ITMT 430

WEEK 03 – CHAPTER 09

THE TECHNICAL PRACTICES OF FLOW

Objectives

- Create Technical Practices and architecture required to enable fast flow
- Create this flow without creating chaos and disruptions
- Implement Continuous Delivery System
- Understand what a deployment pipeline is
- Understand how to enable and architect for a low-risk release

Deployment Pipeline

Goal of a pipeline

- Create fast and reliable flow form Dev to Ops (the 1st way)
- To do this we **must** ensure that we are always using a production like environment**--**even in our test environment
- Your goal will be to create two environments
  - Test
  - Production

Em Campbell Pretty

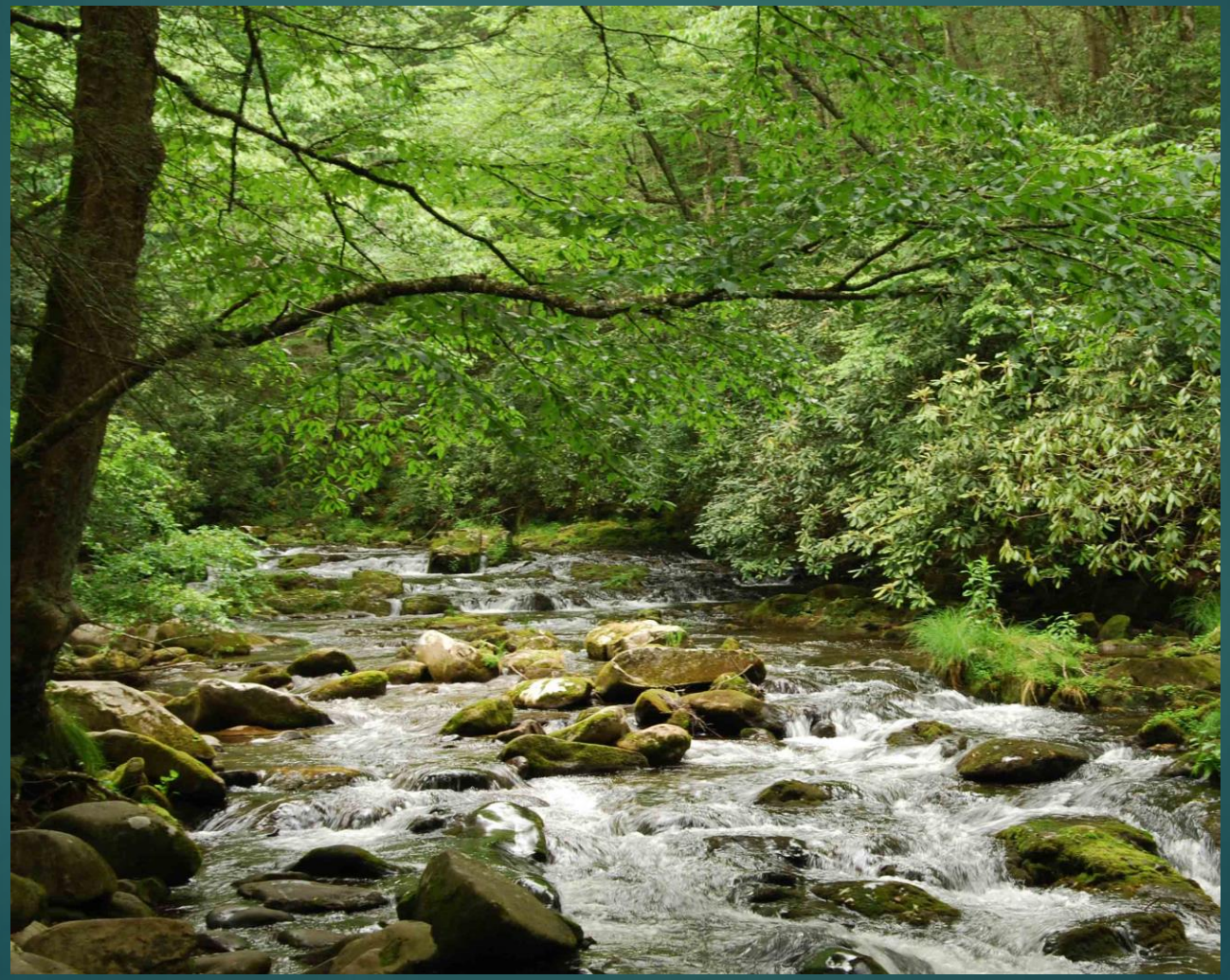
- Enterprise Data Warehouse lead at an Australian Telecom
- $200 million dollar program
- At the time of her arrival—there were 10 streams of work in progress
- All using Waterfall
- All behind schedule (Why? Large batch perhaps?)
- Only one stage was doing any testing
- All projects falling well short of design specs
- Underperforming

Project Description (as the kids say…)

1 Year Later…

- They moved to an Agile project management
  - Many smaller meetings
  - Smaller code batches
  - Smaller easier measurable goals in relation to code and testing
  - Surprisingly no real change in the previous slide
  - What could be the issue?
  - Many complained that it was a “management” or “leadership” issue
  - Called a company wide meeting – full stop to solve this

Value Stream



Value Stream

- By looking at the stream they realized their issues
  - Only 50% of their code in the development and test environments matched what was in production
  - They encountered so many defects because they weren't testing what they were deploying
  - Re-engineered the processes
  - Now all changes were in version control
  - Single point of truth for the system
  - Problems thought to be "management" or "person-based" turned out to be poorly understood process based

Dev, Test, and Prod

- Enable Dev, Test, and Prod environments at the start
  - Since we are a school and not a company we will hold off on the production environment for the moment
- Dev = Developers
  - Every person on the team is technically a developer
  - Each person should be able to deploy and build your entire project from your Github repository on their own laptop (especially the developers)
  - Requires sound build processes from Ops and actual documentation
  - How?
    - VirtualBox
    - Vagrant
    - Packer

Test

- Test is an environment that is not local to your laptop but an exact copy of your production environment
  - Webservers, hardware, distributed configurations, etc. etc.
  - No lead time – this environment is always available or can be built with the latest changes all from a single source repository

Test Could Be…

- Virtualized Environment
  - Vmware images, Vagrant scripts, Amazon EC2 instances
- Building the automated environment
  - PXE install on bare metal
- Using "Infrastructure as Code"
  - Configuration management tools such:
  - Chef, Puppet, Ansible, CFEngine, Salt, MAAS
- Automated OS configuration tools
  - Preseed or kickstart or Windows Answer Files
- Assembling an environment from a set of virtual images
  - Vagrant and Packer
- Spinning up new environments in the Cloud (AWS, GCE, MS Azure, OpenStack)

This works because…

- We carefully defined out specifications of the environment ahead of time.
- We are documenting and pushing everything
- This reduces defects and increases throughput and productivity.
- Let's see this in action