

Microsoft Official Course



AZ-400T07

Designing a DevOps Strategy

AZ-400T07

Designing a DevOps Strategy

Information in this document, including URL and other Internet Web site references, is subject to change without notice. Unless otherwise noted, the example companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted herein are fictitious, and no association with any real company, organization, product, domain name, e-mail address, logo, person, place or event is intended or should be inferred. Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Microsoft Corporation.

Microsoft may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Microsoft, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

The names of manufacturers, products, or URLs are provided for informational purposes only and Microsoft makes no representations and warranties, either expressed, implied, or statutory, regarding these manufacturers or the use of the products with any Microsoft technologies. The inclusion of a manufacturer or product does not imply endorsement of Microsoft of the manufacturer or product. Links may be provided to third party sites. Such sites are not under the control of Microsoft and Microsoft is not responsible for the contents of any linked site or any link contained in a linked site, or any changes or updates to such sites. Microsoft is not responsible for webcasting or any other form of transmission received from any linked site. Microsoft is providing these links to you only as a convenience, and the inclusion of any link does not imply endorsement of Microsoft of the site or the products contained therein.

© 2019 Microsoft Corporation. All rights reserved.

Microsoft and the trademarks listed at <http://www.microsoft.com/trademarks>¹ are trademarks of the Microsoft group of companies. All other trademarks are property of their respective owners.

¹ <http://www.microsoft.com/trademarks>

MICROSOFT LICENSE TERMS

MICROSOFT INSTRUCTOR-LED COURSEWARE

These license terms are an agreement between Microsoft Corporation (or based on where you live, one of its

affiliates) and you. Please read them. They apply to your use of the content accompanying this agreement which

includes the media on which you received it, if any. These license terms also apply to Trainer Content and any

updates and supplements for the Licensed Content unless other terms accompany those items. If so, those terms

apply.

BY ACCESSING, DOWNLOADING OR USING THE LICENSED CONTENT, YOU ACCEPT THESE TERMS. IF YOU DO NOT ACCEPT THEM, DO

NOT

ACCESS, DOWNLOAD OR USE THE LICENSED CONTENT.

If you comply with these license terms, you have the rights below for each license you acquire.

1. DEFINITIONS.

1. "Authorized Learning Center" means a Microsoft IT Academy Program Member, Microsoft Learning Competency Member, or such other entity as Microsoft may designate from time to time.
2. "Authorized Training Session" means the instructor-led training class using Microsoft Instructor-Led Courseware conducted by a Trainer at or through an Authorized Learning Center.
3. "Classroom Device" means one (1) dedicated, secure computer that an Authorized Learning Center owns
 - or
 - controls that is located at an Authorized Learning Center's training facilities that meets
 - or
 - exceeds the hardware level specified for the particular Microsoft Instructor-Led Courseware.
4. "End User" means an individual who is (i) duly enrolled in and attending an Authorized Training Session
 - or Private Training Session, (ii) an employee of a MPN Member, or (iii) a Microsoft full-time employee.
5. "Licensed Content" means the content accompanying this agreement which may include the Microsoft
 - Instructor-Led Courseware or Trainer Content.
6. "Microsoft Certified Trainer" or "MCT" means an individual who is (i) engaged to teach a training session
 - to End Users on behalf of an Authorized Learning Center or MPN Member, and (ii) currently certified
 - as
 - a Microsoft Certified Trainer under the Microsoft Certification Program.

7. "Microsoft Instructor-Led Courseware" means the Microsoft-branded instructor-led training course that educates IT professionals and developers on Microsoft technologies. A Microsoft Instructor-Led Courseware title may be branded as MOC, Microsoft Dynamics or Microsoft Business Group courseware.
8. "Microsoft IT Academy Program Member" means an active member of the Microsoft IT Academy Program.
9. "Microsoft Learning Competency Member" means an active member of the Microsoft Partner Network program in good standing that currently holds the Learning Competency status.
10. "MOC" means the "Official Microsoft Learning Product" instructor-led courseware known as Microsoft Official Course that educates IT professionals and developers on Microsoft technologies.
11. "MPN Member" means an active Microsoft Partner Network program member in good standing.
12. "Personal Device" means one (1) personal computer, device, workstation or other digital electronic device that you personally own or control that meets or exceeds the hardware level specified for the particular Microsoft Instructor-Led Courseware.
13. "Private Training Session" means the instructor-led training classes provided by MPN Members for corporate customers to teach a predefined learning objective using Microsoft Instructor-Led Courseware. These classes are not advertised or promoted to the general public and class attendance is restricted to individuals employed by or contracted by the corporate customer.
14. "Trainer" means (i) an academically accredited educator engaged by a Microsoft IT Academy Program Member to teach an Authorized Training Session, and/or (ii) a MCT.
15. "Trainer Content" means the trainer version of the Microsoft Instructor-Led Courseware and additional supplemental content designated solely for Trainers' use to teach a training session using the Microsoft Instructor-Led Courseware. Trainer Content may include Microsoft PowerPoint presentations, trainer preparation guide, train the trainer materials, Microsoft One Note packs, classroom setup guide and Pre-release course feedback form. To clarify, Trainer Content does not include any software, virtual hard disks or virtual machines.
2. **USE RIGHTS.** The Licensed Content is licensed not sold. The Licensed Content is licensed on a **one copy per user basis**, such that you must acquire a license for each individual that accesses or

uses the

Licensed Content.

- 2.1 Below are five separate sets of use rights. Only one set of rights apply to you.

1. **If you are a Microsoft IT Academy Program Member:**

1. Each license acquired on behalf of yourself may only be used to review one (1) copy of the Microsoft Instructor-Led Courseware in the form provided to you.

If the

Microsoft

Instructor-Led Courseware is in digital format, you may install one (1)

copy on up

to

three (3) Personal Devices. You may not install the Microsoft Instruc-

tor-Led

Courseware

on a device you do not own or control.

2. For each license you acquire on behalf of an End User or Trainer, you may either:

distribute one (1) hard copy version of the Microsoft Instructor-Led

Courseware to one (1) End User who is enrolled in the Authorized Training Session, and only immediately prior to the commencement of the Authorized Training Session that is the subject matter of the Microsoft Instructor-Led Courseware being provided, **or**

provide one (1) End User with the unique redemption code and instructions on

how they can access one (1) digital version of the Microsoft Instructor-Led Courseware, **or**

provide one (1) Trainer with the unique redemption code and instructions on

how they can access one (1) Trainer Content, **provided you comply with the following:**

3. you will only provide access to the Licensed Content to those individuals who have acquired a valid license to the Licensed Content,
4. you will ensure each End User attending an Authorized Training Session has their own valid licensed copy of the Microsoft Instructor-Led Courseware that is the subject of the Authorized Training Session,
5. you will ensure that each End User provided with the hard-copy version of the Microsoft Instructor-Led Courseware will be presented with a copy of this agreement and each End User will agree that their use of the Microsoft Instructor-Led Courseware will be subject to the terms in this agreement prior to providing them with the Microsoft Instructor-Led Courseware. Each individual will be required to denote their acceptance of this agreement in a manner that is enforceable under

ware, local law prior to their accessing the Microsoft Instructor-Led Courseware,

6. you will ensure that each Trainer teaching an Authorized Training Session has their own valid licensed copy of the Trainer Content that is the subject of the Authorized

Training Session,

7. you will only use qualified Trainers who have in-depth knowledge of and experience with the Microsoft technology that is the subject of the Microsoft Instructor-Led

Courseware being taught for all your Authorized Training Sessions,

8. you will only deliver a maximum of 15 hours of training per week for each Authorized Training Session that uses a MOC title, and

9. you acknowledge that Trainers that are not MCTs will not have access to all of the trainer resources for the Microsoft Instructor-Led Courseware.

2. If you are a Microsoft Learning Competency Member:

1. Each license acquired on behalf of yourself may only be used to review one (1) copy of the Microsoft Instructor-Led Courseware in the form provided to you.

If the

Microsoft

Instructor-Led Courseware is in digital format, you may install one (1)

copy on up

to

three (3) Personal Devices. You may not install the Microsoft Instruc-

tor-Led

Courseware

on a device you do not own or control.

2. For each license you acquire on behalf of an End User or MCT, you may either:

distribute one (1) hard copy version of the Microsoft Instructor-Led

Courseware to one (1) End User attending the Authorized Training Session

and

only immediately prior to the commencement of the Authorized Training Session that is the subject matter of the Microsoft Instructor-Led Courseware provided, **or**

provide one (1) End User attending the Authorized Training Session with the

unique redemption code and instructions on how they can access one (1) digital

version of the Microsoft Instructor-Led Courseware, **or**

you will provide one (1) MCT with the unique redemption code and

instructions

on how they can access one (1) Trainer Content, **provided you comply with the following:**

3. you will only provide access to the Licensed Content to those individuals who have acquired a valid license to the Licensed Content,

4. you will ensure that each End User attending an Private Training Session has their own valid licensed copy of the Microsoft Instructor-Led Courseware that is the subject of the Private Training Session,
 5. you will ensure that each End User provided with a hard copy version of the Microsoft Instructor-Led Courseware will be presented with a copy of this agreement and each End User will agree that their use of the Microsoft Instructor-Led Courseware will be subject to the terms in this agreement prior to providing them with the Microsoft Instructor-Led Courseware. Each individual will be required to denote their acceptance of this agreement in a manner that is enforceable under local law prior to their accessing the Microsoft Instructor-Led Courseware,
 6. you will ensure that each Trainer teaching an Private Training Session has their own valid licensed copy of the Trainer Content that is the subject of the Private Training Session,
 7. you will only use qualified Trainers who hold the applicable Microsoft Certification credential that is the subject of the Microsoft Instructor-Led Courseware being taught for all your Private Training Sessions,
 8. you will only use qualified MCTs who hold the applicable Microsoft Certification credential that is the subject of the MOC title being taught for all your Private Training Sessions using MOC,
 9. you will only provide access to the Microsoft Instructor-Led Courseware to End Users, and
 10. you will only provide access to the Trainer Content to Trainers.
3. **If you are a MPN Member:**
1. Each license acquired on behalf of yourself may only be used to review one (1) copy of the Microsoft Instructor-Led Courseware in the form provided to you. If the Microsoft Instructor-Led Courseware is in digital format, you may install one (1) copy on up to three (3) Personal Devices. You may not install the Microsoft Instructor-Led Courseware on a device you do not own or control.

2. For each license you acquire on behalf of an End User or Trainer, you may either:

distribute one (1) hard copy version of the Microsoft Instructor-Led Courseware to one (1) End User attending the Private Training Session, and only immediately prior to the commencement of the Private Training Session that is the subject matter of the Microsoft Instructor-Led Courseware being provided, **or**

provide one (1) End User who is attending the Private Training Session with the unique redemption code and instructions on how they can access one (1) digital version of the Microsoft Instructor-Led Courseware, **or**

you will provide one (1) Trainer who is teaching the Private Training Session with the unique redemption code and instructions on how they can access one (1) Trainer Content,

3. you will only provide access to the Licensed Content to those individuals who have acquired a valid license to the Licensed Content,
4. you will ensure that each End User attending an Private Training Session has their own valid licensed copy of the Microsoft Instructor-Led Courseware that is the subject of the Private Training Session,
5. you will ensure that each End User provided with a hard copy version of the Microsoft Instructor-Led Courseware will be presented with a copy of this agreement and each End User will agree that their use of the Microsoft Instructor-Led Courseware will be subject to the terms in this agreement prior to providing them with the Microsoft Instructor-Led Courseware. Each individual will be required to denote their acceptance of this agreement in a manner that is enforceable under local law prior to their accessing the Microsoft Instructor-Led Courseware,
6. you will ensure that each Trainer teaching an Private Training Session has their own valid licensed copy of the Trainer Content that is the subject of the Private Training Session,
7. you will only use qualified Trainers who hold the applicable Microsoft Certification credential that is the subject of the Microsoft Instructor-Led Courseware being taught for all your Private Training Sessions,
8. you will only use qualified MCTs who hold the applicable Microsoft Certification credential that is the subject of the MOC title being taught for all your Private Training Sessions using MOC,

9. you will only provide access to the Microsoft Instructor-Led Courseware to End Users, and

10. you will only provide access to the Trainer Content to Trainers.

