Chapter 4 – Technical Practices

Continuous Delivery

Move to Production Safety, Quickly and Sustainably

5 Key Principles

1. Build Quality In
   - Eliminate Inspection
   - Use Tools and People to find issues quickly
2. Work in Small Batches
   - Faster to Check'
   - Faster Feedback
3. Computers perform Repetitive Tasks; People solve Problems
4. Relentlessly Pursue Continuous Improvement
   - Strive to get better
5. Everyone is Responsible
   - Collaboration

Outcomes

1. Strong Identification with Organization
2. Higher Software Delivery Performance
3. Lower Change Failure Rates
4. Leads to Generative Culture (Performance Oriented)

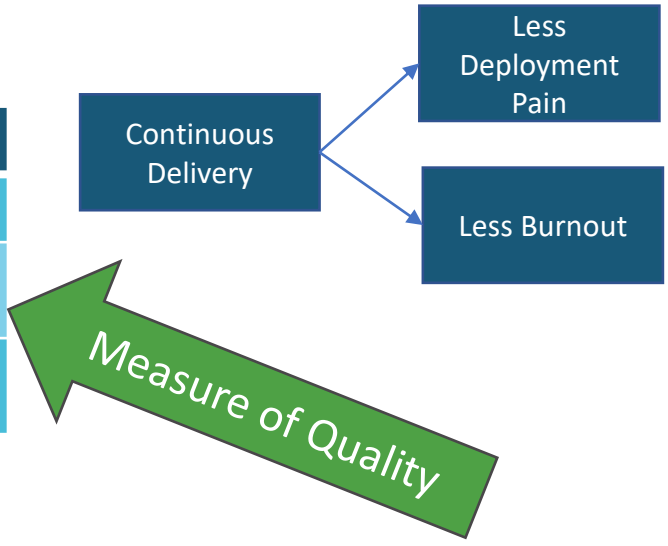
	High Performers	Low Performers
New Work	49%	38%
Unplanned Work (Rework)	21%	27%
Other Work (Meetings, etc)	30%	35%

Foundations

1. Comprehensive Configuration Management
   - Environments are Version Controlled
2. Continuous Integration (CI)
   - Short Branches
   - Commits lead to Tests
3. Continuous Testing
   - Part of Development
   - Automated

Additional Benefits

1. Decrease Deployment Pain and Team Burnout
2. Deployment during Business Hours
3. Developers can accept responsibility for Global Outcomes (Quality)



Continuous Development Practices

1. Version Control
2. Test Automation
   1. Need to be Reliable
   2. Developers responsible to create and maintain
   3. Every Commit should trigger Automated Tests
3. Test Data Management
   1. Able to acquire test data
4. Trunk-Based Development
   1. Fewer Branches (<3)
   2. Shorter Branches (< 1 day)
   3. Don't use Git Workflow
5. Information Security
   1. High Performing teams incorporate Information Security into their Delivery Process
   2. Increases Software Delivery Performance