



Azure Weekend 2020



Agile Planning and Portfolio Management

Eng Teong Cheah

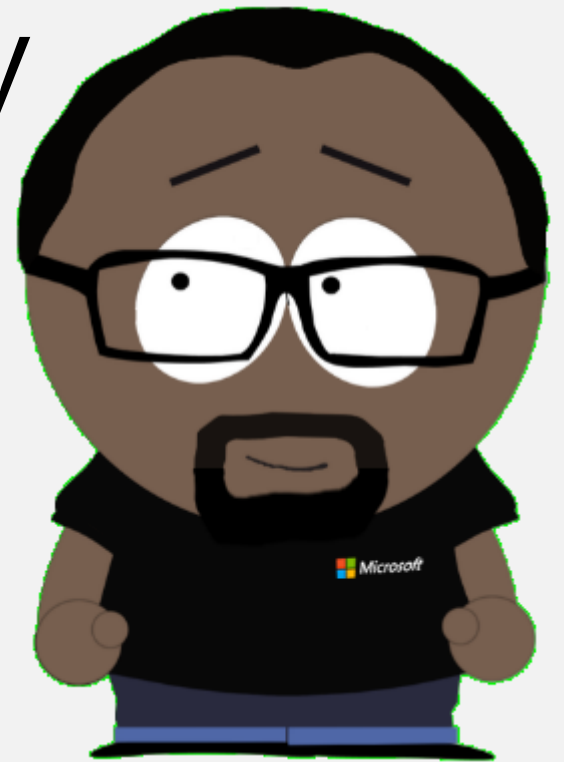
Microsoft MVP



“DevOps is the union of people, process, and products to enable continuous delivery of value to our end users.”

—Donovan Brown

<http://bit.ly/WhatIs-DevOps>





Transformation Planning

Separating Transformation Teams

There are several challenges when creating teams:

1. Availability of staff
2. Disruption of current procedures

To overcome the challenges, create a team that is:

1. Focused on the transformation
2. Well respected in their subject areas
3. Internal and external to the business

Defining Shared Goals

Project must have a clearly-defined set of measurable outcomes, like

- Reduce the time spent on fixing bug by 60%
- Reduce the time spent on unplanned work by 70%
- Reduce the out-of-hours work required by staff to no more than 10% of total working time
- Remove all direct patching of production systems

Setting Timelines for Goals

Measurable goals should have timelines that challenging yet achievable

Timelines should be a constant series of short-term goal –each clear and measurable

Shorter timelines have advantages:

1. Easier to change plans or priorities when necessary
2. Reduce delay between doing the work and getting feedback
3. Easier to keep organizational support when positive outcomes are apparent.



Project Selection

Greenfield and Brownfield Projects Defined

- Greenfield software projects develop in a totally new environment
- Brownfield software projects develop in the immediate presence of existing software applications/systems

Choosing Greenfield and Brownfield Projects

- Greenfield projects
 - Appears to be an easier starting point
 - A blank slate offers the chance to implement everything the way you want
- Brownfield projects
 - Comes with the baggage of existing code bases, existing teams, and often a great amount of technical debt
 - Spending time maintaining existing Brownfield applications, limits the ability to work on new code



Team Structures

Agile Development Practices Defined

- Waterfall approach
 - Define, analyze, build and test, and deliver
 - Hard to accurately define requirements, which can change over time
 - After delivery requires change requests and additional cost
- Agile approach
 - Emphasizes constantly adaptive planning, and early delivery with continual improvement
 - Development methods are based on releases and iterations
 - At the end of each iteration, there should be tested working code
 - Focused on these shorter-term outcomes

Principles of Agile Development

1. Satisfy the customer
2. Welcome changing requirements
3. Deliver working software frequently
4. Work together throughout the project
5. Build projects around motivated individuals
6. Use face-to-face conversation
7. Measure progress through working software
8. Agile processes promote sustainable development
9. Continuous attention to technical excellence and good design
10. Simplicity--the art of maximizing the amount of work not done
11. Use self-organizing teams
12. Reflect on how to become more effective



Migrating to Azure DevOps

What can Azure DevOps do?

- Azure Boards: agile planning, work item tracking, visualisation and reporting tool.
- Azure Pipelines: a language, platform and cloud agnostic CI/CD platform with support for containers or Kubernetes.
- Azure Repos: provides cloud-hosted private git repos.
- Azure Artifacts: provides integrated package management with support for Maven, npm, Python and NuGet package feeds from public or private sources.
- Azure Test Plans: provides an integrated planned and exploratory testing solution.

Designing an Authorization and Access Strategy

- Azure DevOps Services uses either a Microsoft account or Azure Active Directory, to protect and secure your data
- For non-Microsoft tools like Git, NuGet, or Xcode you can use personal access tokens
- Azure DevOps is pre-configured with default security groups and permissions
- You can also configure app access policies and conditional access policies

Migrating or Integrating Existing Work Management Tools

- Azure DevOps can be integrated with a variety of existing work management tools
 - [Trello integration tooling](#) is a free, flexible, and visual way to organize anything with anyone. For example, Visual Studio Team Services.
 - [Solidify](#) offers a tool for Jira to Azure DevOps migration
 - Third party organizations offer commercial tooling to assist with migrating other work management tools like Aha, BugZilla, ClearQuest


Migrating or Integrating Existing Test Management Tools

- Azure Test Plans are used to track sprints and milestones. There is a Test & Feedback extension available in the Visual Studio Marketplace.
- Other tools:
 - [Apache JMeter](#) is open source software written in Java and designed to load test functional behavior and measure performance
 - [Pester](#) is a tool that can be used to automate the testing of PowerShell code
 - [SoapUI](#) provides another testing framework for SOAP and REST testing



Demo

Agile Planning and Portfolio Management





Thank You

Feedback URL: <http://bit.ly/azwefeedback>

