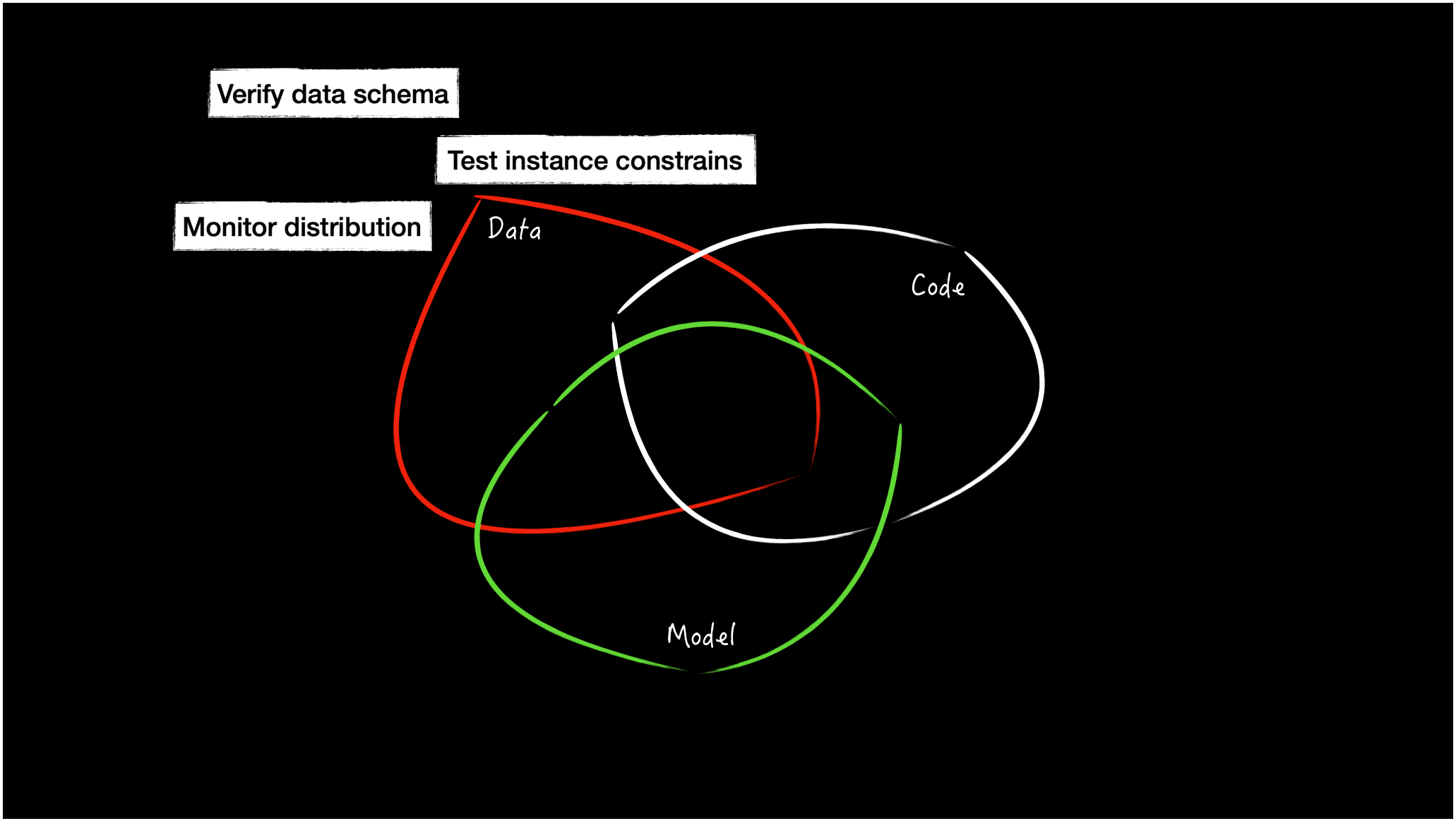
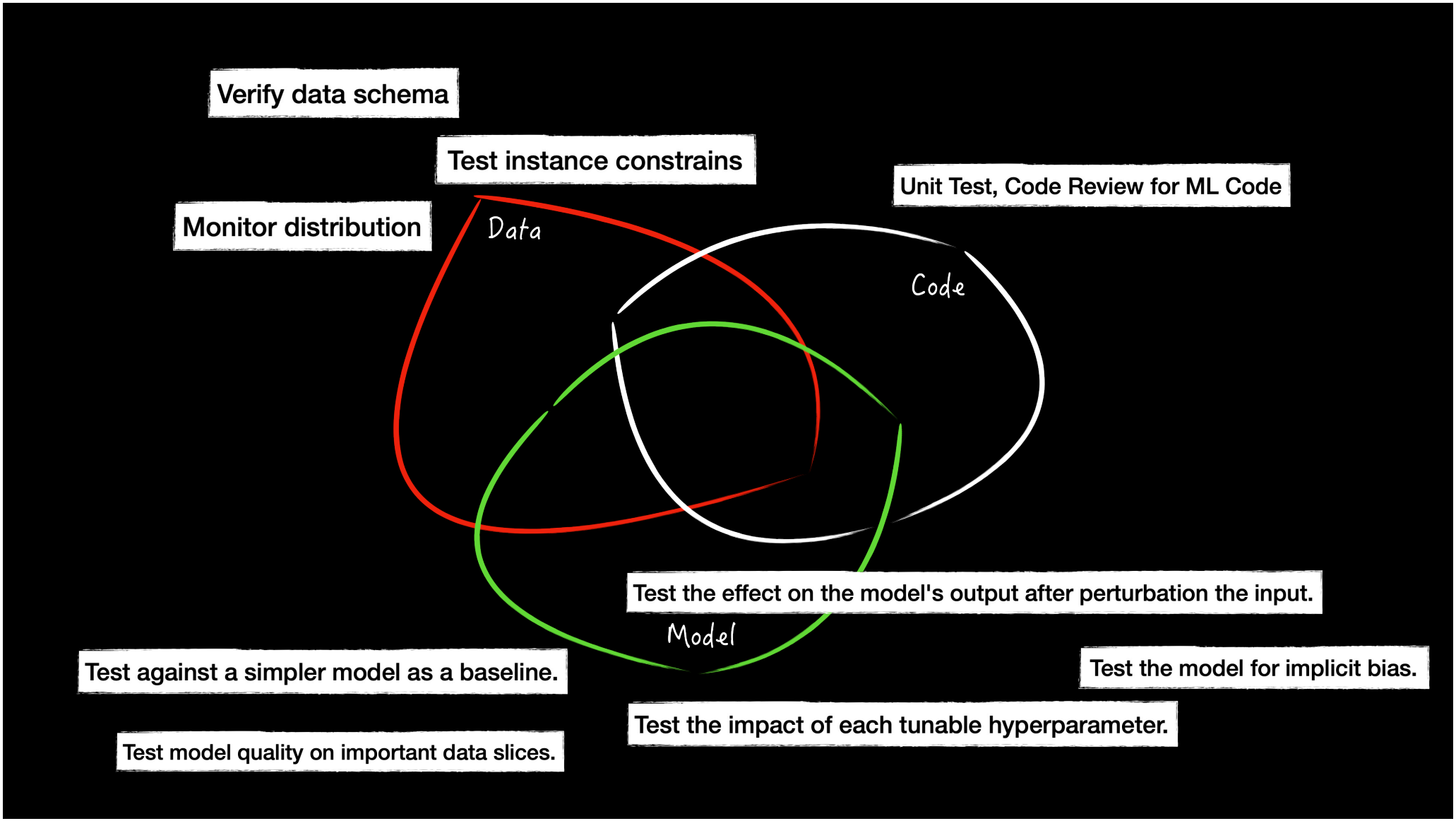
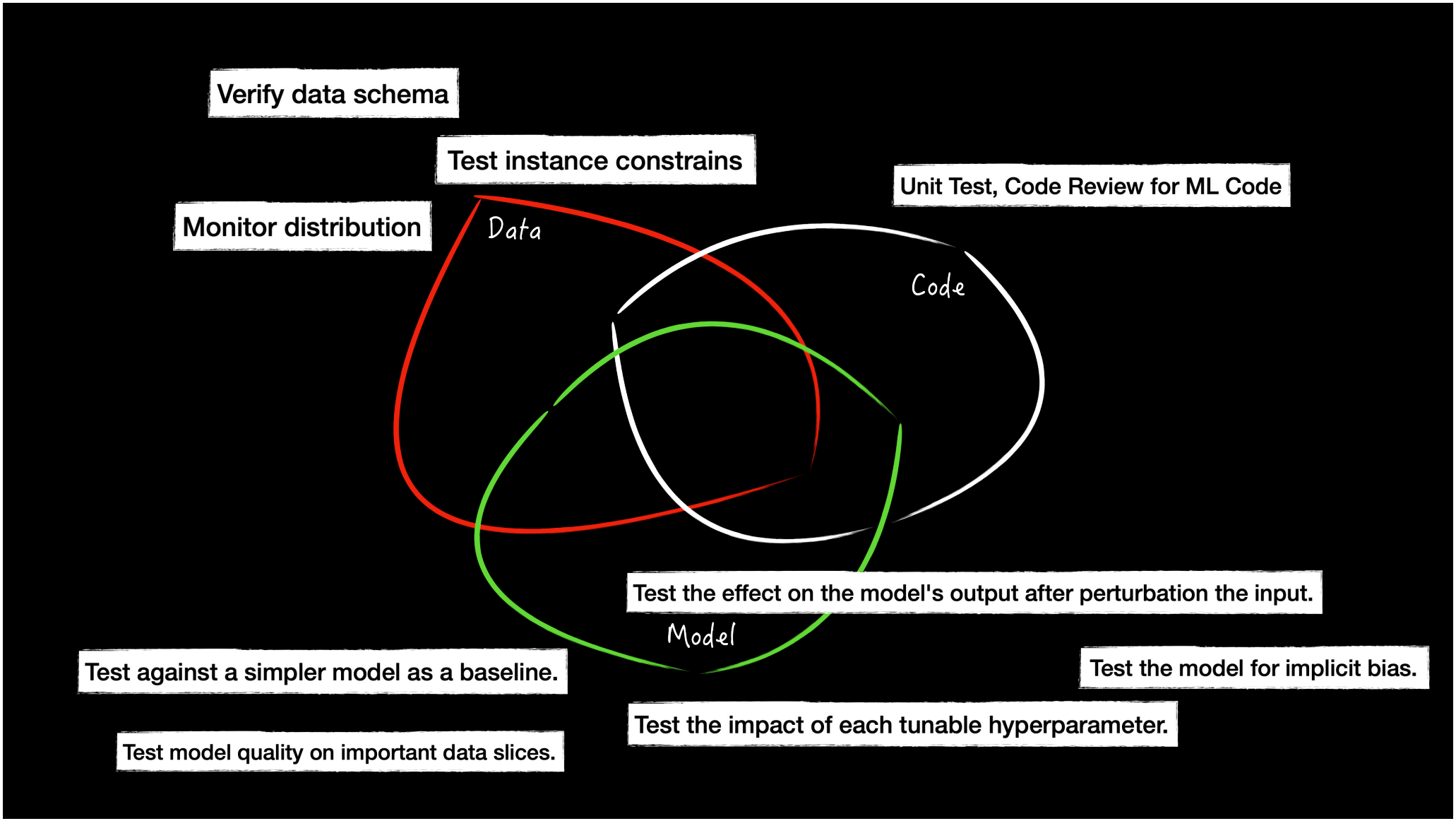
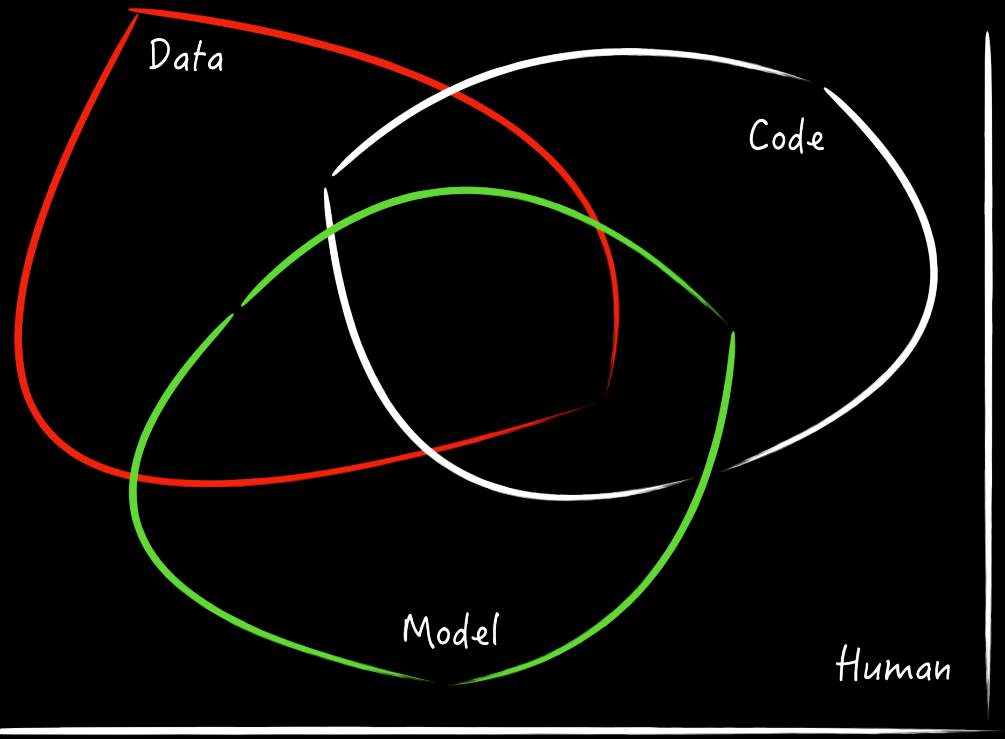
System Quality and Continuous Delivery