4. If you are an End User:

For each license you acquire, you may use the Microsoft

Instructor-Led Courseware solely for your personal training use. If the Microsoft Instructor-Led

Courseware is in digital format, you may access the Microsoft Instructor-Led

Courseware online using the unique redemption code provided to you by the training provider

and install and use one (1) copy of the Microsoft Instructor-Led Courseware on up to

three (3) Personal Devices. You may also print one (1) copy of the Microsoft Instructor-Led

Courseware. You may not install the Microsoft Instructor-Led Courseware on a device you do

not own or control.

5. If you are a Trainer.

1. For each license you acquire, you may install and use one (1) copy of the Trainer

Content in the form provided to you on one (1) Personal Device solely to prepare and

deliver an Authorized Training Session or Private Training Session, and install

one (1) additional copy on another Personal Device as a backup copy, which may be

used only to reinstall the Trainer Content. You may not install or use a copy of the

Trainer Content on a device you do not own or control. You may also print one (1)

copy of the Trainer Content solely to prepare for and deliver an Authorized

Training Session or Private Training Session.

2. You may customize the written portions of the Trainer Content that are logically

associated with instruction of a training session in accordance with the most recent

version of the MCT agreement. If you elect to exercise the foregoing rights, you

agree to comply with the following: (i) customizations may only be used for

teaching Authorized Training Sessions and Private Training Sessions, and (ii) all

customizations will comply with this agreement. For clarity, any use of "customize" re-

fers only to changing the order of slides and content, and/or not using all the

slides or

content, it does not mean changing or modifying any slide or content.

- **2.2 Separation of Components.** The Licensed Content is licensed as a single unit and you may not separate their components and install them on different devices.
 - **2.3 Redistribution of Licensed Content.** Except as expressly provided in the use rights above, you may not distribute any Licensed Content or any portion thereof(including any permitted modifications)to any third parties without the express written permission of Microsoft.
 - **2.4 Third Party Notices.** The Licensed Content may include third party code that Microsoft, not the third party, licenses to you under this agreement. Notices, if any, for the third party code are included for your information only.
 - **2.5 Additional Terms.** Some Licensed Content may contain components with additional terms, conditions, and licenses regarding its use. Any non-conflicting terms in those conditions and licenses also apply to your use of that respective component and supplements the terms described in this agreement.
3. **LICENSED CONTENT BASED ON PRE-RELEASE TECHNOLOGY.** If the Licensed Content's subject matter is based on a pre-release version of Microsoft technology ("**Pre-release**"), then in addition to the other provisions in this agreement, these terms also apply:
1. **Pre-Release Licensed Content.** This Licensed Content subject matter is on the Pre-release version of the Microsoft technology. The technology may not work the way a final version of the technology will and we may change the technology for the final version. We also may not release a final version. Licensed Content based on the final version of the technology may not contain the same information as the Licensed Content based on the Pre-release version. Microsoft is under no obligation to provide you with any further content, including any Licensed Content based on the final version of the technology.
 2. **Feedback.** If you agree to give feedback about the Licensed Content to Microsoft, either directly or through its third party designee, you give to Microsoft without charge, the right to use, share and commercialize your feedback in any way and for any purpose. You also give to third parties, without charge, any patent rights needed for their products, technologies and services to use or interface with any specific parts of a Microsoft technology, Microsoft product, or service that

includes the feedback. You will not give feedback that is subject to a license that requires

Microsoft to license its technology, technologies, or products to third parties because we include

your feedback in them. These rights survive this agreement.

3. **Pre-release Term.** If you are an Microsoft IT Academy Program Member, Microsoft Learning Competency Member, MPN Member or Trainer, you will cease using all copies of the Licensed Content on

the Pre-release technology upon (i) the date which Microsoft informs you is the end date for using

the

Licensed Content on the Pre-release technology, or (ii) sixty(60)days after the commercial release

of

the technology that is the subject of the Licensed Content, whichever is earliest

("Pre-release

term"). Upon expiration or termination of the Pre-release term, you will irretrievably

delete and

destroy all copies of the Licensed Content in your possession or under your control.

4. **SCOPE OF LICENSE.** The Licensed Content is licensed, not sold. This agreement only gives you some rights

to use the Licensed Content. Microsoft reserves all other rights. Unless applicable law gives

you more

rights

despite this limitation, you may use the Licensed Content only as expressly permitted in this agreement. In

doing so, you must comply with any technical limitations in the Licensed Content that only

allows you to use

it in certain ways. Except as expressly permitted in this agreement, you may not:

- access or allow any individual to access the Licensed Content if they have not acquired a valid license for the Licensed Content,
- alter, remove or obscure any copyright or other protective notices(including watermarks), branding or identifications contained in the Licensed Content,
- modify or create a derivative work of any Licensed Content,
- publicly display, or make the Licensed Content available for others to access or use,
- copy, print, install, sell, publish, transmit, lend, adapt, reuse, link to or post, make available or distribute the Licensed Content to any third party,
- work around any technical limitations in the Licensed Content, or
- reverse engineer, decompile, remove or otherwise thwart any protections or disassemble the Licensed

Content except and only to the extent that applicable law expressly permits, despite

this

limitation.

5. **RESERVATION OF RIGHTS AND OWNERSHIP.** Microsoft reserves all rights not expressly granted to you in this agreement. The Licensed Content is protected by copyright and other intellectual property laws and treaties. Microsoft or its suppliers own the title, copyright, and other intellectual property rights in the Licensed Content.
6. **EXPORT RESTRICTIONS.** The Licensed Content is subject to United States export laws and regulations. You must comply with all domestic and international export laws and regulations that apply to the Licensed Content. These laws include restrictions on destinations, end users and end use. For additional information, see www.microsoft.com/exporting.
7. **SUPPORT SERVICES.** Because the Licensed Content is "as is", we may not provide support services for it.
8. **TERMINATION.** Without prejudice to any other rights, Microsoft may terminate this agreement if you fail to comply with the terms and conditions of this agreement. Upon termination of this agreement for any reason, you will immediately stop all use of and delete and destroy all copies of the Licensed Content in your possession or under your control.
9. **LINKS TO THIRD PARTY SITES.** You may link to third party sites through the use of the Licensed Content. The third party sites are not under the control of Microsoft, and Microsoft is not responsible for the contents of any third party sites, any links contained in third party sites, or any changes or updates to third party sites. Microsoft is not responsible for webcasting or any other form of transmission received from any third party sites. Microsoft is providing these links to third party sites to you only as a convenience, and the inclusion of any link does not imply an endorsement by Microsoft of the third party site.
10. **ENTIRE AGREEMENT.** This agreement, and any additional terms for the Trainer Content, updates and supplements are the entire agreement for the Licensed Content, updates and supplements.
11. **APPLICABLE LAW.**
 1. United States. If you acquired the Licensed Content in the United States, Washington state law governs the interpretation of this agreement and applies to claims for breach of it, regardless of conflict of laws principles. The laws of the state where you live govern all other claims, including claims under state consumer protection laws, unfair competition laws, and in tort.

2. Outside the United States. If you acquired the Licensed Content in any other country, the laws of that country apply.
12. **LEGAL EFFECT.** This agreement describes certain legal rights. You may have other rights under the laws of your country. You may also have rights with respect to the party from whom you acquired the Licensed Content. This agreement does not change your rights under the laws of your country if the laws of your country do not permit it to do so.
13. **DISCLAIMER OF WARRANTY. THE LICENSED CONTENT IS LICENSED "AS-IS" AND "AS AVAILABLE." YOU BEAR THE RISK OF USING IT. MICROSOFT AND ITS RESPECTIVE AFFILIATES GIVES NO EXPRESS WARRANTIES, GUARANTEES, OR CONDITIONS. YOU MAY HAVE ADDITIONAL CONSUMER RIGHTS UNDER YOUR LOCAL LAWS WHICH THIS AGREEMENT CANNOT CHANGE. TO THE EXTENT PERMITTED UNDER YOUR LOCAL LAWS, MICROSOFT AND ITS RESPECTIVE AFFILIATES EXCLUDES ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT.**
14. **LIMITATION ON AND EXCLUSION OF REMEDIES AND DAMAGES. YOU CAN RECOVER FROM MICROSOFT, ITS RESPECTIVE AFFILIATES AND ITS SUPPLIERS ONLY DIRECT DAMAGES UP TO US\$5. 00. YOU CANNOT RECOVER ANY OTHER DAMAGES, INCLUDING CONSEQUENTIAL, LOST PROFITS, SPECIAL, INDIRECT OR INCIDENTAL DAMAGES.**

This limitation applies to

- anything related to the Licensed Content, services, content (including code) on third party Internet sites or third-party programs; and
- claims for breach of contract, breach of warranty, guarantee or condition, strict liability, negligence, or other tort to the extent permitted by applicable law.

It also applies even if Microsoft knew or should have known about the possibility of the damages. The above limitation or exclusion may not apply to you because your country may not allow the exclusion or limitation of incidental, consequential or other damages.

Please note: As this Licensed Content is distributed in Quebec, Canada, some of the clauses in this agreement

are provided below in French.

Remarque: Ce le contenu sous licence étant distribué au Québec, Canada, certaines des clauses dans ce contrat

sont fournies ci-dessous en français.

EXONÉRATION DE GARANTIE. Le contenu sous licence visé par une licence est offert «tel quel». Toute utilisation de ce contenu sous licence est à votre seule risque et péril. Microsoft n'accorde aucune autre garantie

expresse. Vous pouvez bénéficier de droits additionnels en vertu du droit local sur la protection des consommateurs, que ce contrat ne peut modifier. La ou elles sont permises par le droit local, les garanties

implicites de qualité marchande, d'adéquation à un usage particulier et d'absence de contrefaçon sont exclues.

LIMITATION DES DOMMAGES-INTÉRÊTS ET EXCLUSION DE RESPONSABILITÉ POUR LES DOMMAGES. Vous pouvez obtenir de

Microsoft et de ses fournisseurs une indemnisation en cas de dommages directs uniquement à hauteur de 5,00 \$

US. Vous ne pouvez prétendre à aucune indemnisation pour les autres dommages, y compris les dommages

spéciaux, indirects ou accessoires et pertes de bénéfices.

Cette limitation concerne:

- tout ce qui est relié au contenu sous licence, aux services ou au contenu (y compris le code) figurant sur des sites Internet tiers ou dans des programmes tiers; et
- les réclamations au titre de violation de contrat ou de garantie, ou au titre de responsabilité stricte, de négligence ou d'une autre faute dans la limite autorisée par la loi en vigueur.

Elle s'applique également, même si Microsoft connaissait ou devrait connaître l'éventualité d'un tel dommage. Si

votre pays n'autorise pas l'exclusion ou la limitation de responsabilité pour les dommages indirects, accessoires

ou de quelque nature que ce soit, il se peut que la limitation ou l'exclusion ci-dessus ne s'appliquera pas à votre égard.

EFFET JURIDIQUE. Le présent contrat décrit certains droits juridiques. Vous pourriez avoir d'autres droits prévus par les lois de votre pays. Le présent contrat ne modifie pas les droits que vous confèrent les lois de

votre pays si celles-ci ne le permettent pas.

