# Learning from failures

Kripa Krishnan
Technical Program Director
Sep.3.2014

Google

# Topics

- DiRT: Disaster simulation exercise at Google
- What we learned
- Applying these lessons

Google

# Introducing DiRT

- DiRT
  - Annual disaster recovery & testing exercise
  - 8 years since inception
  - Multi-day exercise triggering (controlled) failures in systems and process

# Introducing DiRT

- DiRT
  - Annual disaster recovery & testing exercise
  - 8 years since inception
  - Multi-day exercise triggering (controlled) failures in systems and process

- Premise
  - 30-day incapacitation of headquarters following a disaster
  - Other offices and facilities may be affected

# Introducing DiRT

- DiRT
  - Annual disaster recovery & testing exercise
  - 8 years since inception
  - Multi-day exercise triggering (controlled) failures in systems and process
- Premise
  - 30-day incapacitation of headquarters following a disaster
  - Other offices and facilities may be affected
- When
  - "Big disaster": Annually for 3-5 days
  - Continuous testing: Year-round

# Introducing DiRT

- DiRT
  - Annual disaster recovery & testing exercise
  - 8 years since inception
  - Multi-day exercise triggering (controlled) failures in systems and process
- Premise
  - 30-day incapacitation of headquarters following a disaster
  - Other offices and facilities may be affected
- When
  - "Big disaster": Annually for 3-5 days
  - Continuous testing: Year-round
- Who
  - 100s of engineers (Site Reliability, Network, Hardware, Software, Infrastructure, Security, Facilities)
  - Business units (Human Resources, Finance, Safety, Crisis response etc.)

# DiRT - Structure

- First 24 hours
  - Crisis response
  - Operations failover
- Rest of DiRT
  - Series of injected failures
  - 100s of smaller tests run by individual teams
- Team
  - Command Center - 30-40 'insiders'
  - 'Storyteller' narrates the test
  - 100s of teams respond to the tests

# DiRT - Structure

- First 24 hours
  - Crisis response
  - Operations failover
- Rest of DiRT
  - Series of injected failures
  - 100s of smaller tests run by individual teams
- Team
  - Command Center - 30-40 'insiders'
  - 'Storyteller' narrates the test
  - 100s of teams respond to the tests
  - 100s of teams write post-mortems for each of the tests
  - 100s of teams fix issues found in post-mortems

Google

# Rolling out a disaster (Example 1/4 - first 6 hours)

- Research and Test
  - Scenario: Bay area earthquake, later an aftershock
  - Story: Meteor hit the bay area. Zombies emerge from ground.
  - Structural damage, flooding communications hampered

- Response
  - Safety (Evacuations?)
  - People 'Finder' and ‘Alerter’
  - Incident management, team and operational failovers
  - Communicate Communicate

- Did we learn anything?
  - Mass evacuations
  - Satellite phones - a good idea?
  - People 'alerter'

Record. Fix.

# Rolling out a disaster (Example 2/4)

- Test
  - Distributed offices responsible for carrying operations
  - Oh wait! Unrelated fiber cut incident! Datacenter down!
- Response
  - Teams bring back systems with no help or communication from HQ
- Did we learn anything?
  - Where is our monitoring?
  - Is it possible to DoS cafes?
  - Great - approvals system works! Now what?
  - Phones - wait...
  - How quickly can we recover? Good enough?

Record. Fix.

# Rolling out a disaster (Example 3/4)

- Test
  - Oh wait! Massive power outage! Many days!

- Response
  - Emergency funding to respond.
  - Run on backup generators.
  - Longer-term outage - capacity concerns addressed.

- Did we learn anything?
  - Can we seamlessly cut to generators at full load?
  - For how long? Do we get into trouble?
  - If the datacenter was down for n hours, and we fixed everything why is nothing coming back up?

Record. Fix.

# Rolling out a disaster (Example 4/4)

- Test
  - Meanwhile at HQ, there are a lot of issues to resolve.
  - 'I am travelling, send me back.'
  - 'Do customers need to pay us?'

- Response
  - New policies on the fly
  - Sharing resources (food, shelter etc.)

- Did we learn anything?
  - Creativity and a culture that promotes flexibility helps a lot.
  - Communications is hard
  - Exhaustion and decisions

Record. Fix.

# Meta: What did we learn?

- Post-mortem culture: Failures are inevitable - the best we can do is be prepared for them and learn from them. Fix!

# Meta: What did we learn?

- Post-mortem culture: Failures are inevitable - the best we can do is be prepared for them and learn from them. Fix!
- Continuous simulations and testing: An untested plan is not really a plan. Test against them. All the time.

# Meta: What did we learn?

- Post-mortem culture: Failures are inevitable - the best we can do is be prepared for them and learn from them. Fix!
- Continuous simulations and testing: An untested plan is not really a plan. Test against them. All the time.
- Test everything: There is no *real* distinction between business continuity and disaster recovery - people and technology co-exist.

# Meta: What did we learn?

- Post-mortem culture: Failures are inevitable - the best we can do is be prepared for them and learn from them. Fix!
- Continuous simulations and testing: An untested plan is not really a plan. Test against them. All the time.
- Test everything: There is no *real* distinction between business continuity and disaster recovery - people and technology co-exist.
- Rinse and repeat: Repetition is important. A system or document that is never used is not helpful when it matters.

Thank you!