Jin Guo
SOCS McGill University


Verify data schema
Test instance constrains
Monitor distribution
Data
Code
Model

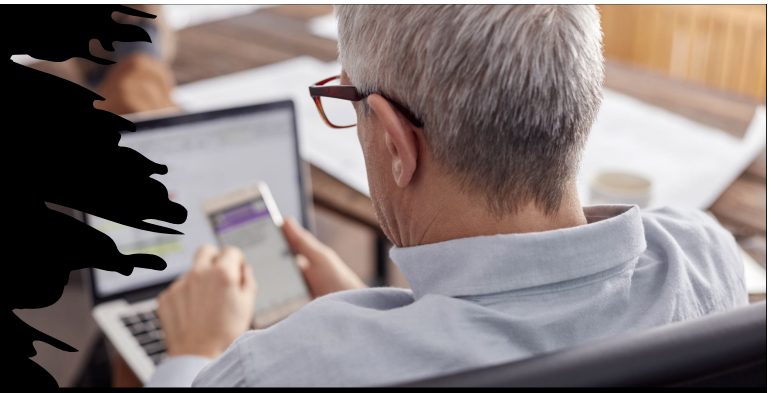

Verify data schema
Test instance constrains
Monitor distribution
Data
Unit Test, Code Review for ML Code
Code
Test the effect on the model's output after perturbation the input.
Model
Test against a simpler model as a baseline.
Test the model for implicit bias.
Test the impact of each tunable hyperparameter.
Test model quality on important data slices.


Verify data schema
Test instance constrains
Monitor distribution
Data
Unit Test, Code Review for ML Code
Code
Test the effect on the model's output after perturbation the input.
Model
Test against a simpler model as a baseline.
Test the model for implicit bias.
Test the impact of each tunable hyperparameter.
Test model quality on important data slices.



Test the relationship between offline proxy metrics and the actual online impact metrics.

Telemetry Design



Responsible for collecting data emitted from a system, about its state and behavior.

- Understand the impact on users

- Gather new training data

- Monitoring system works correctly

Understand Impact on Users

To determine if users are getting positive or negative outcomes and if the system is achieving its goals.

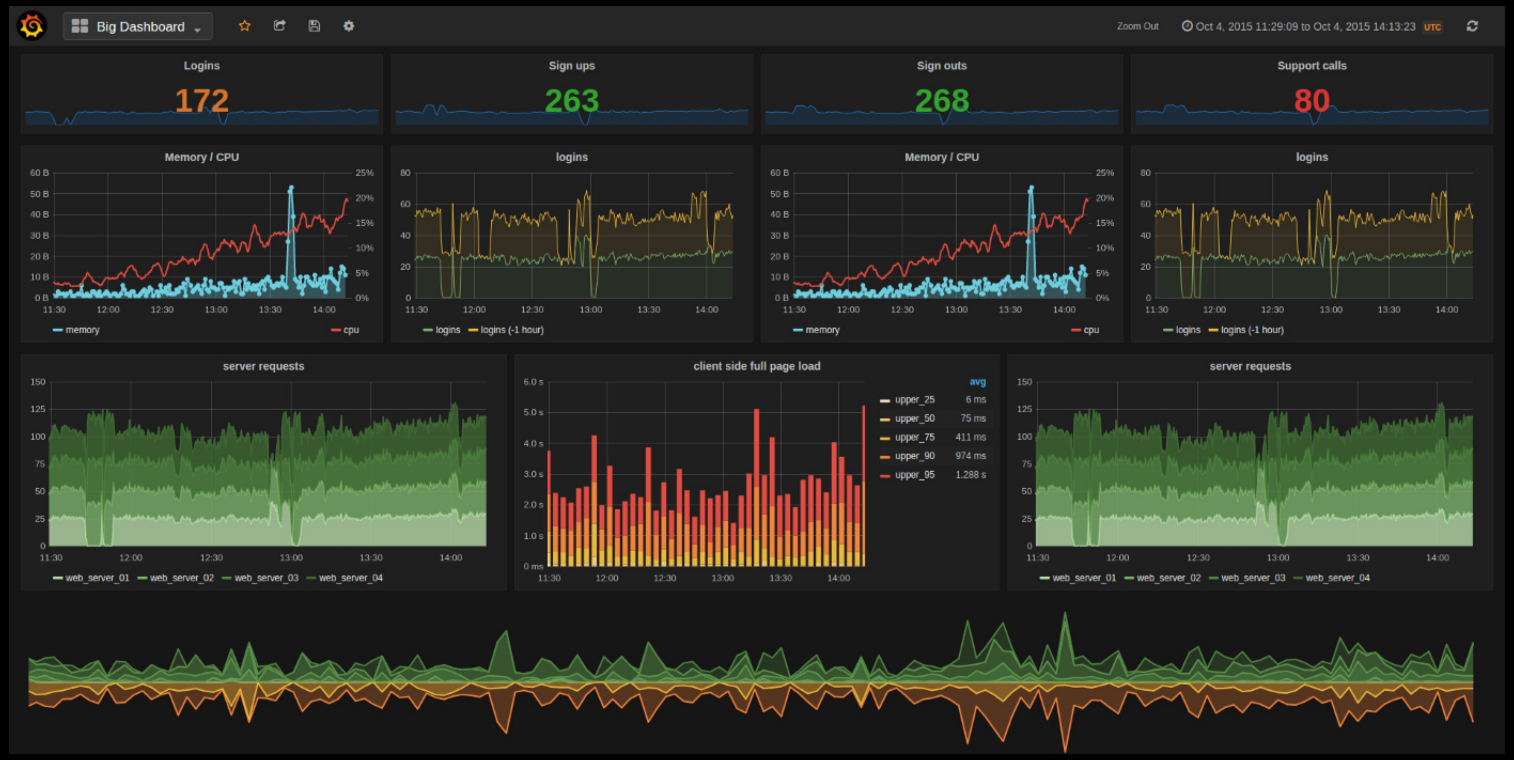
- ✓ Which experiences do users receive and how often do they receive them?
- ✓ What actions do users take in each experience?
- ✓ What experiences tend to drive users to look for help or to undo or revert their actions?
- ✓ What is the average time between users encountering a specific experience and leaving the application?
- ✓ Do users who interact more with the intelligent part of the system tend to be more or less engaged (or profitable) over time?

