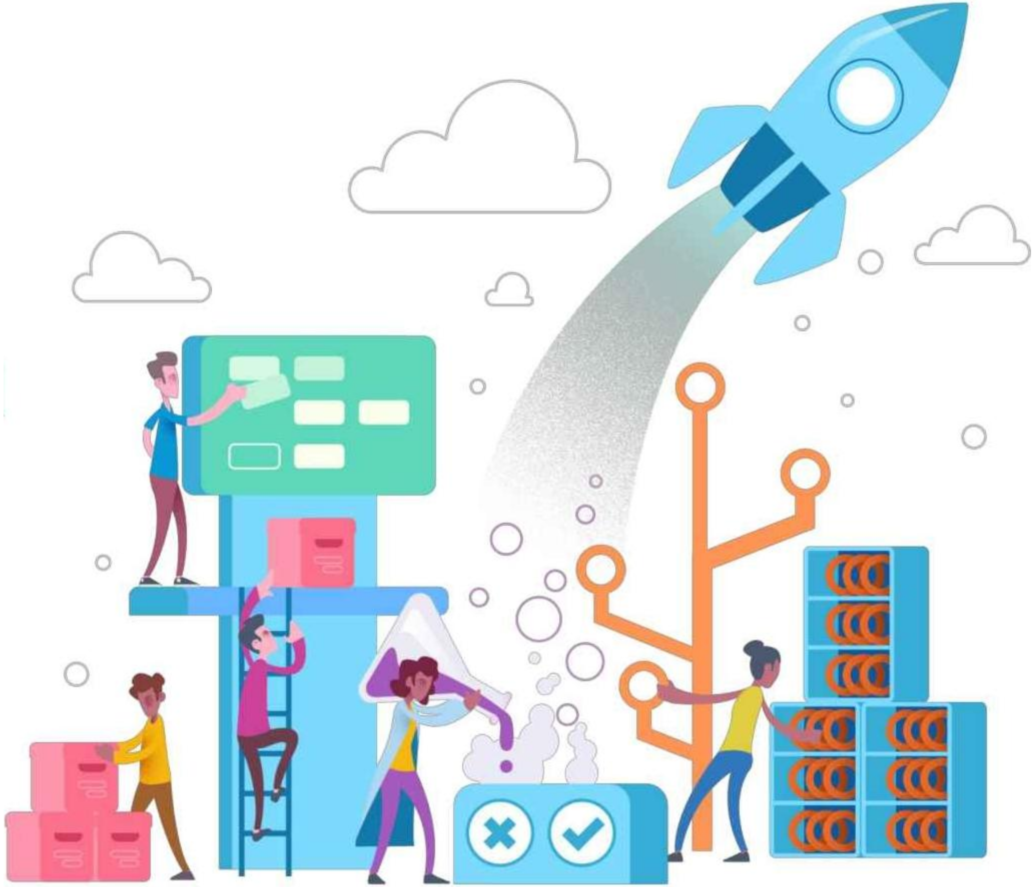
SDLC with Azure DevOps

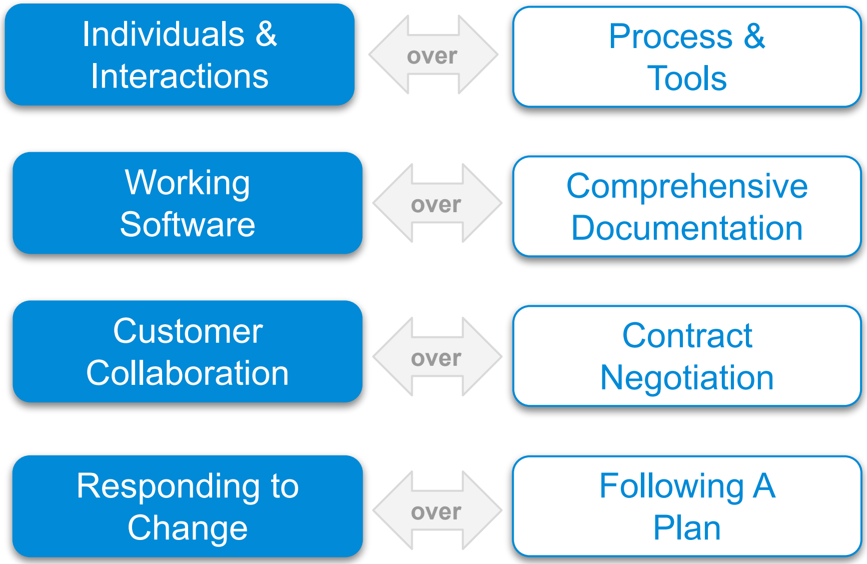
Pravin MJ



Agile Software Development

“Agile is a Software development methodology emphasizing on **incremental delivery**, **team collaboration**, **continual planning** and **continual learning.** “

Major Values



Individuals & Interactions over Process & Tools

Working Software over Comprehensive Documentation

Customer Collaboration over Contract Negotiation

Responding to Change over Following A Plan

The Agile Manifesto

Is it still relevant now, in 2020 ?