Revised November 2014



Contents

■	Module 0 Welcome	1
	Start Here	1
■	Module 1 Planning for DevOps	7
	Transformation Planning	7
	Project Selection	9
	Team Structures	13
	Module 1 Lab and Review Questions	20
■	Module 2 Planning for Quality and Security	23
	Planning a Quality Strategy	23
	Planning Secure Development	29
	Module 2 Lab and Review Questions	33
■	Module 3 Migrating and Consolidating Artifacts and Tools	35
	Migrating and Consolidating Artifacts	35
	Migrating and Integrating Source Control	38
	Module 3 Lab and Review Questions	41



Module 0 Welcome

Start Here

Azure DevOps Curriculum

Welcome to the **Design a DevOps Strategy** course. This course is part of a series of courses to help you prepare for the AZ-400, **Microsoft Azure DevOps Solutions**¹ certification exam.

The DevOps certification exam is for DevOps professionals who combine people, process, and technologies to continuously deliver valuable products and services that meet end user needs and business objectives. DevOps professionals streamline delivery by optimizing practices, improving communications and collaboration, and creating automation. They design and implement strategies for application code and infrastructure that allow for continuous integration, continuous testing, continuous delivery, and continuous monitoring and feedback.

Exam candidates must be proficient with Agile practices. They must be familiar with both Azure administration and Azure development and experts in at least one of these areas. Azure DevOps professionals must be able to design and implement DevOps practices for version control, compliance, infrastructure as code, configuration management, build, release, and testing by using Azure technologies.

AZ-400 Study Areas	Weights
Implement DevOps Development Processes	20-25%
Implement Continuous Integration	10-15%
Implement Continuous Delivery	10-15%
Implement Dependency Management	5 -10%
Implement Application Infrastructure	15-20%
Implement Continuous Feedback	10-15%
Design a DevOps Strategy	20-25%

There are seven exam study areas. Each study area has a corresponding course. While it is not required that you have completed any of the other courses in the DevOps series before taking this course, it is

¹ <https://www.microsoft.com/en-us/learning/exam-AZ-400.aspx>

highly recommended that you start with the first course in the series, and progress through the courses in order.

✓ This course will focus on preparing you for the **Design a DevOps Strategy** area of the AZ-400 certification exam.

About this Course

Course Description

This course provides the knowledge and skills to design a DevOps strategy. Students will learn how to plan for transformation, select a project, and create team structures. Students will also learn how to develop quality and security strategies. Planning for migrating and consolidating artifacts and source control will also be covered.

Level: Intermediate

Audience

Students in this course are interested in planning DevOps projects or in passing the Microsoft Azure DevOps Solutions certification exam.

Prerequisites

- Students should have fundamental knowledge about Azure, version control, Agile software development, and core software development principles. It would be helpful to have experience in an organization that delivers software.
- It is recommended that you have experience working in an IDE, as well as some knowledge of the Azure portal. However, students who may not have a technical background in these technologies, but who are curious about DevOps practices as a culture shift, should be able to follow the procedural and expository explanations of continuous integration regardless.

Expected learning objectives

- Plan for a transformation with shared goals and timelines.
- Select a project and identify project metrics and KPIs.
- Create a team and agile organizational structure.
- Develop a project quality strategy.
- Plan for secure development practices and compliance rules.
- Migrate and consolidate artifacts.
- Migrate and integrate source control measures.

Syllabus

This course includes content that will help you prepare for the Microsoft Azure DevOps Solution certification exam. Other content is included to ensure you have a complete picture of DevOps. The course content includes a mix of videos, graphics, reference links, module review questions, and hands-on labs.

Module 1 – Planning for DevOps

In this module, students will learn about transformation planning, project selection, and team structures. Content includes:

- Transformation Planning

- Project Selection
- Team Structures
- Lab: Agile Planning and Portfolio Management with Azure Boards

Module 2 – Planning for Quality and Security

In this module, students will learn about developing a quality strategy and planning for secure development. Content includes:

- Planning a Quality Strategy
- Planning Secure development
- Lab: Feature Flag Management with LaunchDarkly and AzureDevOps

Module 3 – Migrating and Consolidating Artifacts and Tools

In this module, students will learn about migrating and consolidating artifacts, and migrating and integrating source control measures. Content includes:

- Migrating and Consolidating Artifacts
- Migrating and Integrating Source Control
- Lab: Integrating Azure Repos and Azure Pipelines with Eclipse

✓ This course uses the **Microsoft DevOps Lab Environment**² to provide a hands-on learning environment.

Lab Environment Setup

We highly recommend that you complete the assigned hands-on lab work. To do the hands-on labs, you will need to complete the following steps.

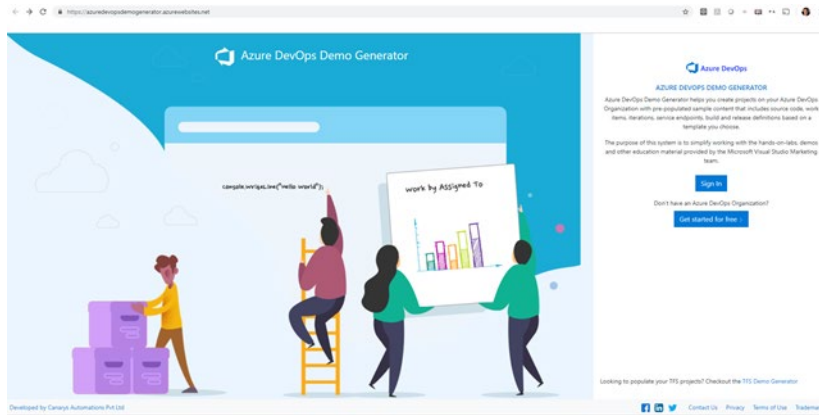
1. Sign up for a free **Azure DevOps account**³. Use the Sign up for a free account button to create your account. If you already have an account proceed to the next step.
2. Sign up for free **Azure Account**⁴. If you already have an account proceed to the next step.
3. To make it easier to get set up for testing Azure DevOps, a **Azure DevOps Generator Demo**⁵ program has been created. Click the **Sign In** button and sign in with your Azure DevOps account.

² <https://azuredevopslabs.com/>

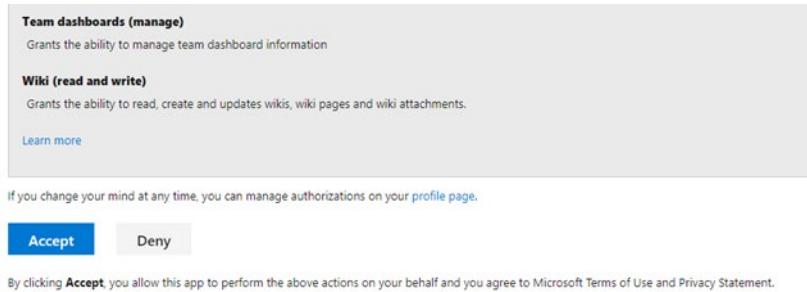
³ <https://www.azuredevopslabs.com/>

⁴ <https://azure.microsoft.com/en-us/free/>

⁵ <https://azuredevopsgenerator.azurewebsites.net/>

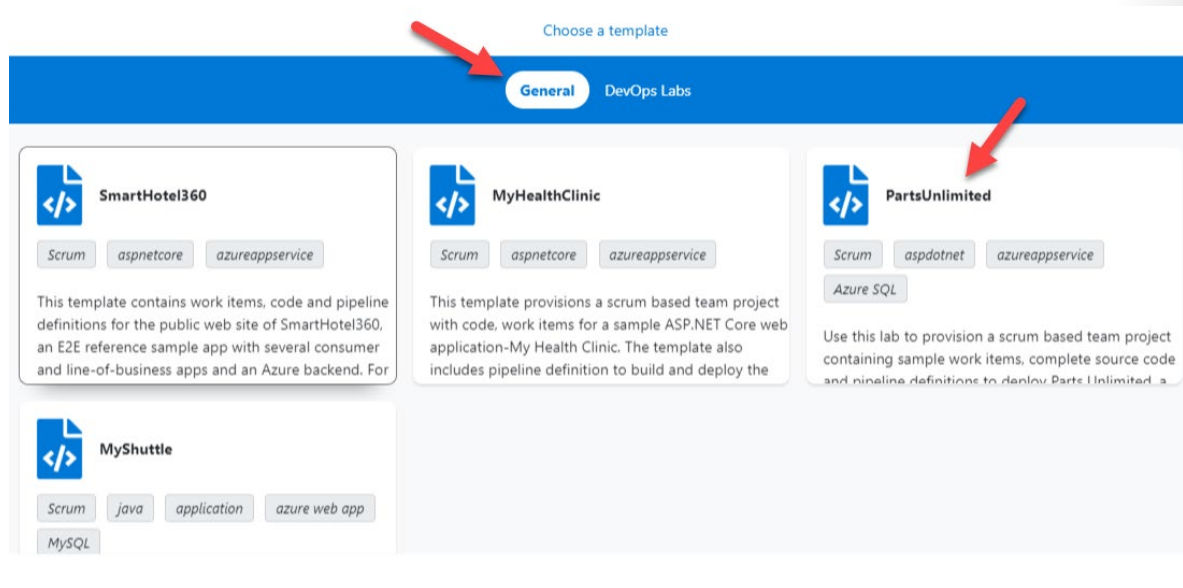


4. You will then be asked to confirm that the generator site can have permission to create objects in your Azure DevOps account.



5. If you agree, click the **Accept** button and you should be greeted by the Create New Project screen:

6. Select the appropriate organization (if you have more than one) and enter **Parts Unlimited** as the **New Project Name**, then click the ellipsis to view the available templates. These will change over time but you should see a screen similar to the following:

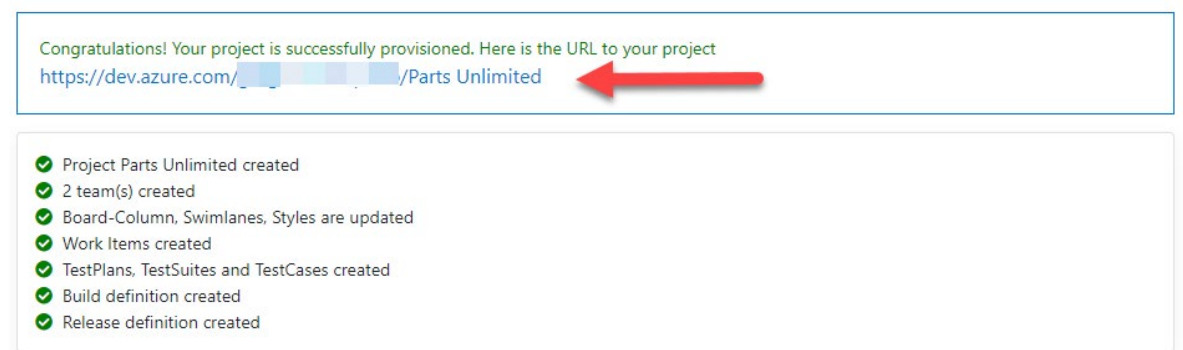


7. From the **General** tab, choose **PartsUnlimited**, then click **Select Template**.

The 'Create New Project' screen shows the following fields:

- Select Organization**: A dropdown menu.
- New Project Name**: A text box containing 'Parts Unlimited'.
- Selected Template**: A dropdown menu showing 'PartsUnlimited'.
- Create Project**: A blue button to proceed.

8. Now that the Create New Project screen is completed, click **Create Project** to begin the project creation phase.



9. When the project is successfully completed, click the link provided to go to your team project within Azure DevOps.

- ✓ Note that because Azure DevOps was previously called VSTS (Visual Studio Team Services), some of the existing hands-on labs might refer to VSTS rather than Azure DevOps.



