reducing operation incident
Conclusion

• Achieve QoS by implementing CD
• Strong conviction required
• people first, tools after
• small iterative steps