# Recap: VS Live 2018

PRESENTED BY WING CHAN

# Agenda

- Things about VS Live 2018
- Topics/Sessions attended
- Sessions Highlighted
  - Microsoft DevOps Transformation
  - Deliver on Cadence, Release on Demand
- Q&A

# Things about VS Live 2018

- What: .NET developer conference
- When: August 13 – 17, 2018
- Where: Microsoft Headquarters, Redmond, WA
- Who: Ken Plunk & Wing Chan
- 5 main tracks:
  - DevOps
  - Cloud Computing
  - Data Science/AI
  - Web/Mobile development
  - Software Practices
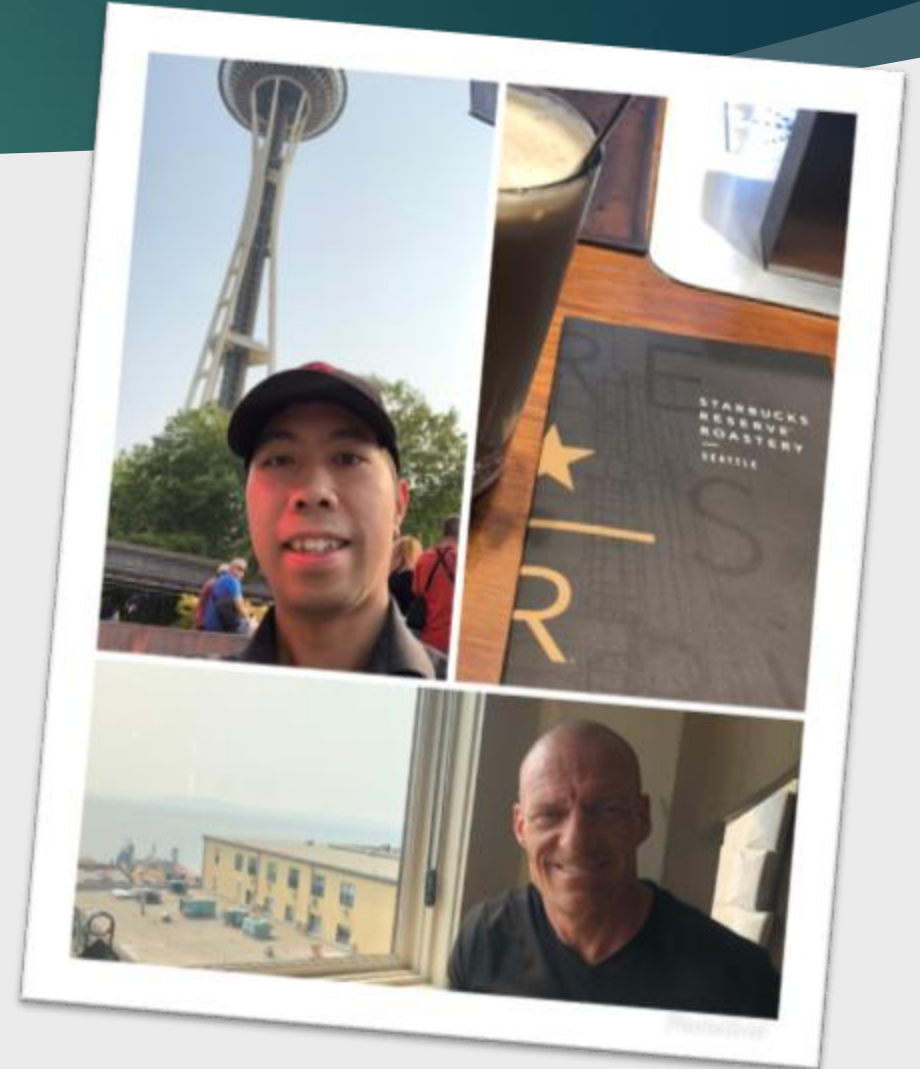
# Topics/ Sessions Attended

- Mobile development
  - Xamarin by Microsoft
  - Flutter by Google
- Data science/AI
  - Azure Cognitive Service
  - TensorFlow
- Cloud Computing
  - Microservice Architecture
  - Azure DevOps with VSTS, Docker and Kubernetes
- Software Practices
  - Agile/Scrum
  - Azure DevOps/VSTS

