MERGE HELLS?

Feature Toggles to the rescue

By @leenasn at #rootconf

Familiar?

What went wrong?

Continuous Integration

"Continuous integration (CI) is the practice, in software engineering, of *merging all developer working copies to a shared mainline* several times a day."

Mainline Development

Incomplete Features?

Feature Toggles

```
def apply_dark_charges
  if SELECTIVE_DARK_CHARGES
    return DARK_CHARGES.
  end
  return NORMAL_CHARGES
end
```

Configuration Files

Available tools

- Rollout
- Togglz
- Qandidate-toggle
- Gutter

Release Toggles

- Deploy != Release
- Avoid Rollback Hells

Experimental Toggles

- **A/B Testing**
- **Canary Releasing**

Ops Toggles

- **Designing for failure**
- **Manage Circuit Breakers**

The Dark Side

```
REFERRALS = true
FASTER_SHIPPING_COSTS = true
MOBILE_PRODUCT_PAGES = true
PERFORMANCE_IMPORVEMENTS = true
MULTIPLE_PRODUCTS = true
FASTER_SHIPPING_NEW_DESIGN = true
SELECTIVE_DARK_CHARGES = true
ADD_ANOTHER_COLOR_OF_SAME_PRODUCT = false
NEW_DESIGN_FROM_HOME_PAGE = true
EW_GEOLOOKUP_API = true
ADD_ANOTHER_COLOR_OF_SAME_PRODUCT = false
```

$$ Question

How to manage it?

Expiry Date

Cathedral vs Bazaar model

Summary

Continuous Integration

- Follow Mainline Development
- First step towards Continuous Delivery/Deployment

Feature Toggles

- **Deploy != Release**
- **Quick Experiments**
- **Set Expiry date**

References

http://martinfowler.com/articles/feature-toggles.html

http://www.multunus.com/blog/2016/03/merge-hells-feature-toggles-rescue/

http://www.multunus.com/blog/2013/06/github-workflow-vs-mainline-development/

Leena S N

@leenasn / leena.sn@multunus.com