Module 1 Planning for DevOps

Transformation Planning

Lesson Overview

This lesson includes the following topics:

- Separating Transformation Teams
- Defining Shared Goals
- Setting Timelines for Goals

Separating Transformation Teams

Unless you are building an entirely new organization, one of the big challenges with any Develops Transformation Project is that you'll be taking actions that conflict at least in some way with ongoing business states.

There are many aspects to this. The first challenge is the availability of staff. If the staff members, who were leading the transformation project, are also involved in existing day-to-day work within the organization, it will be difficult for them to focus on the transformation, at least in any meaningful way, particularly if their existing role directly impacts customer outcomes. We all know the desperate situations that involve customers will always win over a long-term project, like Develops Transformations.

Another issue will be the way that the organization currently operates. Existing processes and procedures have been implemented to support current business outcomes. The disruption that is required for a true Develops Transformation will usually end up challenging those existing processes and procedures. Doing that is often very difficult. Dr. Vijay Govindarajan and Dr. Chris Trimble of Dartmouth College have researched what's involved in allowing innovation to occur in organizations and noted that when this is successful, it's often been in spite of the existing organizational processes. They concluded that it only works where a separate team is created to

pursue the transformation. For DevOps transformations, the separate team should be made up of staff members, all of whom are focused on and measured on the transformation outcomes, and not involved in the operational day-to-day work. The team might also include some external experts that can fill the knowledge gaps and help to advise on processes that are new to the existing staff members. Ideally the staff members who were recruited for this should already be well-regarded throughout the organization and as a group they should offer a broad knowledge base so they can think outside the box.

Defining Shared Goals

Most management courses would tell you that if you want to change something, you need to make sure that it's measurable. DevOps transformations are no different. The project needs to have a clearly-defined set of measurable outcomes.

These outcomes should include specific measurable targets like:

- Reduce the time spent on fixing bugs 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.
- ✓ One of the key aims of DevOps is to provide greater customer value, so outcomes should have a customer value focus.

Setting Timelines for Goals

Measurable goals also need to have timelines. While it's easy to set longer-term goals, it's also easy to put off work when you know it's not needed for a while.

While overall projects should have timelines that span anywhere from a few months to a year or two in any DevOps transformation project, it's important to have a constant series of short-term goals. Every few weeks, the improvements made should be clear and measurable and ideally, also obvious to the organization and/or its customers. Clearly the timeline should not be too short. They should always be challenging yet achievable. A review should occur after each short-term goal to assist in planning the next one. There are several advantages of the shorter timelines. One key advantage is that it's easier to change plans or priorities when necessary. Another is that the reduced delay between doing work and getting feedback helps to ensure that the learnings and feedback are incorporated quickly. Finally, it's easier to keep organizational support when positive outcomes are apparent.

Project Selection

Lesson Overview

This lesson includes the following topics:

- Greenfield and Brownfield Projects Defined
- Choosing Greenfield and Brownfield Projects
- Choosing Systems of Record vs. Systems of Engagement
- Selecting Groups to Minimize Initial Resistance
- Identifying Project Metrics and KPIs

Greenfield vs Brownfield Projects Defined

The terms greenfield and brownfield have their origins in residential and industrial building projects. A greenfield project is one done on a green field, that is, undeveloped land. A brownfield project is one that was done on land that has been previously used for other purposes. Because of the land use that has previously occurred there could be challenges with reusing the land. Some of these would be obvious, like existing buildings but could also be less obvious, like polluted soil.

Applied to Software or DevOps Projects

The same terms are routinely applied to software projects and commonly used to describe DevOps projects. On the surface it can seem that a greenfield DevOps project would be easier to manage and to achieve success. There was no existing code base, no existing team dynamics or politics, and possibly no existing, rigid processes. Because of this there's a common misconception that DevOps is really only for greenfield projects, and that it suits startups best. However, a large number of very successful brownfield DevOps projects have occurred. The beauty of these projects is that there's often already a large gap between the customer expectations and what is being delivered, and the teams involved may well realize that the status quo needs to change, because they've lived the challenges and the limitations associated with what they're currently doing.

Choosing Greenfield and Brownfield Projects

When starting a DevOps transformation, you might need to choose between Greenfield and Brownfield projects. There is a common misconception that DevOps suits Greenfield projects better than Brownfield projects, but this is not the case.

Greenfield Projects

A Greenfield project will always appear to be an easier starting point, because a blank slate offers the chance to implement everything the way that you want. You might also have a better chance of avoiding existing business processes that do not align with your project plans.

For example, if current IT policies do not allow the use of cloud-based infrastructure, this might be allowed for entirely new applications that are designed for that environment from scratch. As another example, you might be able to sidestep internal political issues that are well-entrenched.

Brownfield Projects

While Brownfield projects come with the baggage of existing code bases, existing teams, and often a great amount of technical debt, they can still be ideal projects for DevOps transformations.

When your teams are spending large percentages of their time just maintaining existing Brownfield applications, you have limited ability to work on new code. It's important to find a way to reduce that time, and to make software releases less risky. A DevOps transformation can provide that.

The existing team members will often have been worn down by the limitations of how they have been working in the past, and be keen to try to experiment with new ideas. These are often systems that the organizations will be currently depending upon, so it might also be easier to gain stronger management buy in for these projects because of the size of the potential benefits that could be derived. Management might also have a stronger sense of urgency to point brownfield projects in an appropriate direction, when compared to greenfield projects that don't currently exist.

Choosing Systems of Record vs Systems of Engagement

When selecting systems as candidates for starting a DevOps transformation, it's important to consider the types of systems that you operate.

Some researcher suggests that organizations often use Bimodal IT; a practice of managing two separate, coherent modes of IT delivery - one focused on stability and predictability, and the other on agility.

Systems of Record

Systems that are considered to be providing the truth about data elements are often called systems of record. These systems have historically evolved slowly and carefully. For example, it is crucial that a banking system accurately reflect your bank balance.

Systems of record emphasize accuracy and security.

Systems of Engagement

Many organizations have other systems that are more exploratory. These often use experimentation to solve new problems. Systems of engagement are ones that are modified regularly. Making changes quickly is prioritized over ensuring that the changes are right.

There is a perception that DevOps suit systems of engagement more than systems of record. But the lessons from high performing companies show that this just isn't the case. Sometimes, the criticality of doing things right with a system of record is an excuse for not implementing DevOps practices. Worse, given the way that applications are interconnected, an issue in a system of engagement might end up causing a problem in a system of record anyway. Both types of systems are important. While it might be easier to start with a system of engagement when first starting a DevOps Transformation, DevOps practices apply to both types of systems. The most significant outcomes often come from transforming systems of record.

Selecting Groups to Minimize Initial Resistance

Not all staff members within an organization will be receptive to the change that is required for a DevOps transformation. In discussions around continuous delivery, users are often categorized into three general buckets:

- **Canaries** who voluntarily test bleeding edge features as soon as they are available.
- **Early adopters** who voluntarily preview releases, considered more refined than the code that canary users are exposed to.
- **Users** who consume the products, after passing through canaries and early adopters.

While development and IT operations staff might generally be expected to be less conservative than users, their attitudes will also range from very conservative, to early adopters, and to those happy to work at the innovative edge.

Ideal DevOps team members

For a successful DevOps transformation, the aim is to find team members with the following characteristics:

- They already think there is a need to change.
- They have previously shown an ability to innovate.
- They are already well-respected within the organization.
- They have a broad knowledge of the organization and how it operates.
- Ideally, they already believe that DevOps practices are what is needed

Ideal target improvements

It is also important to roll out changes incrementally. There is an old saying in the industry that any successful large IT system was previously a successful small IT system. Large scale systems that are rolled out all at once, have a very poor record of success. Most fail, no matter how much support management has provided.

When starting, it is important to find an improvement goal that:

- Can be used to gain early wins.
- Is small enough to be achievable in a reasonable time-frame.
- Has benefits that are significant enough to be obvious to the organization.

This allows constant learning from rapid feedback, and the ability to recover from mistakes quickly.

✓ The aim is to build a snowball effect where each new successful outcome adds to previous successful outcomes. This will maximize the buy-in from all those affected.

Identifying Project Metrics and KPIs

Previously, the concept of shared goals was mentioned. As well as being agreed by team members, the goals needed to be specific, measurable, and time-bound. To ensure that these goals are measurable, it is important to establish (and agree upon) appropriate metrics and Key Performance Indicators (KPIs). While there is no specific list of metrics and KPIs that apply to all DevOps projects, the following are commonly used:

Faster Outcomes

- **Deployment Frequency.** Increasing the frequency of deployments is often a critical driver in DevOps projects.
- **Deployment Speed.** As well as increasing how often deployments happen, it's important to decrease the time that they take.
- **Deployment Size.** How many features, stories, and bug fixes are being deployed each time?
- **Lead Time.** How long does it take from starting on a work item, until it is deployed?

Efficiency

- **Server to Admin Ratio.** Are the projects reducing the number of administrators required for a given number of servers?
- **Staff Member to Customers Ratio.** Is it possible for less staff members to serve a given number of customers?
- **Application Usage.** How busy is the application?
- **Application Performance.** Is the application performance improving or dropping? (Based upon application metrics)?

Quality and Security

- **Deployment Failure Rates.** How often do deployments (and/or applications) fail?
- **Application Failure Rates.** How often do application failures occur, such as configuration failures, performance timeouts, etc?
- **Mean Time to Recover.** How quickly can you recover from a failure?
- **Bug Report Rates.** You don't want customers finding bugs in your code. Is the amount they are finding increasing or decreasing?
- **Test Pass Rates.** How well is your automated testing working?
- **Defect Escape Rate.** What percentage of defects are being found in production?
- **Availability.** What percentage of time is the application truly available for customers?
- **SLA Achievement.** Are you meeting your service level agreements (SLAs)?
- **Mean Time to Detection.** If there is a failure, how long does it take for it to be detected?

Culture

- **Employee Morale.** Are employees happy with the transformation and where the organization is heading? Are they still willing to respond to further changes? (This can be very difficult to measure)
 - **Retention Rates.** Is the organization losing staff?
- ✓ It is important to choose metrics that focus on specific business outcomes, and that achieve a return on investment and increased business value.

Team Structures

Lesson Overview

This lesson covers the following topics related to team structures:

- Agile Development Practices Defined
- Principles of Agile Development
- Creating Organizational Structures for Agile Practices
- Mentoring Team Members on Agile Practices
- Enabling In-Team and Cross-Team Collaboration
- Selecting Tools and Processes for Agile Practices

Waterfall

Traditional software development practices involve determining a problem to be solved, analyzing the requirements, building and testing the required code, and then delivering the outcome to users. This is often referred to as a waterfall approach. The waterfall Model follows a sequential order; a project development team only moves to the next phase of development or testing if the previous step is completed successfully. It's what an engineer would do when building a bridge or a building. So, it might seem appropriate for software projects as well.

However, the waterfall methodology has some drawbacks. One, in particular, relates to the customer requirements. Even if a customer's requirements are defined very accurately at the start of a project, because these projects often take a long time, by delivery, the outcome may no longer match what the customer needs. There's a real challenge with the gathering of customer requirements in the first place. Even if you built exactly what the customer asked for, it'll often be different to what they need. Customers often don't know what they want until they see it or are unable to articulate what they need.

Agile

By comparison, Agile methodology emphasizes constantly adaptive planning and early delivery with continual improvement. Rather than restricting development to rigid specifications, it encourages rapid and flexible responses to change as they occur. In 2001, a group of highly regarded developers published a manifesto for Agile software development. They said that development needs to favor individuals and interactions over processes and tools, working software over comprehensive documentation, customer collaboration over contract negotiation, and responding to changes over following a plan. Agile software development methods are based on releases and iterations. One release might consist of several iterations. Each iteration is similar to a very small independent project and after being estimated and prioritized, features, bug fixes and enhancements and refactoring work is assigned to a release, and then assigned again to a specific iteration within the release, generally on a priority basis. At the end of each iteration, they should be tested working code. In each iteration, the team must focus on the outcomes of the

previous iteration and learn from that.