# Session 1:Microsoft's DevOps Transformation

- Presented by Abel Wang (Senior Cloud Developer Advocate)
- Share Microsoft's DevOps team transforming from Waterfall to Agile
- Formula 1 Pit Stops: 1950s vs Today
- https://www.youtube.com/watch?v=RRy_73ivcms

# The Journey

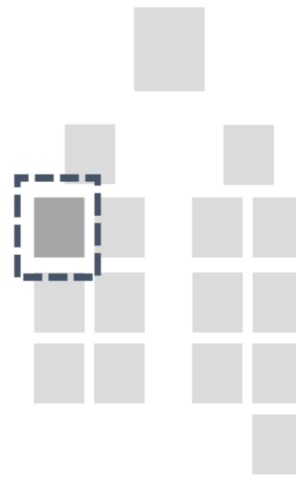- How Microsoft became agile and transformed a box software product into a service...
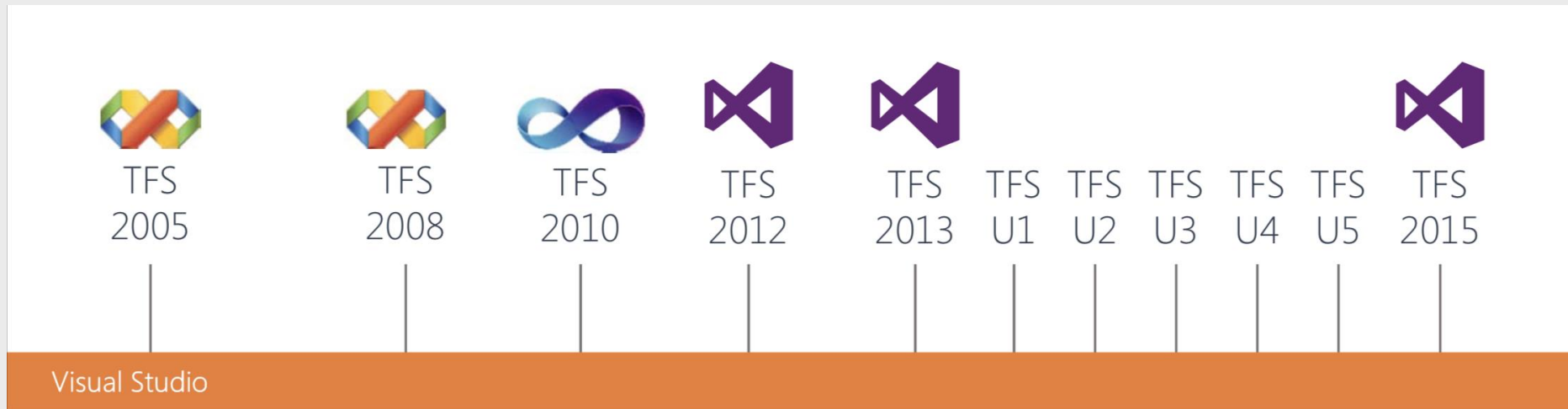
# Team Roles/Structure

- Program Manager – Responsible to ensure we're building the right thing.

- Engineer – Responsible to ensure we're building products that are fast, reliable, well-engineered, high quality and meet customers' needs.

