

Evolving Continuous Delivery

Chris Read

What We're Told

What We're Told

- **Single** Source Repository

What We're Told

- **Automate** Build and Testing

What We're Told

- **Publish** Latest Distributable

What We're Told

- Every Commit Builds

What We're Told

- Test in **Production** Like Environment

What We're Told

- Keep Builds **Fast**

What We're Told

- Use **Information** Radiators

What We're Told

- Automate Deployment

What We're Told

- Build Binary **Once**

What We're Told

- **Promote** Binary Through Stages

Evolution

Initial State

- New team of **talented** and **impatient** developers
- Starting to create trading applications for an **established** desk

Initial State

- Releasing daily from **developer** workstations to **production**
- **No** Continuous Integration!
- Using **Fig** for dependency management

Stage I

- **Standardise** and Refactor the build scripts
- Add **Continuous Integration** server
- Set up an **Information Radiator**

Stage II

- Create a **standard** deployment script
- Turn the scripts into **dependencies**
- Try a **different** CI server

Stage III

- **Sideline** the Continuous Integration loop
- Bake the Continuous Integration **safeties** into the deployment scripts

Stage IV

- Automate server builds
- Start to scale services out

Stage V

- Fracture **services** out into stacks
- Bring the Continuous Integration loop back **inline**
- **Stage** the binaries

What Have I Learned?

- The Things We're Told **still hold**, but implementation will vary greatly
- Constantly **re-evaluate** your tools and your process - ensure you have the right tool
- Always **question** the return on investment

Thank You