An advantage of having teams focused on shorter term outcomes is that teams are also less likely to waste time over engineering features or allowing an unnecessary scope creep to occur. Agile software development helps teams keep focused on business outcomes.

Comparison of Waterfall and Agile Methodologies

Waterfall	Agile
divided into distinct phases	separates the project development lifecycle into sprints
can be quite rigid	known for flexibility
all project development phases, such as design, development, and test, are completed once	follows an iterative development approach, so each phase may appear more than once
define requirements at start of project with little change expected	requirements are expected to change and evolve
focus on completing the project	focus on meeting customer's demands

Principles of Agile Development

The **Agile Alliance**¹ says that its mission is to support people who explore and apply agile values, principles, and practices to make building software solutions more effective, humane, and sustainable.

They have published a **Manifesto for Agile Software Development**².

From that, they have distilled the **12 Principles Behind the Agile Manifesto**³.

1. Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
2. Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
3. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
4. Business people and developers must work together daily throughout the project.
5. Build projects around motivated individuals. Give them the environment and support they need and trust them to get the job done.
6. The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
7. Working software is the primary measure of progress.
8. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
9. Continuous attention to technical excellence and good design enhances agility.
10. Simplicity—the art of maximizing the amount of work not done—is essential.
11. The best architectures, requirements, and designs emerge from self-organizing teams.

¹ <https://www.agilealliance.org/>

² <https://www.agilealliance.org/agile101/the-agile-manifesto/>

³ <https://www.agilealliance.org/agile101/12-principles-behind-the-agile-manifesto/>

12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

Creating Organizational Structures for Agile Practices

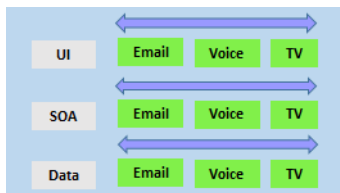
For most organizations, reorganizing to be agile is very difficult. It requires a mind-shift and a culture-shift that challenges many existing policies and processes within the organization.

This isn't surprising because good governance in organizations, particularly in large organizations often ends up leading to a large number of quite rigid rules, operating structures, and methods. It also tends to avoid wide delegation of authority.

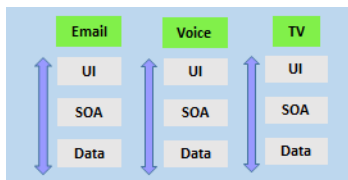
While most large organizations haven't moved to an agile structure, most are now experimenting with doing so. Their business environments are volatile and complex and they have seen the limitations of their current structures, particularly in regard to an inability to cope with change fast enough. They realize that it is common today for long-term established businesses and their industries, to be disrupted by startups.

Horizontal vs vertical teams

Traditionally, horizontal team structures divide teams according to the software architecture. In this example, the teams have been divided into user interface, service-oriented architecture, and data teams:

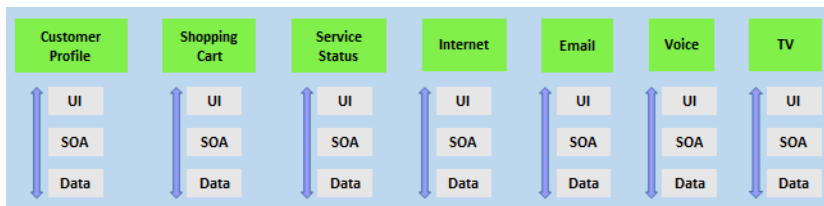


By comparison, vertical team structures span the architecture and are aligned with product outcomes:



Vertical teams have been shown to provide stronger outcomes in Agile projects. It's important that each product has a clearly-identified owner.

Another key benefit of the vertical team structure is that scaling can occur by adding teams. In this example, feature teams have been created rather than just project teams:



Mentoring Team Members on Agile Practices

When they first start an Agile transformation, many teams hire external coaches or mentors. Agile coaches help teams or individuals to adopt

Agile methods or to improve the current methods and practices. They must be agents of change by helping people to understand how they work and

encouraging them to adopt new methods. Agile coaches typically work with more than one team, and they try to remove any roadblocks from

inside or outside the organization. Doing this needs a variety of skills apart from just coaching and mentoring.

They'll need skills in teaching and facilitating. Agile coaches must be both trainers and consultants.

While it's desirable to have formal Agile training for staff members, no matter how good any Agile course is,

there's a world of difference between learning a concept within a few days and putting it into practice.

There is more than one type of Agile coach. Some coaches are technical experts who aim to show staff members how to apply specific concepts, like test-driven development and the implementation of continuous

integration or deployment. These coaches might perform peer programming sessions with staff members.

Other coaches are focused on Agile processes, determining requirements, and managing work activities.

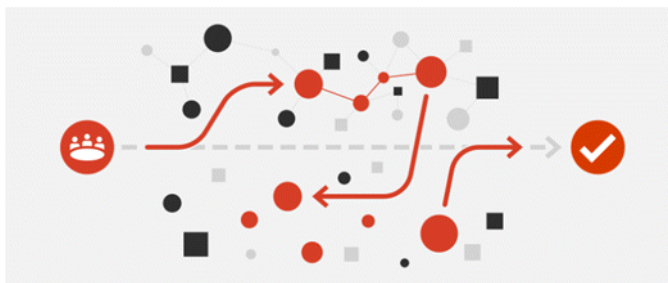
They might assist in how to run effective stand-up and review meetings. Some coaches may themselves act as scrum masters. They might mentor staff in how to fill these roles. Over time, though, it's important for

team members to develop an ability to mentor each other. Teams should aim to be self-organizing.

Team members are often expected to learn as they work and to acquire

skills from each other. To make this effective, though, the work itself needs to be done in a collaborative way, not by individuals working by themselves.

Enabling In-Team and Cross-Team Collaboration



Effective collaboration is critical for well-functioning Agile teams. Enabling this requires both cultural changes, cross-functional team collaboration, and tooling.

Cultural changes

Over recent decades, offices have often become open-spaces with few walls. Ironically, this can limit collaboration and ambient noise and distractions often also reduce productivity. Staff tend to work better when they have quiet comfortable working environments. Defined meeting times and locations lets staff choose when they want to interact with others.

Asynchronous communication should be encouraged but there should not be an expectation that all communications will be responded to urgently. Staff should be able to focus on their primary tasks without feeling like they are being left out of important decisions.

All meetings should have strict time-frames, and more importantly, have an agenda. If there is no agenda, there should be no meeting.

As it is becoming harder to find the required staff, great teams will be just as comfortable with remote or work-from-home workers as they are for those in the office. To make this successful though, collaboration via communication should become part of the organization's DNA.

Staff should be encouraged to communicate openly and frankly. Learning to deal with conflict is important for any team, as there will be disagreements at some point. Mediation skills training would be useful.

Cross-functional teams

It is clear that members of a team need to have good collaboration, it's also important to have great collaboration with wider teams, to bring people with different functional expertise together to work toward a common goal. Often, these will be people from different departments within an organization.

Faster and better innovation can occur in these cross-functional teams. People from different areas of the organization will have different views of the same problem, and they are more likely to come up with alternate solutions to problems or challenges. Existing entrenched ideas are more likely to be challenged.

Cross-functional teams can also minimize turf-wars within organizations. The more widely that a project appears to have ownership, the easier it will be for it to be widely accepted. Bringing cross-functional teams together also helps to spread knowledge across an organization.

Recognizing and rewarding collective behavior across cross-functional teams can also help to increase team cohesion.

Collaboration tooling

The following collaboration tools are commonly used by agile teams:

Skype (Microsoft)⁴. Some teams use Skype widely as it is a very commonly-deployed application. Because it is designed for general purpose communication, it is easy to enlist people from outside the team such as external stakeholders. Many will already have Skype installed. It does not currently offer more advanced team collaboration tooling.

Slack⁵. Is a very commonly used tool for collaboration in Agile and DevOps teams. From a single interface, it provides a series of separate communication channels. These can be organized by project, team, or topic. Conversations are retained and are searchable. It is very easy to add both internal and external team members. Slack directly integrates with many third party tools like **GitHub**⁶ for source code and **DropBox**⁷ for document and file storage.

Teams (Microsoft)⁸. Is a group chat application from Microsoft. It provides a combined location with workplace chat, meetings, notes, and storage of file attachments. A user can be a member of many teams.

⁴ <https://www.skype.com/en/>

⁵ <https://slack.com/>

⁶ <https://github.com/>

⁷ <https://dropbox.com/>

⁸ <https://products.office.com/en-us/microsoft-teams/group-chat-software>

Google Hangouts⁹ - Hangouts is another commonly used tool that includes which messaging, video chat, messaging and voice over IP (VOIP) features. It allows conversations between two or more users and saves chat histories online.

✓ Other common tools that include collaboration offerings include WebEx, GoToMeeting, FlowDock, Asana, ProofHub, RedBooth, Trello, DaPulse, and many others.

Selecting Tools and Processes for Agile Practices

While developing using Agile methods doesn't require specific tooling, the use of tools can often enhance the outcomes achieved. It's important to realize though, that the most important tool for Agile development is the process itself. You should become familiar with the processes that you need to follow, before you try to work out how to implement tools. Several categories of tools are commonly used.

Physical tools

Note that not all tools need to be digital tools. Many teams make extensive use of white boards for collaborating on ideas, index cards for recording stories, and sticky notes for moving tasks around. Even when digital tools are available, it might be more convenient to use these physical tools during stand up and other meetings.

Collaboration tools

These tools were discussed in the previous topic.

Project management tools

These tools usually include project planning and execution monitoring abilities (including how to respond to impediments), automation for stand up meetings, management and tracking of releases, and a way to record and work with the outcomes of retrospectives. Many include Kanban boards and detailed sprint planning options.

Most of these tools will also provide detailed visualizations, often as a graphic dashboard that shows team progress against assigned goals and targets. Some tools also integrate directly with code repositories and CI/CD tools and add code-related metrics including quality metrics, along with direct support for code reviews.



As well as a complete CI/CD system, Azure DevOps includes flexible Kanban boards, traceability through Backlogs, customizable dashboards, built-in scrum boards and integrates directly with code repositories. Code changes can be linked directly to tasks or bugs.

Apart from Azure DevOps, other common tools include Jira Agile, Trello, Active Collab, Agilo for Scrum, SpiraTeam, Icescrum, SprintGround, Gravity, Taiga, VersionOne, Agilean, Wrike, Axosoft, Assembla, PlanBox, Asana, Binfire, Proggio, VivifyScrum, and many others.

⁹ <https://hangouts.google.com/>

Screen recording tools

It might seem odd to add screen recording tools into this list but they are really helpful when working with remote team members, for recording bugs in action, and for building walkthroughs and tutorials that demonstrate actual or potential features.

There is a screen recorder built into Windows but other common ones include SnagIt, Camtasia, OBS, and Loom.

Module 1 Lab and Review Questions

Module 1 Lab



In this lab, **Agile Planning and Portfolio Management with Azure Boards¹⁰**, you will learn about the agile planning and portfolio management tools and processes provided by Azure Boards and how they can help you quickly plan, manage, and track work across your entire team. You will explore the product backlog, sprint backlog, and task boards which can be used to track the flow of work during the course of an iteration. We will also take a look at how the tools have been enhanced in this release to scale for larger teams and organizations. Tasks include:

1. Working with teams, areas, and iterations.
2. Working with work items.
3. Managing sprints and capacity.
4. Customizing Kanban Boards.
5. Defining dashboards.
6. Customizing team processes.

✓ Note that you must have already completed the Lab Environment Setup in the Welcome section.