- Teams: Cross discipline, 10-12 people, self-managing, autonomous backlog, intact for 12-18 months in the same physical team rooms
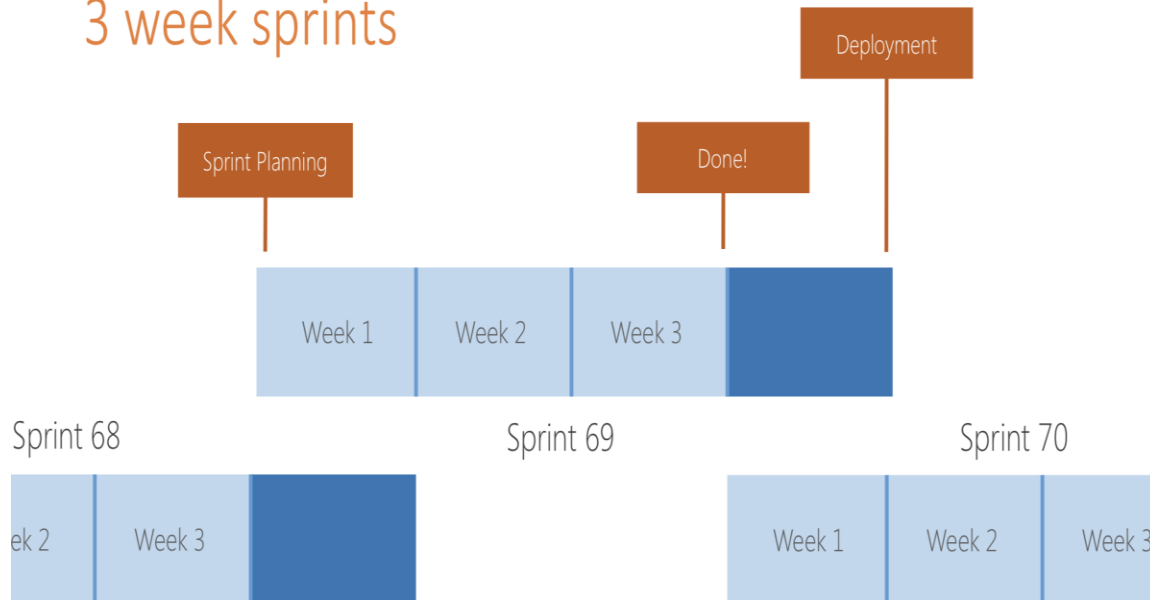
## Our teams

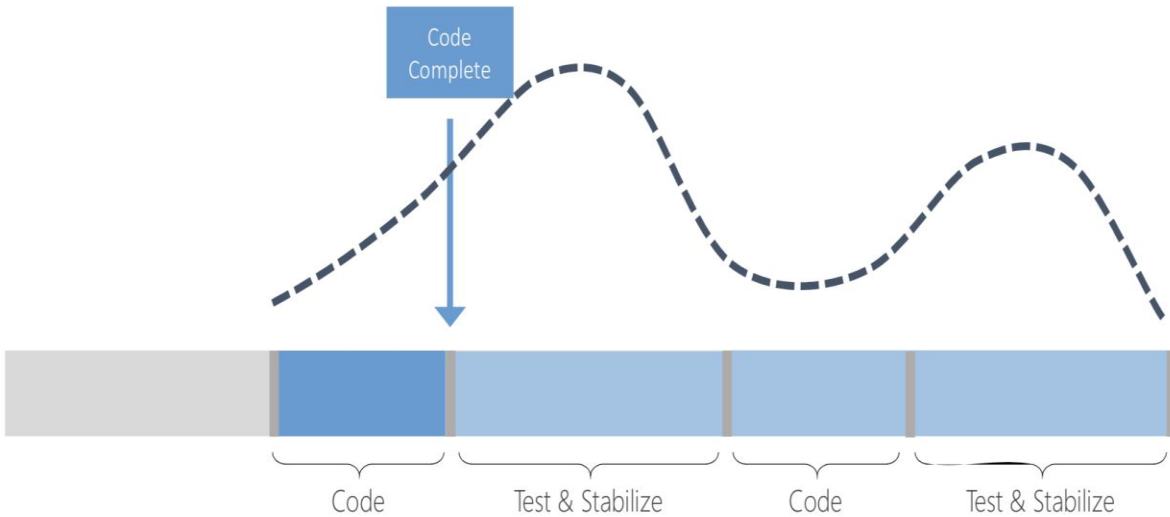Program Management

Engineering

# 3 week sprints

# Planning

## Planning

Leadership is responsible for the big picture

| Sprint | Plan | Season | Scenario |
|--------|------|--------|----------|
| 3 weeks | 3 sprints | 6 months | 18 months |
| 1 | 3 | 6 | 18 |