Activity

- Topic 1: Shopping app feature that detects the shoe brand from photos;
- Topic 2: Tagging uploaded photos with friends' names;
- Topic 3: Recommended personalized playlists;
- Topic 4: Code completion recommendation in IDE.

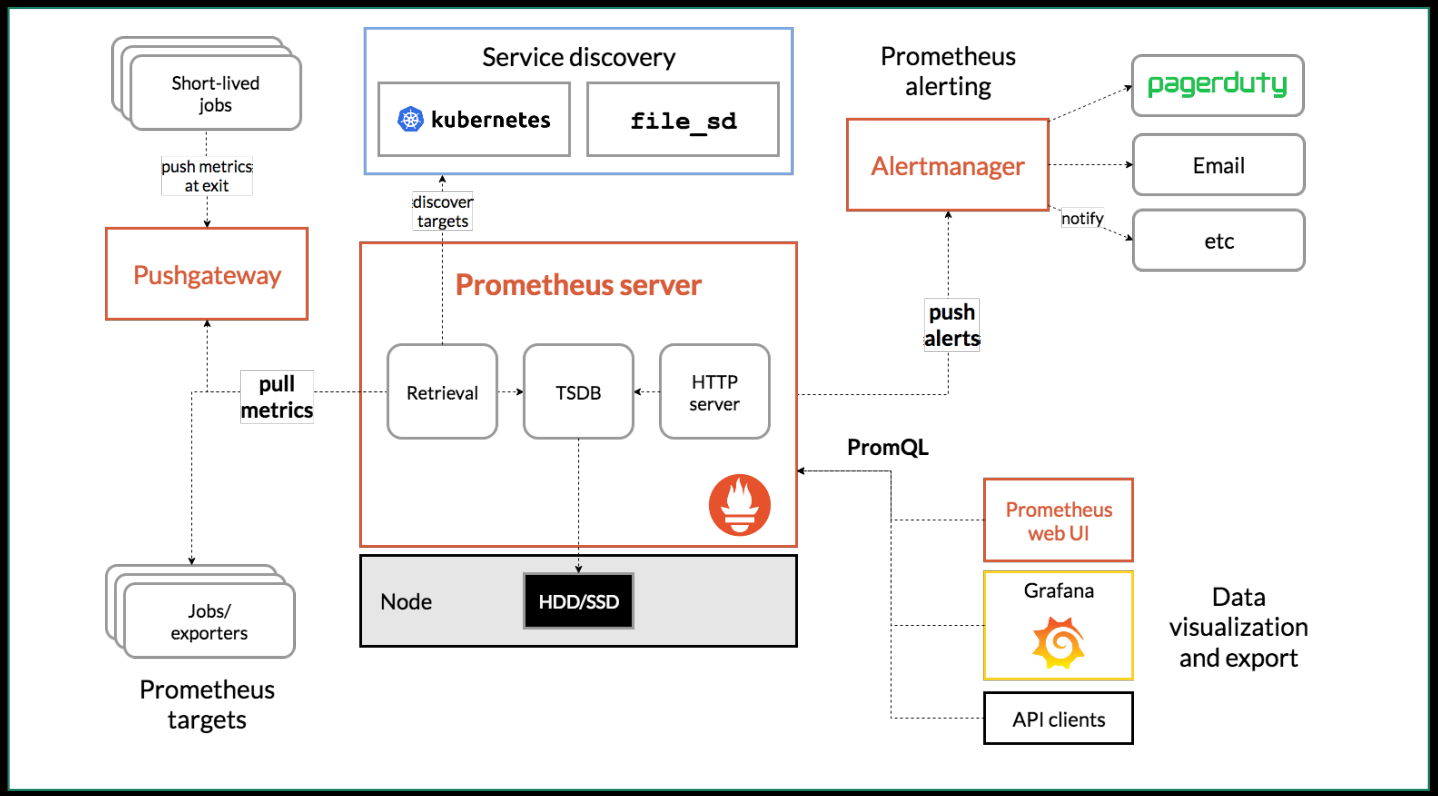
- What information should the telemetry system capture?
- How are you going to use the information to identify and debug potential problems?
- How costly is it to collect the data? How do you plan to manage the cost?
- Any challenges/risks for the your telemetry design?

Monitoring and alerting



https://medium.com/@pacroy/application-telemetry-with-prometheus-sap-blogs-c4b5b6239d28ke

Monitoring and alerting



https://prometheus.io/docs/introduction/overview/

More monitoring considerations

- Latency
- Scale
- Cost
- Automation
- Maintainability
- …

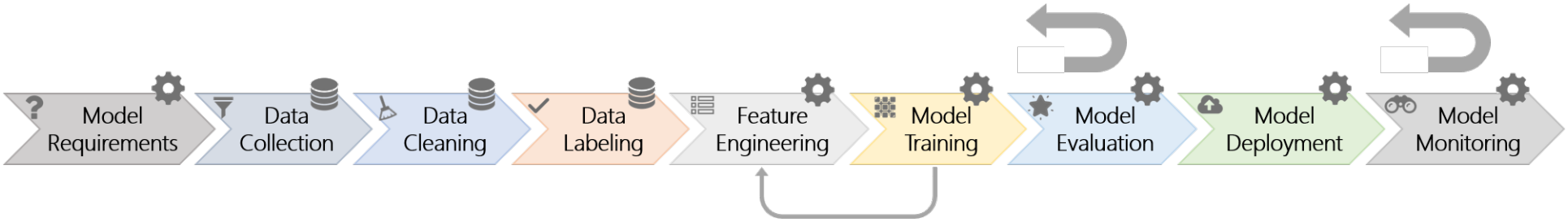


Fig. 1. The nine stages of the machine learning workflow. Some stages are data-oriented (e.g., collection, cleaning, and labeling) and others are model-oriented (e.g., model requirements, feature engineering, training, evaluation, deployment, and monitoring). There are many feedback loops in the workflow. The larger feedback arrows denote that model evaluation and monitoring may loop back to any of the previous stages. The smaller feedback arrow illustrates that model training may loop back to feature engineering (e.g., in representation learning).

Amershi, Saleema, Andrew Begel, Christian Bird, Robert DeLine, Harald Gall, Ece Kamar, Nachiappan Nagappan, Besmira Nushi, and Thomas Zimmermann. "Software engineering for machine learning: A case study." In *2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP)*, pp. 291-300. IEEE, 2019.

Serving ML Models

- Embedded model

  Treat the model artifact as a dependency that is built and packaged within the consuming application.

- Model deployed as a separate service

  The model is wrapped in a service that can be deployed independently of the consuming applications.