Module 1 Review Questions

Systems of Record and Engagement

Would a system that manages inventory in a warehouse be considered a System of Record, or a System of Engagement?

Suggested Answer

System of Record. Systems that are providing the truth about data elements are often called Systems of Record.

Agile Tools

¹⁰ <https://www.azuredevopslabs.com/labs/azuredevops/agile/>

An Agile tool is used to manage and visualize work by showing tasks moving from left to right across columns representing stages. What is this tool commonly called?

Suggested Answer

Kanban Board. A Kanban Board lets you visualize the flow of work and constrain the amount of work in progress. Your Kanban board turns your backlog into an interactive signboard, providing a visual flow of work.

Lead Time Measuring

As a project metric, what is Lead Time measuring?

Suggested Answer

Lead Time Measuring is how long it takes from starting on a work item, until it is deployed.

Cross Functional Teams

What is a cross-functional team?

Suggested Answer

A team that brings people with different functional expertise, and often from different departments, together to work toward a common goal.

Technical Debt

Would you find large amounts of technical debt in a Greenfield project or a Brownfield project?

Suggested Answer

Brownfield Project. A Brownfield Project comes with the baggage of existing code bases, existing teams, and often a great amount of technical debt, they can still be ideal projects for DevOps transformations.



Module 2 Planning for Quality and Security

Planning a Quality Strategy

Lesson Overview

This lesson covers the following topics related to planning a quality strategy:

- Measuring and Managing Quality Metrics
- Feature Flags Defined
- Planning Feature Flag Lifecycles
- Technical Debt Defined
- Measuring and Managing Technical Debt
- Planning Effective Code Reviews
- Planning Performance Testing

Measuring and Managing Quality Metrics

One of the promises of DevOps is to deliver software both faster and with higher quality. Previously, these two metrics have been almost opposites. The faster you went, the lower the quality. The higher the quality, the longer it took. But DevOps processes can help you to find problems earlier, and this usually means that they take less time to fix.

Common quality-related metrics

In the previous module, we've mentioned some general project metrics and KPIs. The following is a list of metrics that directly relate to the quality of both the code being produced, and of the build and deployment processes.

- **Failed Builds Percentage** - Overall, what percentage of builds are failing?
- **Failed Deployments Percentage** - Overall, what percentage of deployments are failing?
- **Ticket Volume** - What is the overall volume of customer and/or bug tickets?
- **Bug Bounce Percentage** - What percentage of customer or bug tickets are being re-opened?

- **Unplanned Work Percentage** - What percentage of the overall work being performed is unplanned?

Feature Flags Defined

Feature Flags allow you to change how our system works without making changes to the code. Only a small configuration change is required.

In many cases, this will also only be for a small number of users. Feature Flags offer a solution to the need to push new code into trunk and have it deployed, but not have it functional yet. They are commonly implemented as the value of variables that are used to control conditional logic.

Imagine that your team are all working in the main trunk branch of a banking application. You've decided it's worth trying

to have all the work done in the main branch to avoid messy operations of merge later, but you need to make sure

that substantial changes to how the interest calculations work can happen, and people depend on that code everyday.

Worse, the changes will take you weeks to complete. You can't leave the main code broken for that period of time.

A Feature Flag could help you get around this. You can change the code so that other users who don't have

the Feature Flag set will keep using the original interest calculation code, and the members of your team who are

working on the new interest calculations and who have the Feature Flag set see the code that's been created. This is an example of

a business Feature Flag that's used to determine business logic. The other type of Feature Flag is a Release Flag.

Now, imagine that after you complete the work on the interest calculation code, you're perhaps nervous about publishing a new code out to all users at once. You have a group of users who are better at dealing with new code and issues if they arise, and these people are often called Canaries. The name is based on the old use

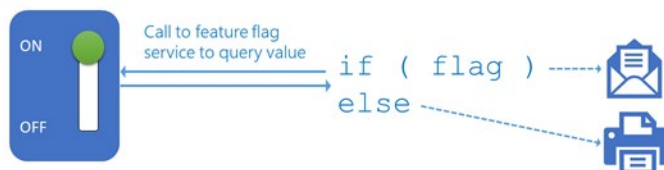
of Canaries in coal mines. You change the configuration, so that the Canary users also have the Feature Flag set and they will

start to test the new code as well. If problems occur, you can quickly disable the flag for them again.

Another release flag might be used for AB testing. Perhaps you want to find out if a new feature makes it faster for users to complete a task. You could have half the users working with the original version of the code and the other half of the users working with the new version of the code. You can then directly compare the outcome and decide if

the feature is worth keeping. Note that Feature Flags are sometimes called Feature Toggles instead.

Planning Feature Flag Lifecycles



While feature flags can be really useful, they can also introduce many issues of their own.

As soon as you introduce a feature flag, you have actually added to your overall technical debt. Just like other technical debt, they are easy to add but the longer they are part of your code, the bigger the technical debt becomes, because you've added scaffolding logic that's needed for the branching within the code.

The cyclomatic complexity of your code keeps increasing as you add more feature flags, as the number of possible paths through the code increases.

Using feature flags can make your code less solid and can also add these issues:

- The code is harder to test effectively as the number of logical combinations increases
- The code is harder to maintain because it's more complex
- The code might even be less secure
- It can be harder to duplicate problems when they are found

Because of this, feature flags need to be removed as soon as they aren't needed for the reason they were added. A plan for managing the lifecycle of feature flags is critical. As soon as you add a flag, you need to plan for when it will be removed.

Feature flags shouldn't be repurposed. There have been high profile failures that have occurred because teams decided to reuse an old flag they thought was no longer part of the code, for a new purpose.

Tooling for release flag management

The amount of effort required to manage feature flags should not be underestimated. It's important to consider using tooling that tracks:

- Which flags exist
- Which flags are enabled in which environments, situations, or target customer categories
- The plan for when the flags will be used in production
- The plan for when the flags will be removed

Using a feature flag management system lets you get the benefits of feature flags while minimizing the risk of increasing your technical debt too high.

Technical Debt Defined

Technical debt: a term that describes the future cost that will be incurred by choosing an easy solution today instead of using better practices because they would take longer to complete

The term technical debt was chosen for its comparison to financial debt. It's common for people in financial debt to make decisions that seem appropriate or the only option at the time, but in so doing, interest accrues. The more interest that accrues, the harder it is for them in the future and the less options that are available to them later. With financial debt, soon interest accrues on interest, creating a snowball effect. Similarly, technical debt can build up to the point where developers are spending almost all their time sorting out problems and doing rework, either planned or unplanned, rather than adding value.

So, how does this happen? The most common excuse is tight deadlines. When developers are forced to create code quickly, they'll often take shortcuts. As an example, instead of refactoring a method to include new functionality, let's just copy to create a new version of it. Then I only test my new code and can avoid

the level of testing that might be required if I change the original method because it's used by other parts of the code.

The problem is, now I have two copies of the same code that I need to modify in the future instead of one,

and I run the risk of the logic diverging. There are many causes. For example, there might simply be a lack of technical skills and maturity among the developers or no clear product ownership or direction.

The organization might not have coding standards at all. So, the developers didn't even know what they should be producing. The developers might not have clear requirements to target.

Well, they might be subject to last minute requirement changes. Necessary refactoring work might be delayed. There might not be any code quality testing, manual or automated.

In the end, it just makes it harder and harder to deliver value to customers in

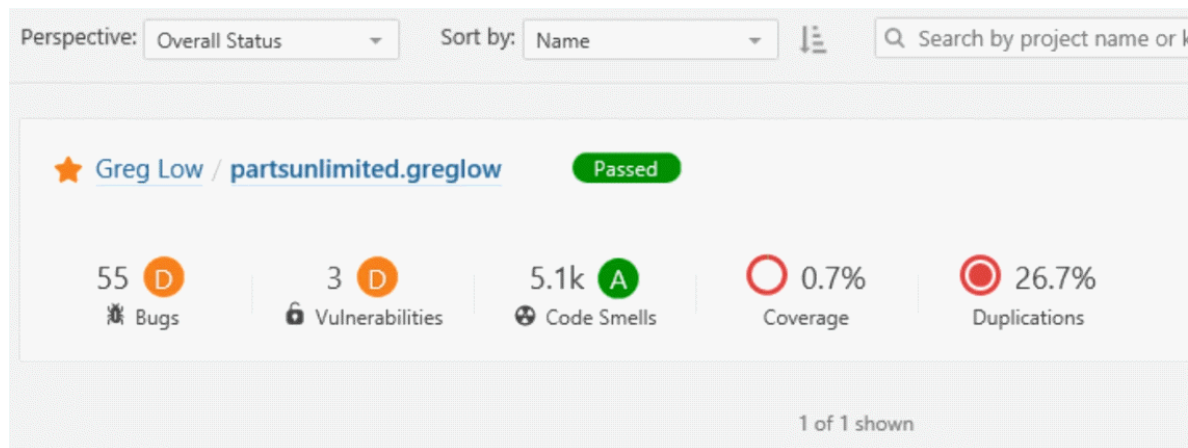
a reasonable time frame and at a reasonable cost. Technical debt is one

of the main reasons that projects fail to meet their deadlines.

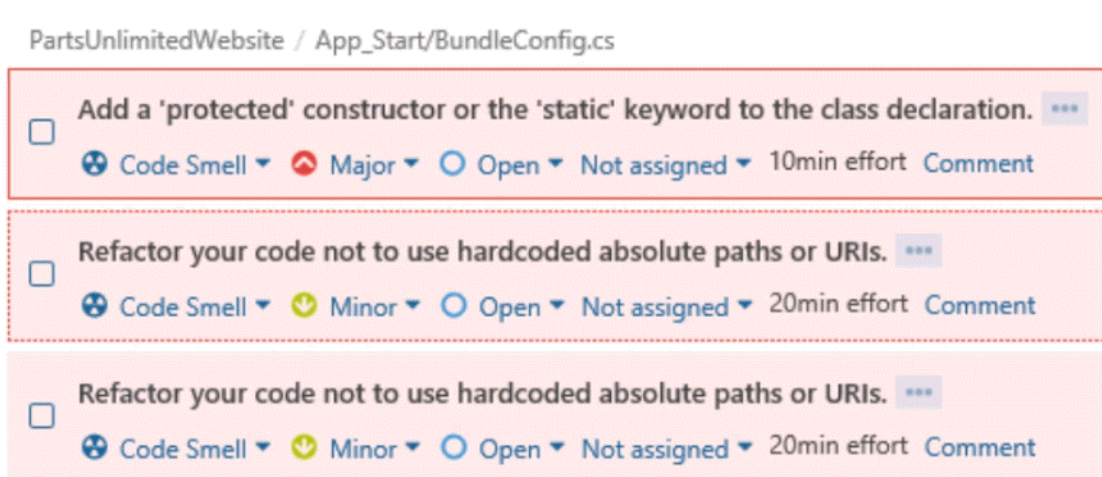
Measuring and Managing Technical Debt

It is important to integrate the assessment and measurement of technical debt and of code quality overall, as part of your Continuous Integration and Deployment pipelines in Azure DevOps.

In the Continuous Integration course in this series, we showed how to add support for SonarCloud into an Azure DevOps pipeline. After it is added and a build performed, you can see an analysis of your code:



If you drill into the issues, you can then see what the issues are, along with suggested remedies, and estimates of the time required to apply a remedy.



Planning Effective Code Reviews

Most developers would agree that code reviews can improve the quality of the applications they produce, but only if the process for the code reviews is effective.

It's important, up front, to agree that everyone is trying to achieve better code quality. Achieving code quality can seem challenging because there is no one single best way to write any piece of code, at least code with any complexity.

Everyone wants to do good work and to be proud of what they create. This means that it's easy for developers to become over-protective of their code. The organizational culture must let all involved feel that the code reviews are more like mentoring sessions where ideas about how to improve code are shared, than interrogation sessions where the aim is to identify problems and blame the author.

