A photograph of the Golden Gate Bridge in San Francisco, partially obscured by thick fog. The bridge's red-orange structure is visible on the left, with the suspension cables and towers extending into the distance. The water below is dark and choppy, with white foam from waves visible at the bottom left.

Jak zmieniało się  
testowanie aż do  
ery (?) DevOps



#7

AiOO  
DEVOPS AND MORE

# Trasa

Duży obrazek

Testowanie jako process

Wdrożenie

Jaki jest problem

Gdzie jest klient

Wizja

Automatyzacja

Testowanie jako aktywność

Zespół

Pieniądze

Process





Po co nam tester (QA) w zespole?



- Osoba zmotywowana na jakość
- QA vs QC (testowanie)
- Krytycznie myślący **inżynier**
- Czarnowidz (maybe?)
  - 5 x WHY?



# Koncept





# Etapy rozwoju

Rola testera w zespole ewoluowała w czasie. Zbiegło się to z postępem technologiczny i hasłami jak: continuous integration, continuous delivery, infrastructure as a code, ... as code,

Klasyczne podejście

## Tester

W klasycznym podejściu tester musiał być w zespole.

DevOps Boom!

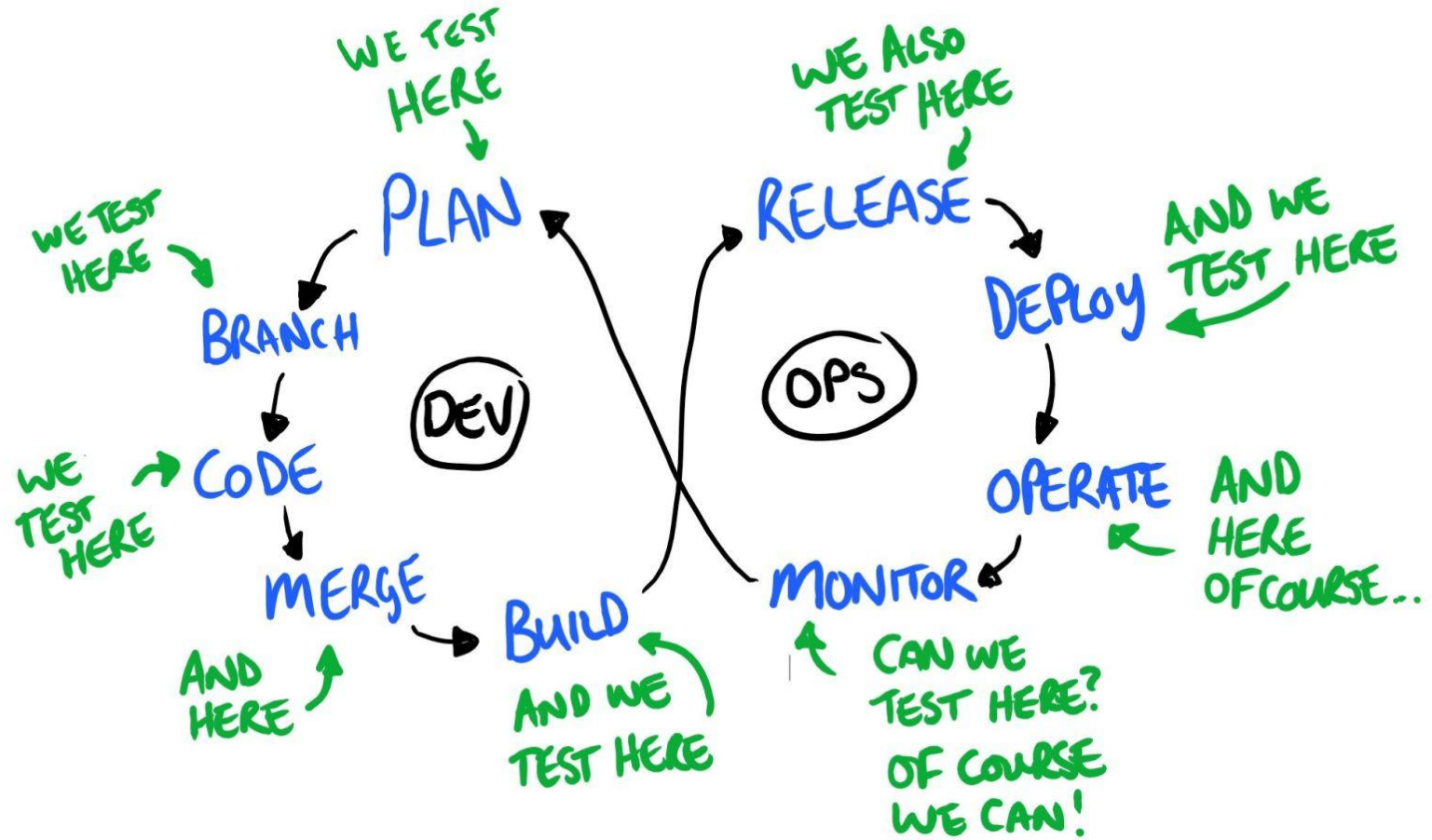
## User

Szybkie i ciągłe dostarczanie wyeliminowało rolę testera.

Cel

## QA

Jakość musi być częścią organizacji.







**QA ~= Quality Assistance**



*Mr/Ms Agile*

- Helping building solutions instead of breaking