- Model published as data

  The model is treated and published independently; the consuming application choose the model at runtime as a data artifact

Updating Models

- Model iteration

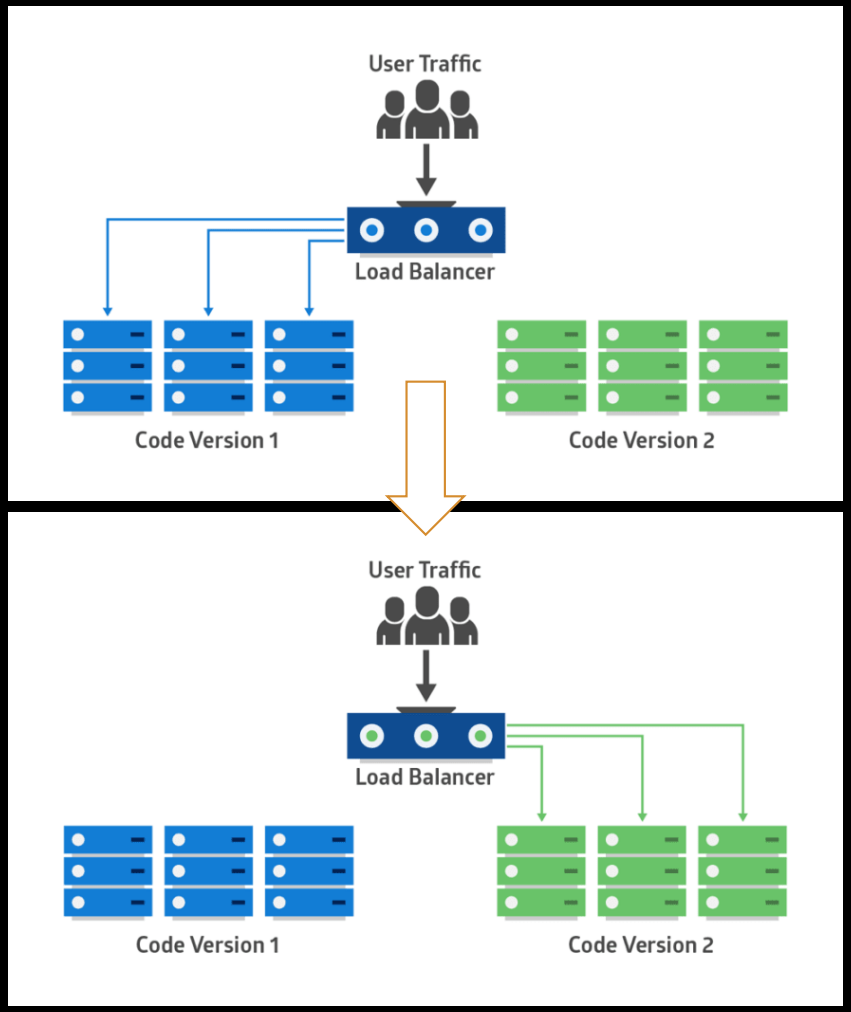
  A new feature is added to an existing model architecture, or the model architecture is changed.

- Data iteration

  The model architecture and features remain the same, but (continue) trained with new data

Deployment Options

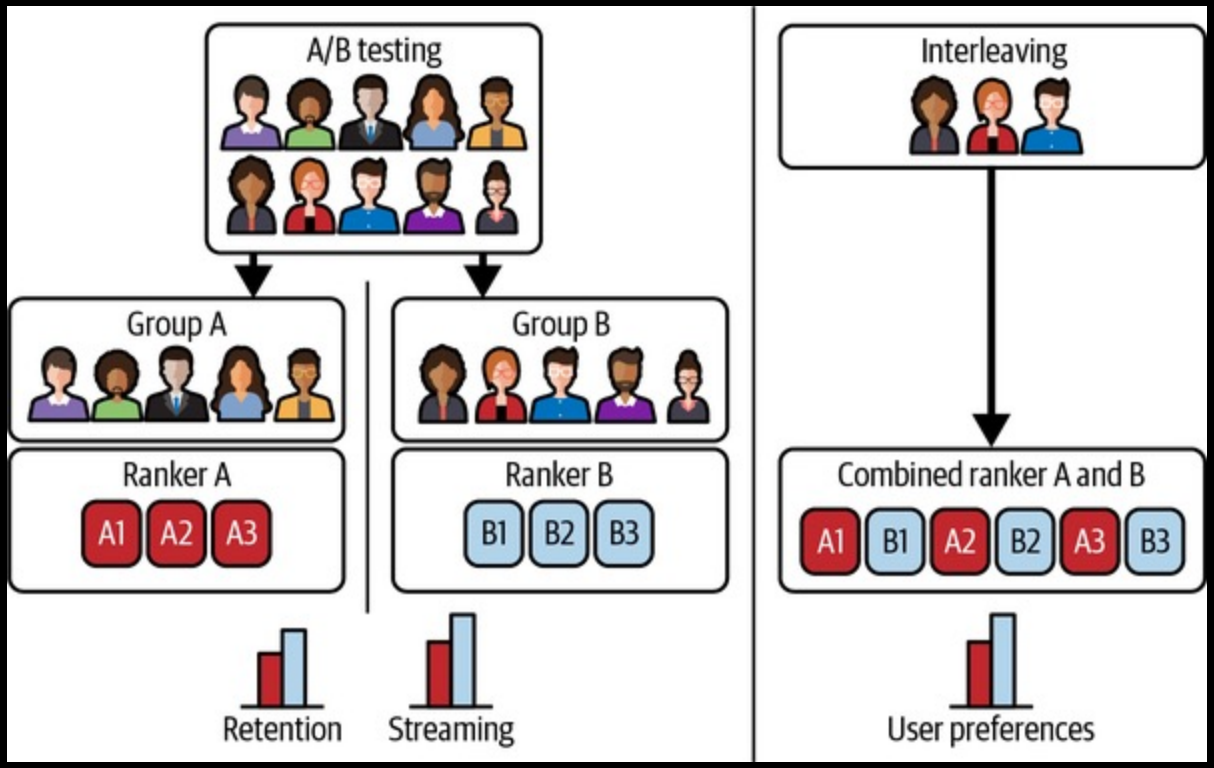
- Shadow models



https://dev.to/mostlyjason/intro-to-deployment-strategies-blue-green-canary-and-more-3a3

Deployment Options

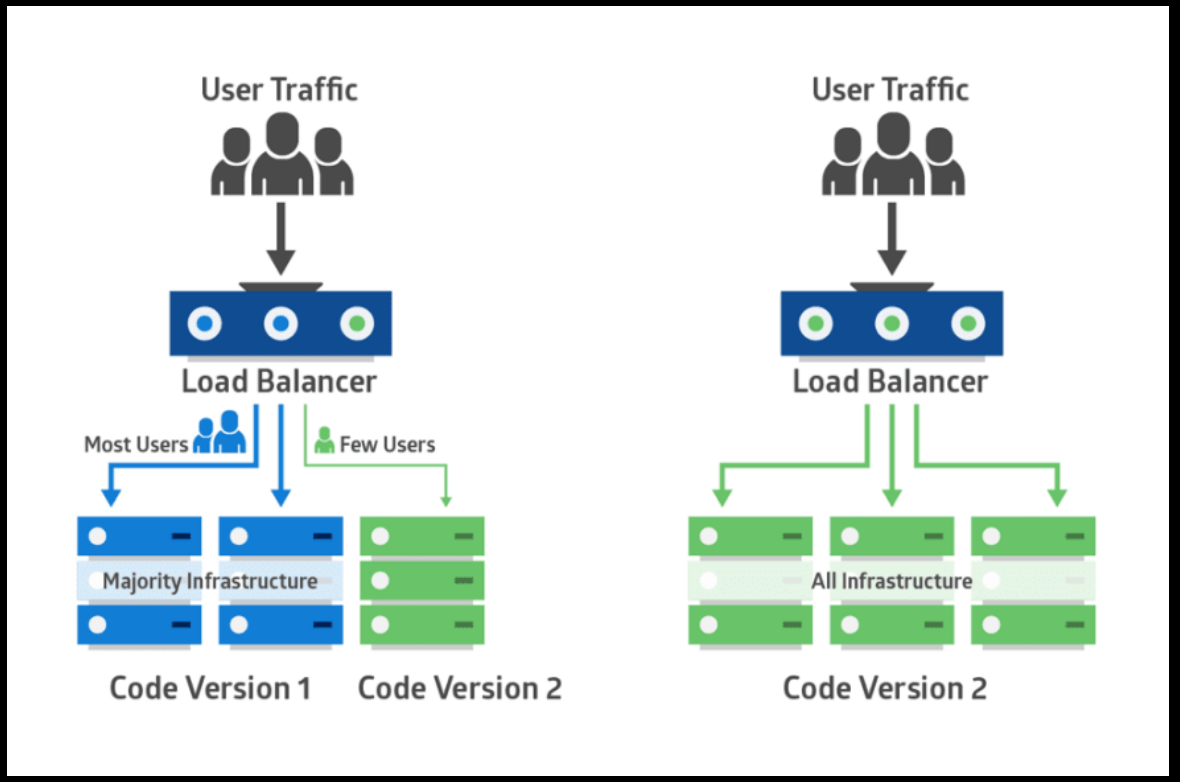
- Shadow models
- Competing models



Designing Machine Learning Systems: An Iterative Process for Production-Ready Applications, by Chip Huyen, O'Reilly Media

Deployment Options

- Shadow models
- Competing models
- Canary Release



https://dev.to/mostlyjason/intro-to-deployment-strategies-blue-green-canary-and-more-3a3