The knowledge sharing that can occur in mentoring-style sessions can be one of the most important outcomes of the code review process. It often happens best in small groups (perhaps even just two people), rather than in large team meetings. And it's important to highlight what has been done well, not just what needs to be improved.

Developers will often learn more in effective code review sessions than they will in any type of formal training. Reviewing code should be seen as an opportunity for all involved to learn, not just as a chore that must be completed as part of a formal process.

It's easy to see two or more people working on a problem and think that one person could have completed the task by themselves. That's a superficial view of the longer-term outcomes. Team management needs to understand that improving the code quality reduces the cost of code, not increases it. Team leaders need to establish and foster an appropriate culture across their teams.

Planning Performance Testing

Users today have close to zero patience with applications that are slow or not responding. It's critical to ensure that performance testing and load testing both happen as a routine part of building and deploying applications.

While many organizations in the past have been more concerned about how quickly they can deliver features, and that's one of the primary drivers of DevOps, it's generally more important to prioritize the customer experience. DevOps is about delivering value to customers, not just about delivering code fast.

Load testing vs performance testing

When performance testing is being discussed, it's often confused with load testing. Load testing is used to find out how a system performs with many concurrent users, large amounts of data, and over a period of time. By comparison, performance testing looks at how responsive an application is, how efficiently it uses resources, how stable and reliable it is.

Planning to implement performance testing

It's common for testing teams to focus on features, and spend time working out if an application functions as expected. There's an old saying though, that *Performance is a Feature*.

Often performance testing doesn't start until the development team thinks their code is quite stable. Performance testers should be running code as soon as new code is available. This ensures that the feedback to developers is timely and reduces the time to rectify any issues.

There are several aspects to performance testing:

- You might need to check how long user actions in an application take
- You might need to work out how much load you can put on a system before it breaks
- You might need to find out what the bottlenecks are as you increase the number of users, or the size of the data
- You might need to make sure that the application can run for long periods without having performance degrade through issues like memory leaks

Without automated testing tools, you are unlikely to be able to perform meaningful performance testing, or to integrate it within a DevOps pipeline. Often a suite of tools will be needed.

Creating a plan

When some teams are asked about their performance testing plan, it's common for teams to respond with a list of tools they are planning to use but a list of tools is not a plan. You also must work out how testing environments will be configured, you need to determine the processes to be used, and you need to determine what success or failure looks like.

- What are the expectations of the business?
- What are the expectations of the target users?
- What are the metrics you are measuring?
- Have you defined KPIs?

Performance testing needs to be part of your planning, right from the start. If you use a story or Kanban board, you might consider having an area near it where you can plan out your testing strategy. As part of the iteration planning, gaps in the testing strategy should be highlighted.

It is also important to work out how you will monitor performance once a release has been deployed, not just to measure performance beforehand.

Planning Secure Development

Lesson Overview

This lesson covers the following topics related to planning secure development:

- Secure Development Strategies Defined
- Planning to Implement OWASP Secure Coding Practices
- Inspecting and Validating Code Bases for Compliance
- Inspecting and Validating Infrastructure for Compliance

Inspecting and Validating Code Bases for Compliance

Security for applications is extremely important today. Every day, new services worldwide seem to carry stories about some company systems that have been breached. More importantly, private company and customer data

has been disclosed. This has been happening for a long time. In many cases, it wasn't visible to the public. Often, private information was disclosed, and yet the people affected were not even notified.

Governments across the world are frequently enacting legislation to require information about breaches to become

public and to require notifications to those affected.

So, what are the issues? Clearly, we need to protect information from being disclosed to people that should not have access to it. But more importantly, we need to ensure that the information isn't altered or destroyed when it shouldn't be, and we need to make sure it is destroyed when it's supposed to be. We need to make sure we properly authenticate who is accessing the data and that they have the correct permissions to do so. Through historical or archival data or logs, we need to be able to find evidence when something has gone wrong.

There are many aspects to building and deploying secure applications.

First, there is a general knowledge problem. Many developers and other staff assume they understand security, but they don't. Cybersecurity is a constantly evolving discipline. A program of ongoing education and training is essential.

Second, we need to ensure that the code is created correctly and securely implements the required features, and we need to make sure that the features were designed with security in mind in the first place. Third, we need to ensure that the application complies with the rules and regulations that is required to meet.

We need to test for this while building the code and retest periodically even after deployment. It's commonly accepted that security isn't something you can just add to an application or a system later.

Secure development must be part of every stage of the software development life cycle.

This is even more important for critical applications and those that process sensitive or highly confidential information. Application security concepts haven't been a focus for developers in the past. Apart from the education and training issues, it's because their organizations have emphasized fast development of features.

With the introduction of DevOps practices however, security testing is much easier to integrate.

Rather than being a task performed by security specialists, security testing should just be part of the day-to-day delivery processes. Overall, when the time for rework is taken into account,

adding security to your DevOps practices can reduce the overall time taken to develop quality software.

Planning to Implement OWASP Secure Coding Practices

The starting point for secure development is to use secure coding practices. The **Open Web Application Security Project (OWASP)**¹ is a global charitable organization focused on improving the security of software. OWASP's stated mission is to make software security visible, so that individuals and organizations are able to make informed decisions. They offer impartial and practical advice.

OWASP regularly publish a set of Secure Coding Practices. Their guidelines currently cover advice in the following areas:

- Input Validation
- Output Encoding
- Authentication and Password Management
- Session Management
- Access Control
- Cryptographic Practices
- Error Handling and Logging
- Data Protection
- Communication Security
- System Configuration
- Database Security
- File Management
- Memory Management
- General Coding Practices

To learn about common vulnerabilities, and to see how they appear in applications, OWASP also publishes an intentionally vulnerable web application called **The Juice Shop Tool Project**². It includes vulnerabilities from all of the **OWASP Top 10**³.

In 2002, Microsoft underwent a company-wide re-education and review phase to focus on producing secure application code. The book, **Writing Secure Code by David LeBlanc, Michael Howard**⁴, was written by two of the people involved and provides detailed advice on how to write secure code.

For more information, you can see:

The OWASP Foundation - <http://owasp.org>

OWASP Secure Coding Practices Quick Reference Guide - https://www.owasp.org/images/0/08/OWASP_SCP_Quick_Reference_Guide_v2.pdf

OWASP Code Review guide - https://www.owasp.org/images/2/2e/OWASP_Code_Review_Guide-V1_1.pdf

¹ <http://owasp.org>

² https://www.owasp.org/index.php/OWASP_Juice_Shop_Project

³ https://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project

⁴ <https://www.booktopia.com.au/ebooks/writing-secure-code-david-leblanc/prod2370006179962.html>

Inspecting and Validating Code Bases for Compliance

Automated Code Security Analysis

There are many tools for automating code security analysis. They include the following:

Micro Focus Fortify⁵ provides build tasks that can be used in Azure DevOps continuous integration builds to identify vulnerabilities in source code. It offers two styles of analysis.

- **Fortify Static Code Analyzer (SCA)** searches for violations of security-specific coding rules and guidelines. It works in a variety of languages.
- **Fortify on Demand** is a service for checking application security. The outcomes of an SCA scan are audited by a team of security experts, including the use of Fortify WebInspect for automated dynamic scanning.

Checkmarx CxSAST⁶ is a solution for Static Source Code Analysis (SAST) and Open Source Analysis (OSA) designed for identifying, tracking and fixing technical and logical security flaws.

It is designed to be integrated into Azure DevOps pipelines and allows for early detection and mitigation of crucial security flaws. To improve performance, it is capable of incremental scanning (ie: just checking the code recently altered or introduced).

BinSkim⁷ is a static analysis tool that scans binary files. BinSkim replaces an earlier Microsoft tool called BinScope. In particular, it checks that the executable produced (ie: a Windows PE formatted file) has opted into all of the binary mitigations offered by the Windows Platform, including:

- SafeSEH is enabled for safe exception handling
- ASLR is enabled so that memory is not laid out in a predictable fashion
- Stack Protection is enabled to prevent overflow

OWASP Zed Attack Proxy Scan. Also known as **OWASP ZAP** Scan is an open-source web application security scanner that is intended for users with all levels of security knowledge. It can be used by professional penetration testers.

Inspecting and Validating Infrastructure for Compliance

Early attempts at automating the creation of virtual machines and entire environments focused on writing procedural code in languages like PowerShell. The problem with this type of coding was that it was hard to write idempotent code ie: code that produces the same outcome no matter how many times you run it. **PowerShell Desired State Configuration (DSC)⁸** was a big step forward because it allowed you to define the desired outcome instead of the process for achieving that outcome. It was a declarative mechanism rather than a procedural one.

Today, a variety of tools allow you to define infrastructure as code, and these have already created significant benefits to DevOps teams. Reliable environments (including operating systems and services) can be created on demand, and in a completely repeatable way. The same code can be run again to ensure that the target state is present, and to remove any drift from that state.

⁵ <https://marketplace.visualstudio.com/items?itemName=fortifyvsts.hpe-security-fortify-vsts>

⁶ <https://marketplace.visualstudio.com/items?itemName=checkmarx.cxsast>

⁷ <https://blogs.msdn.microsoft.com/secdevblog/2016/08/17/introducing-binskim/>

⁸ <https://docs.microsoft.com/en-us/powershell/scripting/dsc/overview/overview?view=powershell-6>

Rather than just configuration of the environments though, there is a need to validate the environments, particularly in relation to organizational or regulatory policies that might relate to them.

Validation tooling

Azure Automation State Configuration⁹ builds on PowerShell DSC to make management easier. It allows you to author and manage DSC Configurations, import DSC Resources, and generate DSC Node Configurations (MOF documents), all in the cloud. These items are placed on an Automation server so that target nodes in the cloud or on-premises can retrieve them, and automatically conform to the desired state, and report back on their compliance.

As an example of alternate tooling, **InSpec**¹⁰ is an open-source testing framework for infrastructure with a language for specifying compliance, security and other policy requirements. It works by comparing the actual state of your system with the desired state that you express in InSpec code. It then detects violations and displays findings in the form of a report. The remediation is then done by the administrator.

⁹ <https://docs.microsoft.com/en-us/azure/automation/automation-dsc-overview>

¹⁰ <https://www.inspec.io>

Module 2 Lab and Review Questions

Module 2 Lab



LaunchDarkly is a continuous delivery platform that provides feature flags as a service and allows developers to iterate quickly and safely. LaunchDarkly gives you the power to separate feature rollout from code deployment and manage feature flags at scale.

In this lab, **Feature Flag Management with LaunchDarkly and AzureDevOps¹¹**, you will investigate the use of feature flags and learn how to:

- How to implement a very simple feature flag for an ASP.NET MVC application.
- How to integrate LaunchDarkly with Azure DevOps
- How to roll-out LaunchDarkly feature flags in Azure DevOps release pipelines

Module 2 Review Questions

Feature Flags

What are Feature Flags and what other name are they referred to?

Suggested Answer

Feature Flags allow you to change how a system works, without making changes to the code. Only a small configuration change is required. In many cases, this will also be only for a small number of users. Feature Flags are also known as Feature Toggles.

OWASP

Name some of the OWASP security coding practices.

Suggested Answer

Input Validation, Output Encoding, Authentication and Passwords, Session Management, Access Control, Cryptographic Practices, Error Handling and Logging, Data Protection, Communication Security, System Configuration, Database Security, File Management, Memory Management, and General Coding Practices

Technical Debt

¹¹ <https://www.azuredevopslabs.com/labs/vstsextend/launchdarkly/>

What is meant by the term Technical Debt?

Suggested Answer

Technical Debt is the The future cost that will be incurred by choosing an easy solution today, instead of using better practices because they would take longer to complete.