- Concentrated on preventing bugs rather than finding bugs
- Testing as continuous activity throughout the delivery not being a phase, gate or state for the tasks
- Tester being a champion of “quality is a whole team responsibility” mindset
- Tester being guardian of business value

# Ops & QA

[#5](https://www.youtube.com/watch?v=Zl4logfSmC8)

State of DevOps Report:

<https://services.google.com/fh/files/misc/state-of-devops-2019.pdf>

Metryki:

1. Delivery lead time / Lead time for changes
2. Deployment frequency
3. Deployment failure rate
4. Mean time to restore



# Testowanie

6:28am





# Testowanie jako aktywność

## Klasyczne testowanie funkcjonalne

Testowanie czarnoskrzynkowe, białoskrzynkowe, statyczne, dynamiczne, weryfikacja czy walidacja a może wymagania

## A może powinno być bezpiecznie

Bezpieczeństwo jak aktywność czy proces, designed security, 3rd party dependency & vulnerability

?

...



## Funkcjonalne i niefunkcjonalne

Wszystko powinno działać szybko, ale ile to jest szybko?

## Użyteczność i dostępność

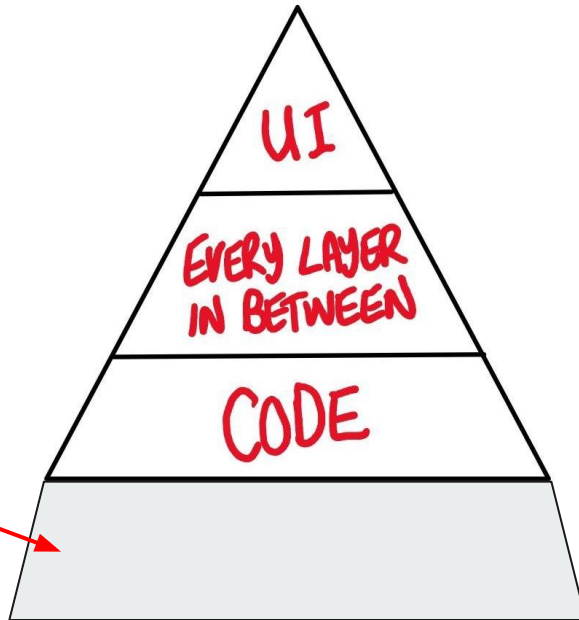
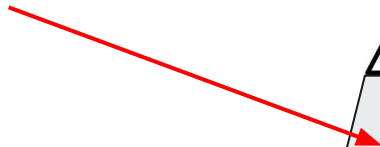
WCAG jak wymagania szczególnie dla public, projektowanie inkluzywne ale przede wszystkim nikt nie czyta dokumentacji. Interface powinien sam się tłumaczyć.