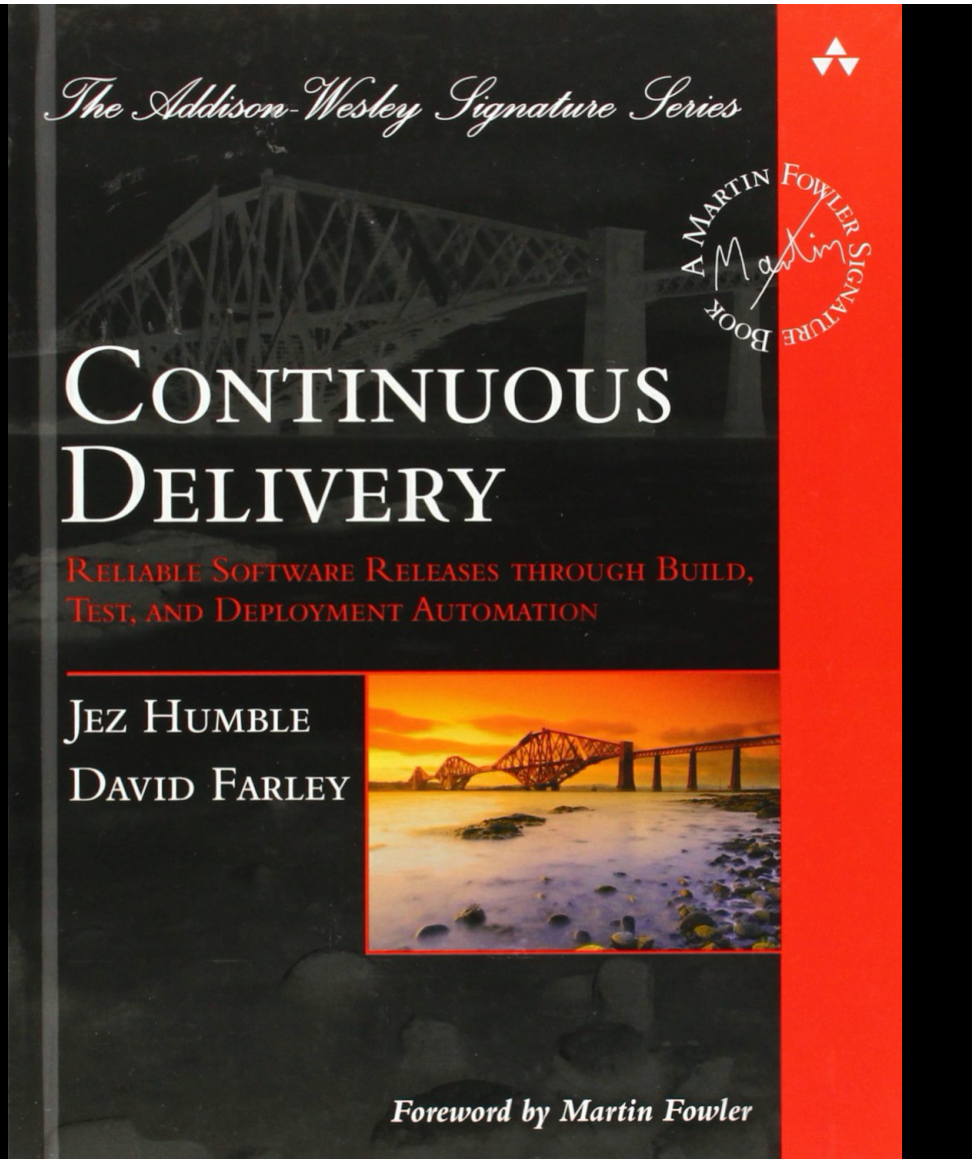
Deployment Options

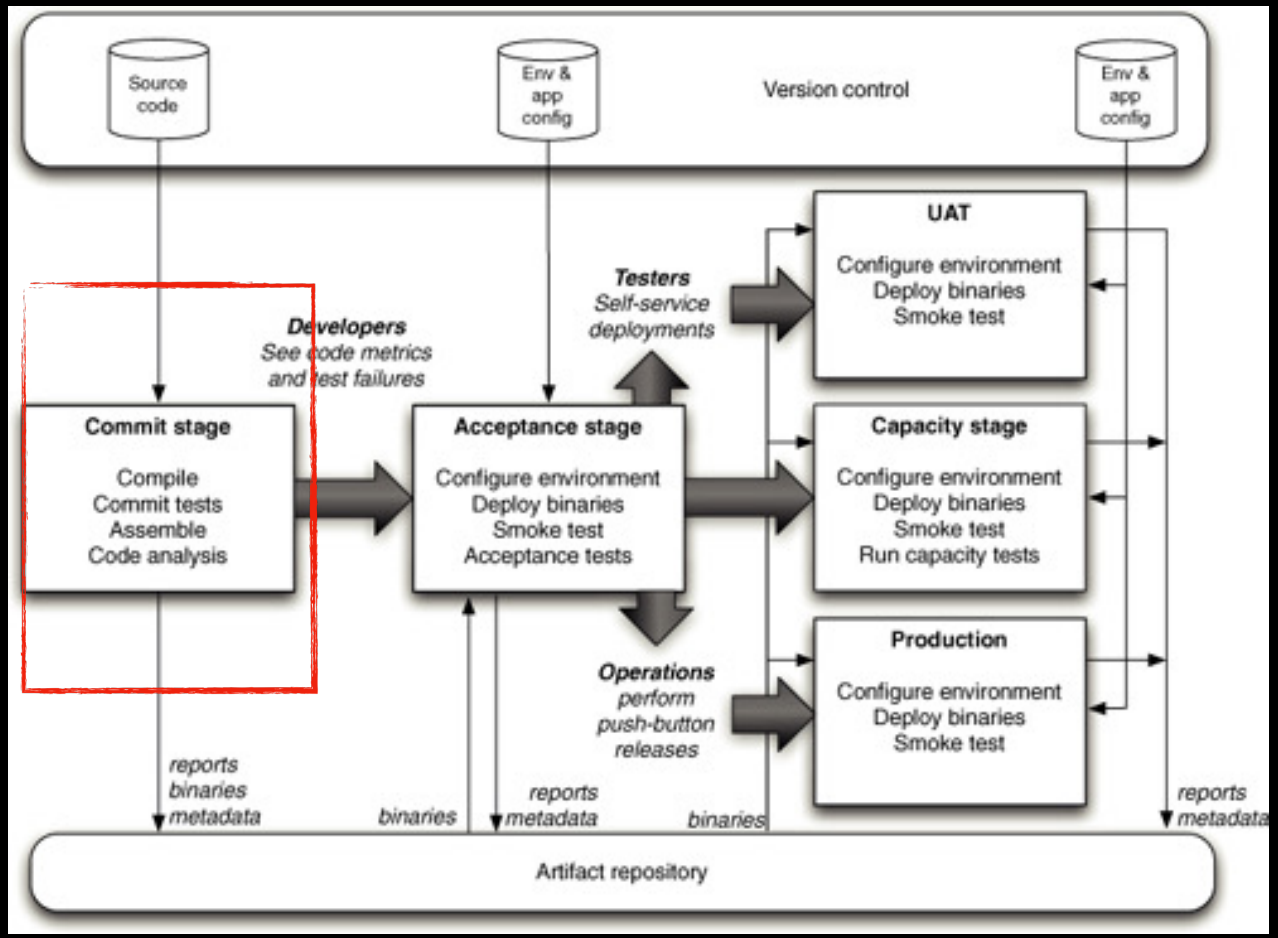
- Shadow models
- Competing models
- Canary Release
- Multiple models

Continuous Delivery

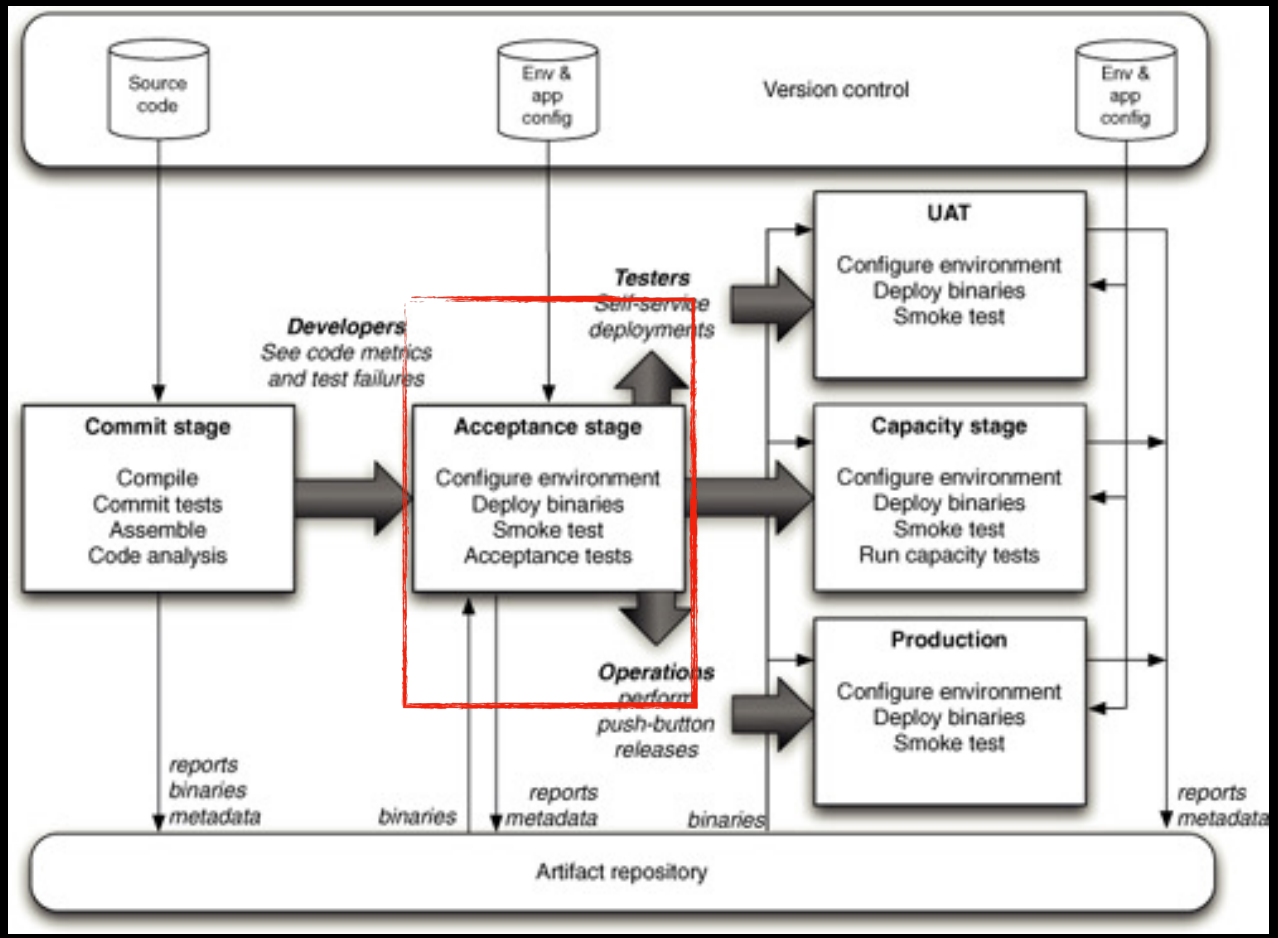
A close, collaborative working relationship between development and deployment teem

