Sharing Stories and Learning Together



## Mark Waite Tales of Testing in Production

### Mistakes, compromises, alternatives, ... and some successes Production is complicated



### Data is different in production

- Distributed differently
- More diverse
- Contains more duplicates
- Has wider range of values
- More likely to contain surprising combinations



Icon made by <u>Smashicons</u> from <u>www.flaticon.com</u>



### Traffic is different in production

- Larger volume
- Has higher peaks
- More dramatic changes



Icon made by phatplus from www.flaticon.com



### Users are different in production

- More diversely named
- Wider range of experiences
- More likely to take surprising paths



Icon made by surang from www.flaticon.com



### The Jenkins story - surprises in production

- Jenkins UI on Internet Explorer 11
- A 20 GB git repo
- An unreliable link to the source code repository
- A Jenkins job with a Cyrillic name
- A Jenkins agent that disconnects randomly
- A Jenkins agent on Raspberry Pi
- A Jenkins agent on s390x mainframe





### Mistakes, compromises, alternatives, ... and some successes Progressive risk reduction



### **Prioritizing our tests**

Choosing our tests by risk, by value, by ...

- Likelihood of use
- Threat of misuse
- Criticality of function
- Cost of failure
- Cost of evaluation



Icon made by Freepik from www.flaticon.com



### Interactive exploration

Use human creativity well

- Wide angle view see as users do
- Subtle comparisons ignore many differences
- Rapidly assess look and feel
- Quickly assess suitability to task
- Detect unexpected side effects
- Abandon low value investigations promptly



Icon made by Freepik from www.flaticon.com



### Automation assisted exploration

Use automation well

- Narrow focus
- Precise comparisons detect many differences
- Insensitive to look and feel
- Confirm detailed results of task
- Unexpected side effects harder to detect
- Potential to scale significantly



Icon made by Freepik from www.flaticon.com



### The Jenkins story - Git plugin interactive and automated

- 2008 2010 Early adoption
- 2010 2013 Increasing use
- 2013 2017 Refactoring
- 2017 now Pipeline and beyond





### INSTALLS AND TESTS IN EARLY ADOPTION

- User needs are more important than tests
- Discover user needs by delivering







### INSTALLS AND TESTS IN INCREASING USE

- User needs are more important than tests
- Discover user needs by delivering



#### • Test interactively for fastest delivery

• Feature and dependency growth is increasing the cost of testing



### INSTALLS AND TESTS IN REFACTORING

- User needs are often failed by regressions
- Discover user needs by delivering



#### Test automation for fastest delivery

- Interactive testing still finds most bugs
- Feature and dependency growth is increasing the cost of testing



Jenkins Git Plugin lines of code vs. test LOC

### INSTALLS AND TESTS IN PIPELINE AND BEYOND

- Regressions are less frequent
- Discover user needs by delivering



- Interactive testing still finds most bugs
- Feature and dependency growth is increasing the cost of testing



Jenkins Git Plugin lines of code vs. test LOC



### Mistakes, compromises, alternatives, ... and some successes Helpful tools



### System configuration accelerators

- Docker
- Kubernetes
- Ansible
- Puppet
- Chef
- Jails (FreeBSD, Solaris)



### ANSIBLE



### **Application configuration accelerators**

- Extension languages (groovy, VBasic, etc.)
- File content recorders (git, svn, etc.)
- File system recorders (ZFS, btrfs, etc.)
- Database content recorders



Icon made by phatplus from www.flaticon.com



### The Jenkins story - git plugin & configuration as code

Testing the git plugin - motivation

- 260 000 active installations
- Business critical for many users
- 500+ opens bugs and enhancements
- Insufficient automated tests
- No performance or scalability automation





### The Jenkins story - git plugin & configuration as code

Testing the git plugin - combinations & complications

- 2 git implementations (CLI git & JGit)
- 6+ operating systems (Windows, Linux, zOS, ...)
- 10+ CLI git versions (1.8 ... 2.28)
- 5+ OpenSSH versions
- 2 JDK's (Java 8, Java 11)





### **My Priorities**

- Don't make it worse
- Help users with their issues
- Explore with a real world setup





### My solution - configuration as code with real world setup

- Started small, improved incrementally
- Docker and configuration as code for repeatability
- Intentional variation for exploration
  - CentOS, Debian, Ubuntu, FreeBSD, OpenBSD, Windows
  - o amd64, ARM64, ARM32, s390x, ppc64le
  - Local, AWS, GCP
- Jenkins jobs to test for bugs and fixes
  - Branch per bug in the jenkins-bugs repository
  - Multibranch pipeline rapidly checks many bugs in parallel
  - 4000+ jobs

by CloudBees

**DEVOPS**WORL

- https://github.com/MarkEWaite/docker-lfs
- https://github.com/MarkEWaite/jenkins-bugs



# Tales of Testing in Production

DEVOPS

by CloudBees

Sharing stories and learning together Mistakes, observations, and ooo