## IaaC/PaaC/...

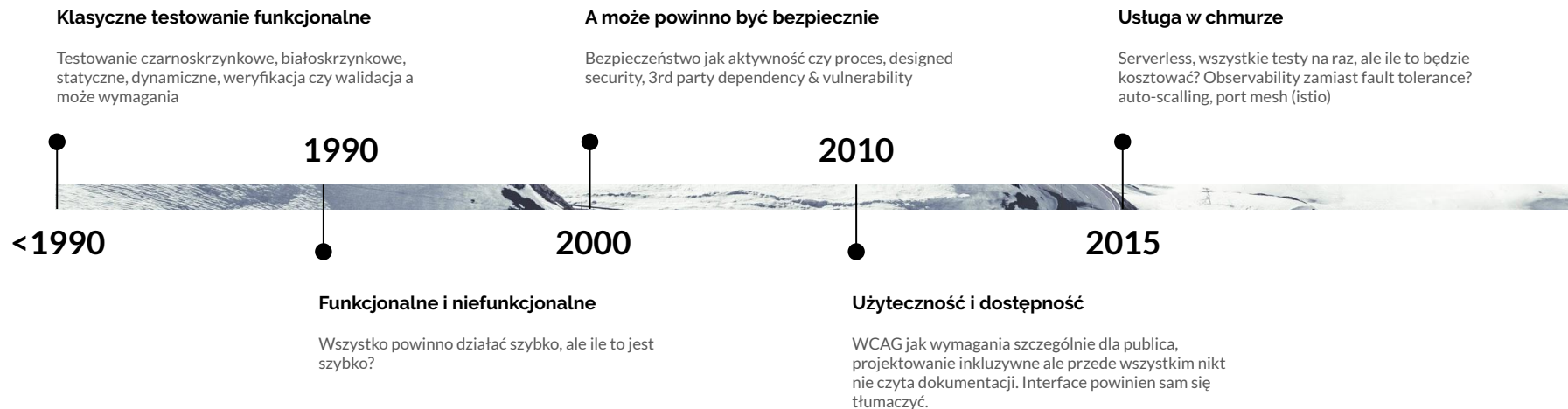
Coś nam doszło?



↑  
ASSERT  
EXPECTATIONS  
& INVESTIGATE  
RISKS AT  
ALL LAYERS...  
↓



# Testowanie jako aktywność



# Konteneryzacja





# Automatyzacja testów



0  
1

## Lepsze zarządzanie narzędziami

Wersjonowanie również środowiska i zależności do testów.

## Zrównoleglenie

Lambda w testach, orkiestracja kontenerów...



0  
2



0  
3

## Testy jako produkt

Jesteśmy coraz bliżej dostarczania testów razem z produktem.



```
FROM csf-docker-delivered.repo.lab.pl.alcatel-lucent.com/os_base/centos-micro:7.6-20190516 AS base

ARG KUBECTL_VERSION=v1.10.3
ENV HOME_DIR=/CSF-PAATEST
ENV PYTHONIOENCODING=utf-8

ARG package_version=">=21.0rc"
ENV PACKAGE_VERSION=${package_version}

COPY files/pip/pip.conf /etc/

RUN echo "Downloading ${FILENAME}" && \
    curl -LO https://repo.lab.pl.alcatel-lucent.com/storage-googleapis-com-kubernetes-client/${FILENAME} && \
    chmod +x ./kubectl && \
    mv ./kubectl /usr/local/bin/kubectl

RUN yum install -y python36 wget python36-crypto sudo docker-client shadow-utils tar && \
    yum clean all && \
    rm -rf /var/cache/yum

FROM base AS build

RUN yum install -y python36-pip gcc python36-devel && yum clean all && rm -rf /var/cache/yum

RUN pip3 --no-cache-dir install --upgrade pip pip-tools setuptools virtualenv

RUN virtualenv /${PACKAGE_NAME} && \
    . /${PACKAGE_NAME}/bin/activate && \
    pip3 --no-cache-dir install ${PIP_FLAGS} "${PACKAGE_NAME}${PACKAGE_VERSION}"
```



FROM base

RUN useradd cloud-user -c "ComPaaS Tests" -d \${HOME\_DIR} -M --uid 1000

RUN echo '%cloud-user ALL=(ALL) NOPASSWD:ALL' >> /etc/sudoers

COPY --from=build --chown=cloud-user /\${PACKAGE\_NAME} /\${PACKAGE\_NAME}

COPY files/ve\_run /usr/local/bin/ve\_run

COPY files/.bashrc \${HOME\_DIR}/.bashrc

WORKDIR \${HOME\_DIR}

RUN chown -R 1000:1000 \${HOME\_DIR}

USER cloud-user

RUN mkdir \${HOME\_DIR}/etc && ln -s \${HOME\_DIR}/etc/ /\${PACKAGE\_NAME}/lib/python3.6/site-packages/paas\_test\_sdk/

ENTRYPOINT ["/usr/local/bin/ve\_run"]



od testów do wdrożenia:

- Nose/PyTest/UnitTest/....
- Python:
  - subprocess
  - assertMatch

od wdrożenia do testów:

- Sensu
- Terratest: local-exec (provisioner)
- Chef eg. Inspect
- Ansible:
  - lineinfile



Tester lubi DevOps



Używa narzędzi i  
danych



Tester to przyjaciel  
każdego DevOps  
inżyniera