Extensive automation of all possible parts of the delivery process

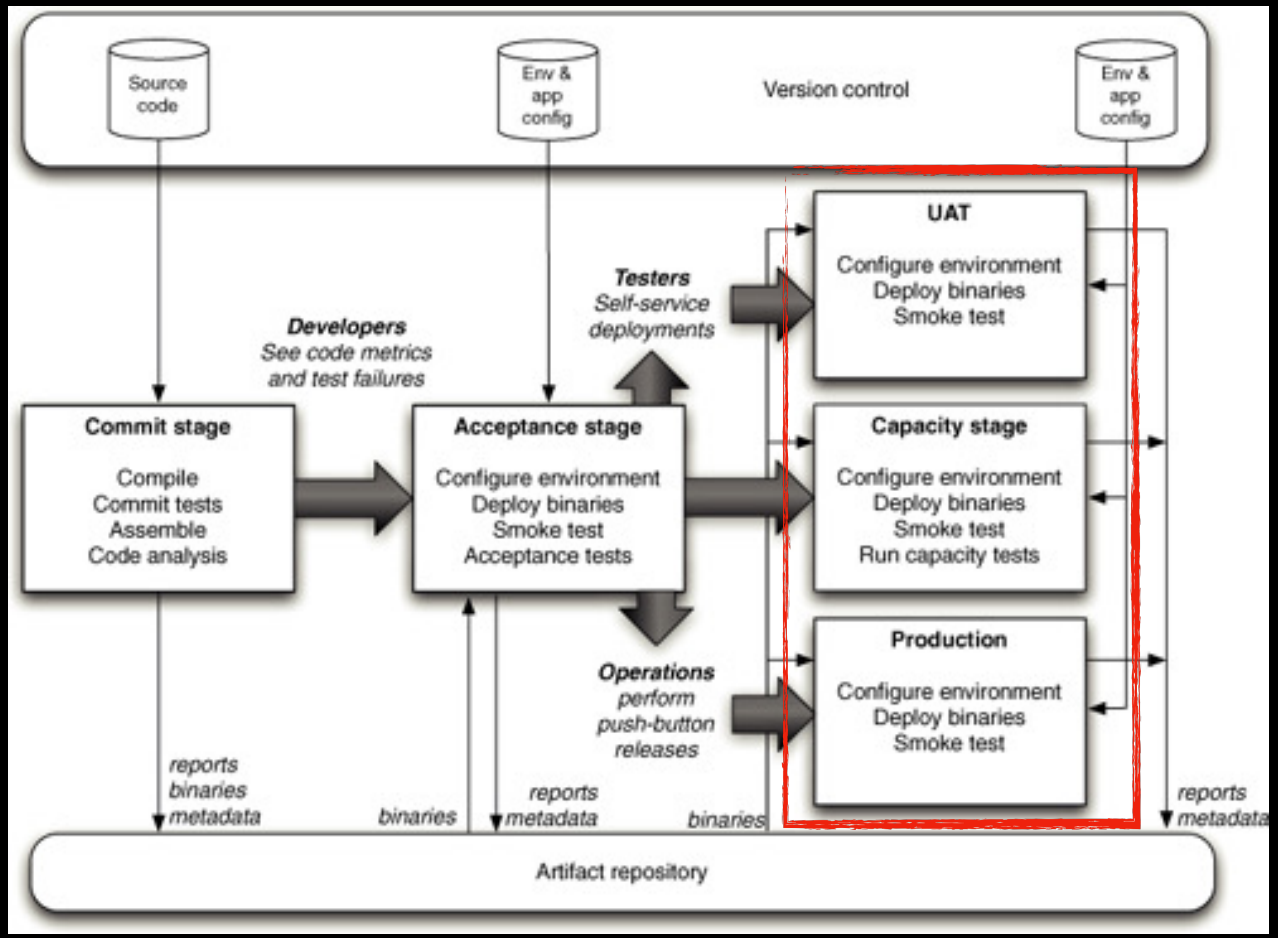




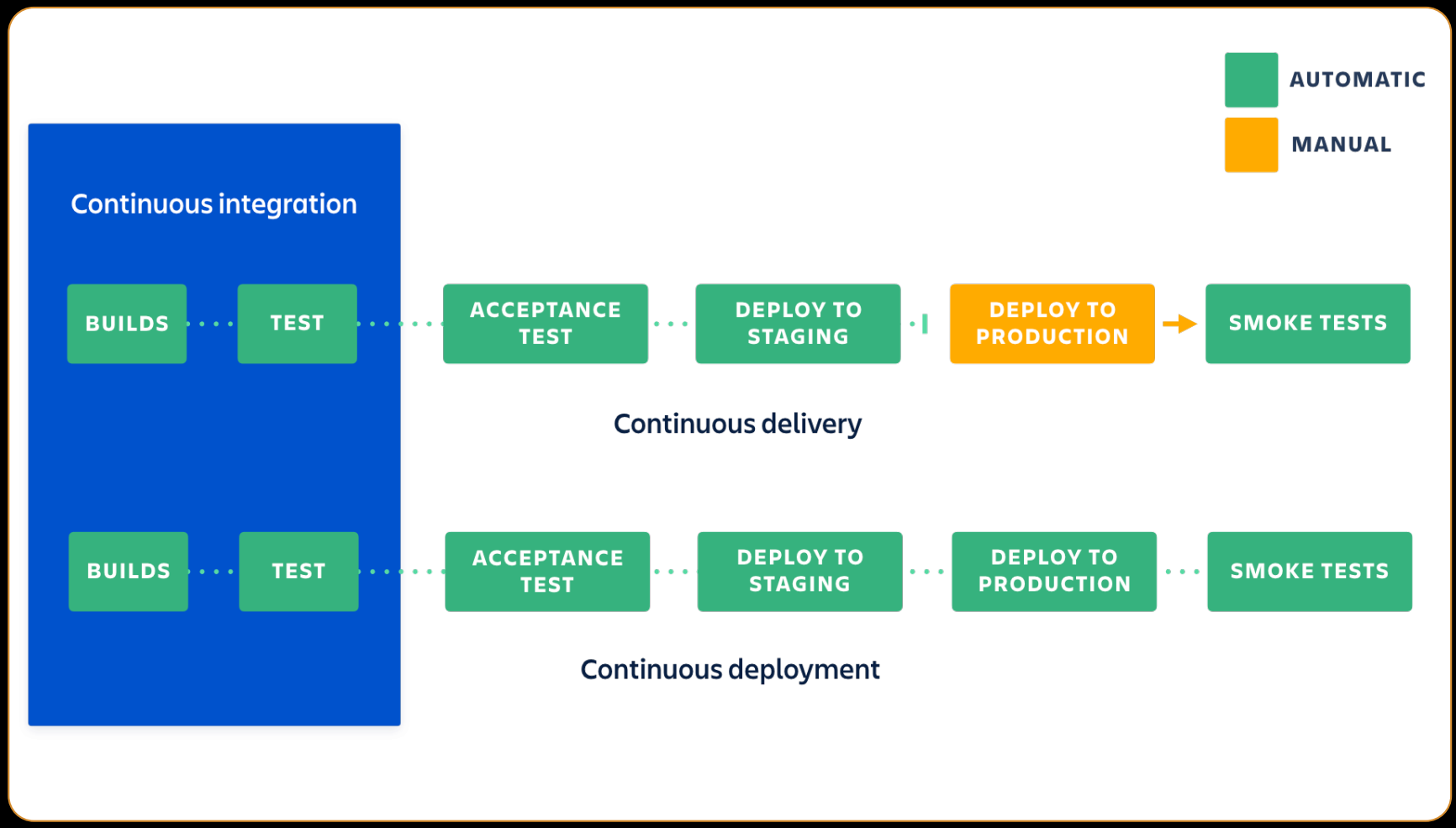
Continuous Delivery Reliable Software Releases through Build, Test, and Deployment Automation by Jez Humble and David Farley