Canaries

What is meant by describing a user as a Canary?

Suggested Answer

A canary is a user who is better at dealing with new code and issues that might arise. Code is released to them first as an early warning of potential issues.

Data Protection

Which two types of data are the biggest concerns for data protection?

Suggested Answer

Sensitive data (personally identifiable or other private data), Confidential data (cause financial or reputational damage to an organization if disclosed).



Module 3 Migrating and Consolidating Artifacts and Tools

Migrating and Consolidating Artifacts

Lesson Overview

This lesson covers the following topics related to migrating and consolidating artifacts:

- Working with Artifact Repositories
- Migrating and Integrating Artifact Repositories
- Migrating and Integrating Source Control

Identifying Existing Artifact Repositories

An **artifact** is a deployable component of your application. Azure pipelines can work with a wide variety of artifact's

sources and repositories. When you're creating a release pipeline, you need to link the required artifact sources.

Often, this will represent the output of a build pipeline from a continuous integration system like Azure pipelines,

Jenkins, or TeamCity. The artifacts that you produce might be stored in source control, like Git or Team Foundation version control.

But you might also be using package management tools when you get repositories. When you need to create

a release though, you need to specify which version of the artifacts are required. By default, the release pipeline will

choose the latest version of the artifacts. But you might not want that. For example, you might need to choose a specific branch, a specific build version, or perhaps you need to specify tags. Azure artifacts is one of

the services that's part of Azure DevOps. Using it can eliminate the need to manage file shares or host private package service. It lets you share code easily by letting you store Maven, npm, or NuGet packages together, cloud hosted, indexed, and matched. Now, while we can do so, there's also no need

to

store your binaries in Git. You can store them directly using universal packages. This is also a great way to protect your packages. Azure artifacts provides universal artifact management from Maven, npm, and NuGet.

As well as sharing packages though, you can then easily access all of your artifacts in builds and releases because it integrates naturally with Azure pipelines and it's CI/CD tooling, along with versioning and testing.

Migrating and Integrating Artifact Repositories

While you can continue to work with your existing artifact repositories in their current locations when using Azure DevOps, there are advantages to migrating them.

NuGet and Other Packages

Azure DevOps Services provides hosted NuGet feeds as a service. By using this service, you can often eliminate the dependencies on on-premises resources such as file shares and locally hosted instances of NuGet.Server. The feeds can also be consumed by any Continuous Integration system that supports authenticated NuGet feeds.

Walkthroughs

For details on how to integrate NuGet, npm, Maven, Python, and Universal Feeds, see the following walkthroughs:

Get started with NuGet packages in Azure DevOps Services and TFS¹

Use npm to store JavaScript packages in Azure DevOps Services or TFS²

Get started with Maven packages in Azure DevOps Services and TFS³

Get started with Python packages in Azure Artifacts⁴

Publish and then download a Universal Package⁵

Steps for Migrating and Integrating Source Control

Similarly to artifacts, you might consider migrating your source control to Azure DevOps by using Azure Repos. However, moving your team from a centralized version control system to a distributed system like Git requires involves much more than learning the new commands. A successful migration to Git requires that you understand the differences in how file history and branches are handled.

The Azure DevOps team recommends the following process:

1. Evaluate the tools and processes you're using.
2. Select a branching strategy for Git.
3. Decide how to migrate history – or if you even want to.
4. Maintain your old version control system.
5. Remove binaries and executables from source control.

¹ <https://docs.microsoft.com/en-us/azure/devops/artifacts/get-started-nuget?view=vsts&tabs=new-nav>

² <https://docs.microsoft.com/en-us/azure/devops/artifacts/get-started-npm?view=vsts&tabs=new-nav%2Cwindows>

³ <https://docs.microsoft.com/en-us/azure/devops/artifacts/get-started-maven?view=vsts&tabs=new-nav>

⁴ <https://docs.microsoft.com/en-us/azure/devops/artifacts/quickstarts/python-packages?view=vsts&tabs=new-nav>

⁵ <https://docs.microsoft.com/en-us/azure/devops/artifacts/quickstarts/universal-packages?view=vsts&tabs=azuredevops>

6. Train your team in the concepts and practice of Git.
7. Perform the actual migration to Git.

Most organizations are also unable to stop work while this migration occurs, so you'll need a plan for managing that. In many cases, a **tip** migration will be used where the history remains on the previous system. While old history becomes less relevant over time, it still is usually needed for reference.

Team Foundation Server

You can bring your data from TFS into Azure DevOps using the Team Foundation Server Database Import Service. The advantage of this is that you'll still have the same work item numbers, check-in numbers, Git commit IDs, and other data. This allows you to move to a scalable, reliable, and globally available hosted service, with a constantly updated feature set.

- ✓ There is a process for **migrating from other systems to Git**⁶.

For more information, you can see:

Considerations for migrating to Git - <https://docs.microsoft.com/en-us/azure/devops/learn/git/centralized-to-git>.

Tool and migration guide - <https://www.microsoft.com/en-us/download/details.aspx?id=54274>

⁶ <https://docs.microsoft.com/en-us/azure/devops/learn/git/migrate-other-systems-to-git>

Migrating and Integrating Source Control

Lesson Overview

This lesson covers the following topics related to migrating and consolidating artifacts:

- Designing an Authorization and Access Strategy
- Migrating or Integrating Existing Work Management Tools
- Migrating or Integrating Existing Test Management Tools
- Designing a License Management Strategy

Designing an Authorization and Access Strategy

Azure DevOps Services uses enterprise-grade authentication. You can use either a Microsoft account or Azure Active Directory (AAD) to protect and secure your data. Many client applications, such as Visual Studio or Visual Studio code, natively support this indication by

other Microsoft Accounts or AAD. Eclipse can also support this if you install a Team Explorer Everywhere plug-in. When you need a non-Microsoft tool and like GIT, NuGet or Xcode to integrate directly with Azure DevOps Services and the tools don't directly support

Microsoft account or AAD account for authentication, you can still use them by setting up personal access tokens.

These tokens can be set up using GIT Credential managers or you can create them manually. Personal access tokens

are also useful when you need to establish access in command line tools, or in tools and tasks in build pipelines and when

calling REST-based APIs because you don't have a UI popping out to perform the authentication.

When access is no longer required you can then just revoke the personal access token. Azure DevOps is pre-configured

with default security groups. Default permissions are assigned to the default security groups. But you can also configure access

at the organization level, the collection level, and at the project or object level. In the organization settings

in Azure DevOps, you can configure app access policies. Based on your security policies, you might allow alternate

authentication methods, allow third party applications to access via OAuth, or even allow anonymous access to some projects.

For even tighter control, you can set conditional access to Azure DevOps. This offers simple ways to help secure

resources when using Azure Active Directory for authentication. Conditional access policies such as Multifactor Authentication can help to minimize the risk of compromised credentials. As part of a conditional

access policy you might require security group membership, a location or network identity, a specific operating system, a managed device, or other criteria.

Migrating or Integrating Existing Work Management Tools

Azure DevOps can be integrated with a variety of existing work management tools. As an example, in the Visual Studio Marketplace, Microsoft offers **Trello integration tooling**⁷.

Migrating from other work management tools to Azure DevOps takes considerable planning. Most work management tools are highly configurable by the end user. This means that there might not be a tool available that will perform the migration without further configuration.

Jira

Jira is a commonly-used work management tool.

In the Visual Studio Marketplace, **Solidify**⁸ offers a tool for Jira to Azure DevOps migration. It does the migration in two phases. Jira issues are exported to files and then the files are imported to Azure DevOps.

If you decide to try to write the migration code yourself, the following blog post provides sample code that might help you to get started:

Migrate your project from Jira to Azure DevOps⁹

Other applications

Third party organizations do offer commercial tooling to assist with migrating other work management tools like Aha, BugZilla, ClearQuest, and others to Azure DevOps.

Migrating or Integrating Existing Test Management Tools

Azure Test Plans are used to track manual testing for sprints and milestones. This allows you to track when that testing is complete.

Azure DevOps also has a Test & Feedback extension available in the Visual Studio Marketplace. The extension is used to help teams perform exploratory testing and provide feedback. All team members (developers, product owners, managers, UX or UI engineers, marketing teams, early adopters), and other stakeholders can use the extension to submit bugs or provide feedback.

Apache JMeter¹⁰ is open source software written in Java and designed to load test functional behavior and measure performance.

Pester¹¹ is a tool that can be used to automate the testing of PowerShell code.

SoapUI¹² is another testing framework for SOAP and REST testing.

✓ If you are using Microsoft Test Manager you should plan to migrate to using Azure Test Plans instead.

For more information, you can see:

Marketplace search for test management - (<https://marketplace.visualstudio.com/search?term=test%20management&target=AzureDevOps&category=All%20categories&sortBy=Relevance>)¹³

⁷ <https://marketplace.visualstudio.com/items?itemName=ms-vsts.services-trello>

⁸ <https://marketplace.visualstudio.com/items?itemName=solidify-labs.jira-devops-migration>

⁹ <http://www.azurefieldnotes.com/2018/10/01/migrate-your-project-from-jira-to-azure-devops/>

¹⁰ <https://docs.microsoft.com/en-us/azure/devops/test/load-test/get-started-jmeter-test?view=vsts>

¹¹ <https://marketplace.visualstudio.com/items?itemName=richardfennellBM.BM-VSTS-PesterRunner-Task>

¹² <https://marketplace.visualstudio.com/items?itemName=AjeetChouksey.soapui>

¹³ <https://marketplace.visualstudio.com/search?term=test%20management&target=AzureDevOps&category=All%20categories&sortBy=Relevance>

Designing a License Management Strategy

For the latest, most up-to-date pricing information, visit Azure DevOps Pricing - <https://azure.microsoft.com/en-us/pricing/details/devops/azure-devops-services/>

Module 3 Lab and Review Questions

Module 3 Lab



Azure DevOps services helps teams modernize their application development lifecycle and go from idea to deployment with continuous integration, testing, and deployment for any app targeting any platform. Azure DevOps services works with many development tool including Visual Studio, Eclipse, IntelliJ, Android Studio, XCode, etc., to make it easy for developers to use Azure DevOps.

In this lab, **Integrating Azure Repos and Azure Pipelines with Eclipse¹⁴**, you will walk you through a typical end-to-end workflow for a Java developer using Azure DevOps and working with Eclipse.

You will learn how to:

- Provision a Azure DevOps Organization team project with sample data and users
- Install Eclipse Photon and Team Explorer Everywhere, the Azure DevOps plug-in for Eclipse
- Install and explore Azure Toolkit for Eclipse
- Setup an Azure Build pipeline to build and test the code, then push it to an Azure Container Registry
- Setup an Azure Web app and configure an Azure Release pipeline to deploy the image to Azure Web App

Module 3 Review Questions

Azure DevOps Authentication Methods

What types of authentication accounts can Azure DevOps use?

Suggested Answer

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 Licensing

¹⁴ <https://www.azuredevopslabs.com/labs/vstsextend/eclipse/>

What licensing options are available for Azure DevOps?

Suggested Answer

Open source projects, Small Teams, and Teams of any size. Open source and Small teams are free. Teams of any size are currently \$30/month.

Artifacts

What is an artifact and why would you use it?

Suggested Answer

An artifact is a deployable component of your application. Azure Artifacts can eliminate the need to manage file shares or to host private package servers. Azure Artifacts provides universal artifact management for Maven, npm and NuGet.

Migration Steps

What are the steps for migrating and integrating source control?

Suggested Answer

Evaluate the tools and processes you're using. Select a branching strategy for Git. Decide how to migrate history – or if you even want to. Maintain your old version control system. Remove binaries and executables from source control. Train your team in the concepts and practice of Git. Perform the actual migration to Git.