Teams are responsible for the detail
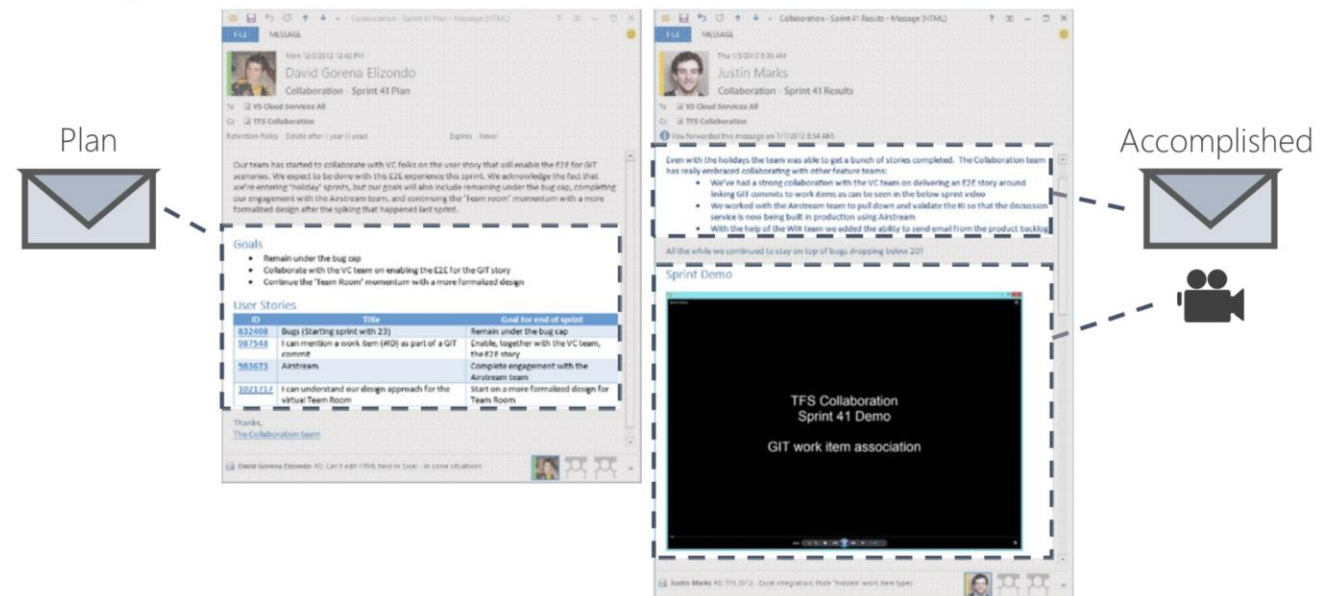
# Quality Before & After

Quality before

Quality after

# Tracking the quality

# Outcome

▶ No down time

▶ Fully automated

▶ Joint ownership between engineering and operations

▶ Decoupled services: Clear contacts

▶ Feature Flags: Launch Darkly (https://launchdarkly.com/)

# Session 2: Deliver on Cadence, Release on Demand

▶ Presented by Richard Hundhausen (Microsoft MVP, Scrum Trainer)

▶ How to reduce "Frictions" in software development teams

▶ 3 types of Frictions

  ▶ People

  ▶ Process

  ▶ Products

# People Frictions

## Friction

- 1. Command and Control
- 2. Component Teams
- 3. QA/DevOps Teams
- 4. Working as Individuals
- 5. Specialists
- 6. Adhoc Team
- 7.  Dislocated Teams

## Frictionless

- 1. Self-Organzation
- 2. Feature Teams
- 3. Cross-functional Dev Team
- 4. Working as a Team
- 5. T-Shaped
- 6. Long-lived Teams
- 7. Collocated Teams

# Process Frictions

**Friction**

- 1. Obfuscated Work
- 2. Multitasking
- 3. Multiple Projects
- 4. Develop by Layers
- 5. Done is Subjective
- 6. Dependencies
- 7. Team decides what to work on
- 8. Discontinuous Integration
- 9. Code Reviews

**Frictionless**

- 1. Visualize Work & Progress
- 2. Focus
- 3. Sprints
- 4. Thin-Slicing
- 5. Definition of "Done"
- 6. Make them Transparent, remove them
- 7. Stakeholder feedback
- 8. Continuous Integration
- 9. Pairing, Swarming, Mobbing

# Product Frictions

## Friction
- 1. Private Repositories
- 2. Pull Requests
- 3. Working in Branches
- 4. Manual Deployments
- 5. Manual Testing
- 6. User Acceptance Testing
- 7. Code Deployment
- 8. Local Optimization

## Frictionless
- 1. Collective Code Ownership
- 2. Pairing, Swarming, Mobbing +CI
- 3. Trunk-based Development
- 4. Automated Deployment
- 5. Automated Testing
- 6. Acceptance Testing
- 7. Feature Release
- 8. Global Optimization(System Thinking)

# Q&A

- Visual Studio Live 2018 Sessions download
- https://vslive.com/redmondslides