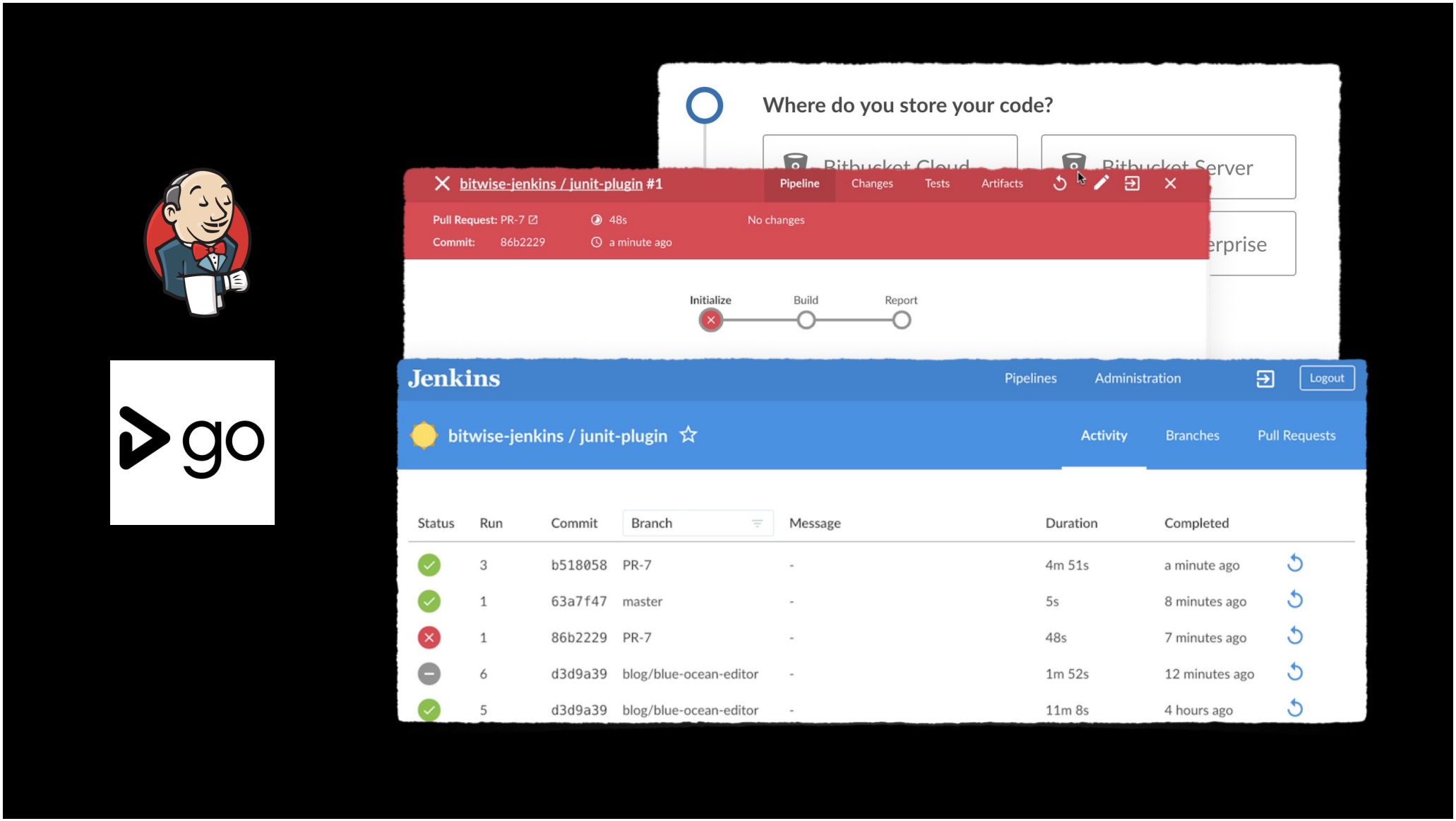
Continuous Delivery Reliable Software Releases through Build, Test, and Deployment Automation by Jez Humble and David Farley



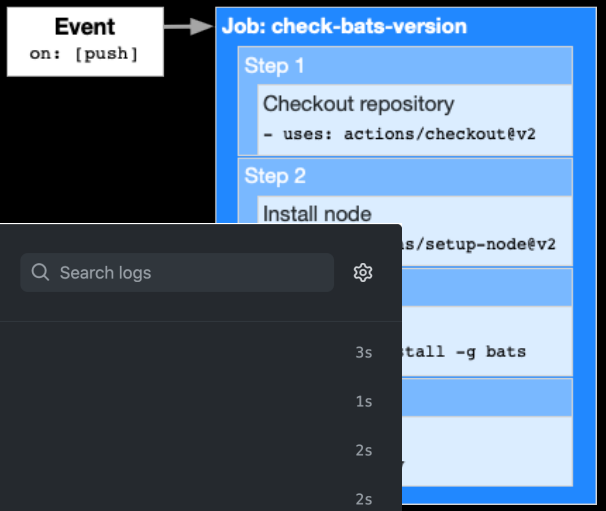
Continuous Delivery Reliable Software Releases through Build, Test, and Deployment Automation by Jez Humble and David Farley

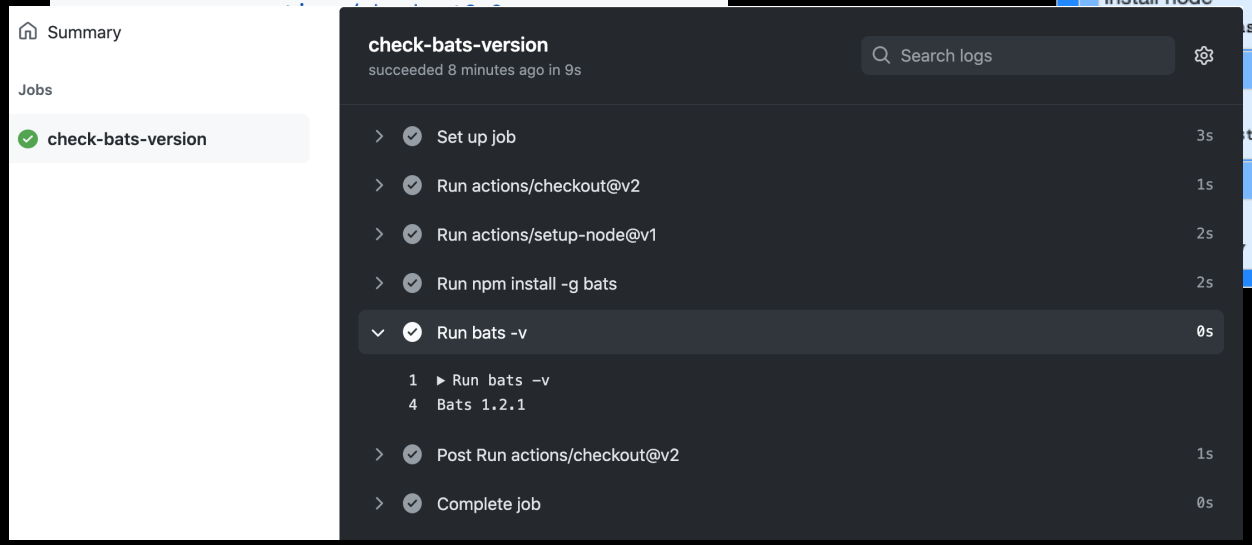


https://www.atlassian.com/continuous-delivery/principles/continuous-integration-vs-delivery-vs-deployment

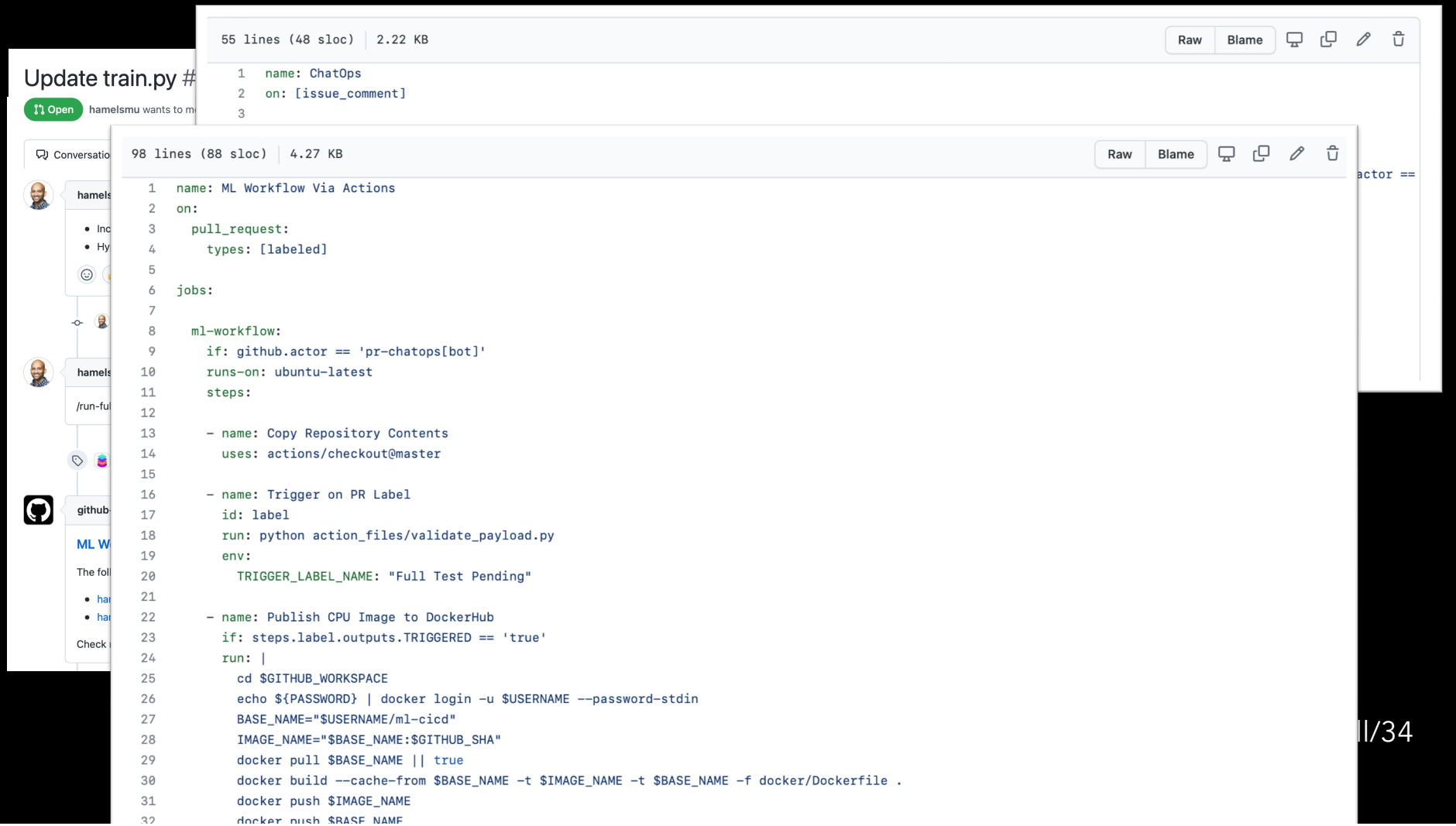

Where do you store your code?
Bitbucket Cloud
Bitbucket Server
erprise
bitwise-jenkins / junit-plugin #1
Pipeline
Changes
Tests
Artifacts
Pull Request: PR-7
48s
No changes
Commit: 86b2229
a minute ago
Initialize
Build
Report
Jenkins
Pipelines
Administration
Logout
bitwise-jenkins / junit-plugin
Activity
Branches
Pull Requests
Status
Run
Commit
Branch
Message
Duration
Completed
3 b518058 PR-7 - 4m 51s a minute ago
1 63a7f47 master - 5s 8 minutes ago
1 86b2229 PR-7 - 48s 7 minutes ago
6 d3d9a39 blog/blue-ocean-editor - 1m 52s 12 minutes ago
5 d3d9a39 blog/blue-ocean-editor - 11m 8s 4 hours ago

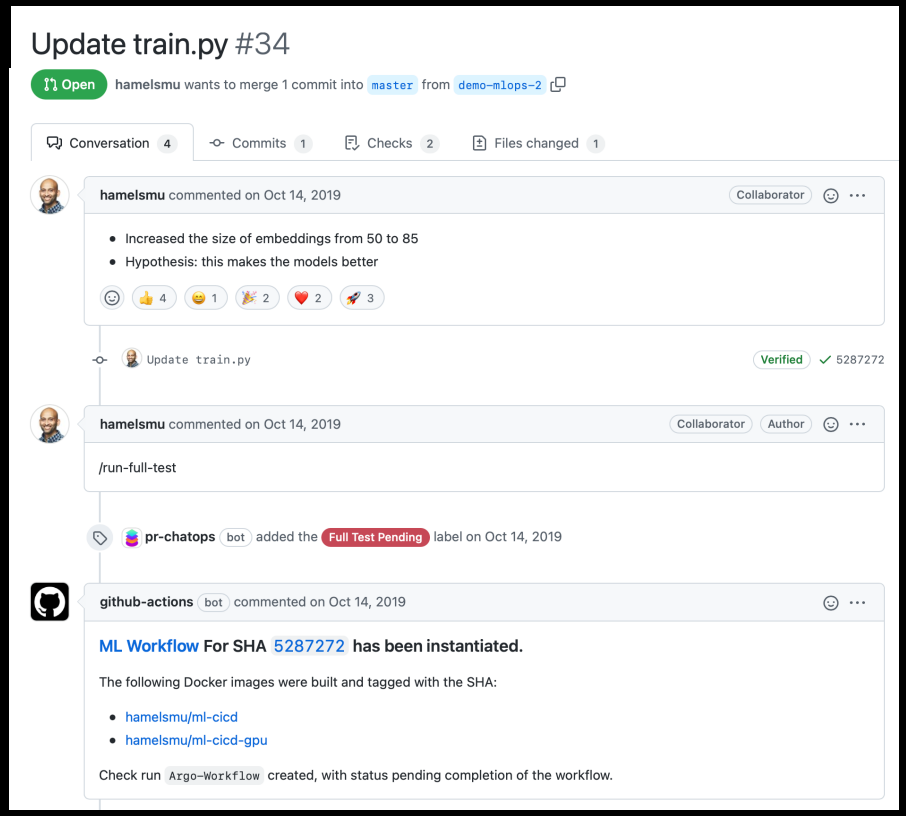
```yaml
name: learn-github-actions
on: [push]
jobs:
  check-bats-version:
    runs-on: ubuntu-latest
    steps:
```





https://docs.github.com/en/actions/learn-github-actions/understanding-github-actions#viewing-the-jobs-activity


Update train.py #
Open
hamelsmu wants to m
Conversatio
hamels
Inc
Hy
hamels
/run-ful
github
ML W
The foll
ha
ha
Check
55 lines (48 sloc) | 2.22 KB
Raw
Blame
1 name: ChatOps
2 on: [issue_comment]
3
98 lines (88 sloc) | 4.27 KB
Raw
Blame
actor ==
1 name: ML Workflow Via Actions
2 on:
3 pull_request:
4 types: [labeled]
5
6 jobs:
7
8 ml-workflow:
9 if: github.actor == 'pr-chatops[bot]'
10 runs-on: ubuntu-latest
11 steps:
12
13 - name: Copy Repository Contents
14 uses: actions/checkout@master
15
16 - name: Trigger on PR Label
17 id: label
18 run: python action_files/validate_payload.py
19 env:
20 TRIGGER_LABEL_NAME: "Full Test Pending"
21
22 - name: Publish CPU Image to DockerHub
23 if: steps.label.outputs.TRIGGERED == 'true'
24 run: |
25 cd $GITHUB_WORKSPACE
26 echo ${PASSWORD} | docker login -u $USERNAME --password-stdin
27 BASE_NAME="$USERNAME/ml-cicd"
28 IMAGE_NAME="$BASE_NAME:$GITHUB_SHA"
29 docker pull $BASE_NAME || true
30 docker build --cache-from $BASE_NAME -t $IMAGE_NAME -t $BASE_NAME -f docker/Dockerfile .
31 docker push $IMAGE_NAME
32 docker push $BASE_NAME
l/34



https://github.com/machine-learning-apps/actions-ml-cicd/pull/34
https://youtu.be/Ll50l3fsoYs

Building a Robust Dev/Op Pipeline

Observability – telemetry design

Automation

Robust to changes

ML Test Score

Breck, Eric, et al. "The ML test score: A rubric for ML production readiness and technical debt reduction." *2017 IEEE International Conference on Big Data (Big Data)*. IEEE, 2017.

1	Feature expectations are captured in a schema.
2	All features are beneficial.
3	No feature's cost is too much.
4	Features adhere to meta-level requirements.
5	The data pipeline has appropriate privacy controls.
6	New features can be added quickly.
7	All input feature code is tested.

Table I
BRIEF LISTING OF THE SEVEN DATA TESTS.

1	Model specs are reviewed and submitted.
2	Offline and online metrics correlate.
3	All hyperparameters have been tuned.
4	The impact of model staleness is known.
5	A simpler model is not better.
6	Model quality is sufficient on important data slices.
7	The model is tested for considerations of inclusion.

Table II
BRIEF LISTING OF THE SEVEN MODEL TESTS

1	Training is reproducible.
2	Model specs are unit tested.
3	The ML pipeline is Integration tested.
4	Model quality is validated before serving.
5	The model is debuggable.
6	Models are canaried before serving.
7	Serving models can be rolled back.

Table III
BRIEF LISTING OF THE ML INFRASTRUCTURE TESTS

1	Dependency changes result in notification.
2	Data invariants hold for inputs.
3	Training and serving are not skewed.
4	Models are not too stale.
5	Models are numerically stable.
6	Computing performance has not regressed.
7	Prediction quality has not regressed.

Table IV
BRIEF LISTING OF THE SEVEN MONITORING TESTS

ML Test Score

Breck, Eric, et al. "The ML test score: A rubric for ML production readiness and technical debt reduction." *2017 IEEE International Conference on Big Data (Big Data)*. IEEE, 2017.

Points	Description
0	More of a research project than a productionized system.
(0,1]	Not totally untested, but it is worth considering the possibility of serious holes in reliability.
(1,2]	There's been first pass at basic productionization, but additional investment may be needed.
(2,3]	Reasonably tested, but it's possible that more of those tests and procedures may be automated.
(3,5]	Strong levels of automated testing and monitoring, appropriate for mission-critical systems.
> 5	Exceptional levels of automated testing and monitoring.

Next Monday

Design for Human-AI Interaction