> The tenets behind the #agilemanifesto weren’t invented by us, they’re the tenets of scientific method. Galileo used them.Archimedes used them.

➡ Agile Teams should be empowered to work how they best see fit.

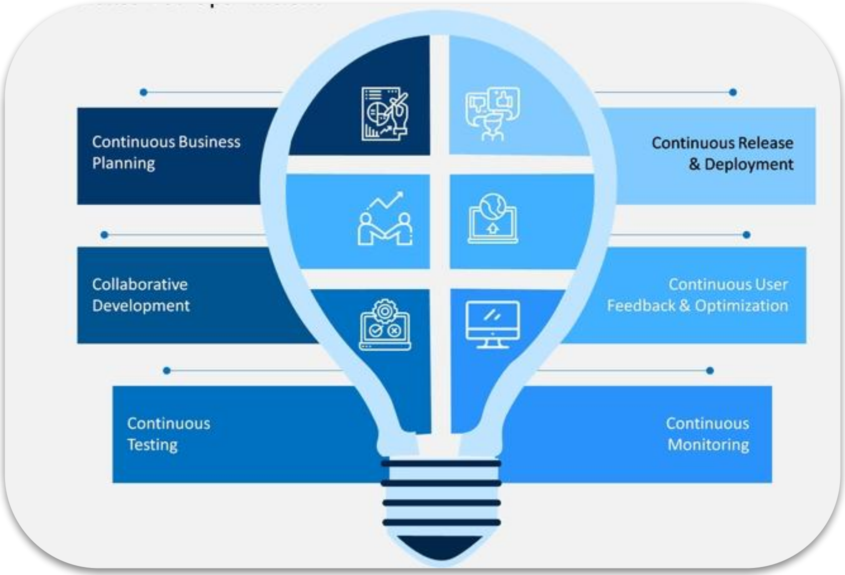
➡ Agile frameworks works alongside cultural values, but if you don’t have the cultural default, then what you could do turn out flawed from get go.

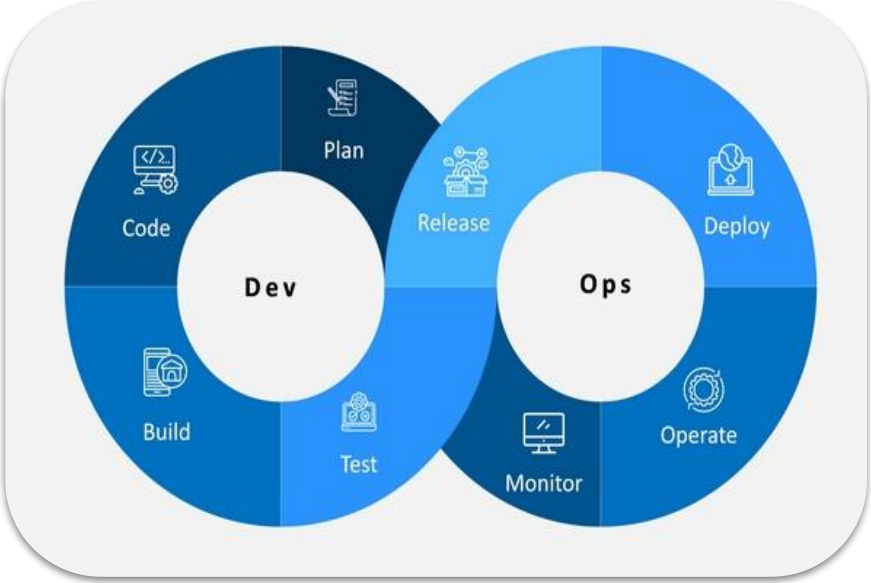
1. Early and continuous delivery of valuable software.
2. Embrace change.
3. Frequent Delivery.
4. Business & Developers Together.
5. Motivated Individuals.
6. Face-to-Face Conversations.
7. Working Software.
8. Technical Excellence.
9. Sustainable and constant development.
10. Simplicity.
11. Self Organizing Teams.
12. Regular Reflection & Adjustment.

What is DevOps?

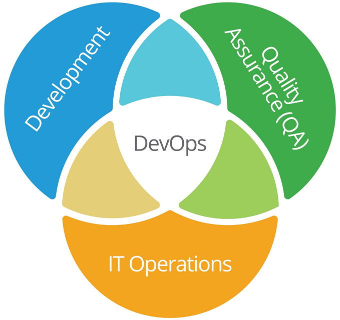
People. Process. Products

DevOps is a **combination** of **cultural philosophies**, **practices**, and **tools** that increases an organization’s ability to deliver applications and services at high velocity: evolving and improving products at a faster pace than organizations using traditional software development and infrastructure management processes.

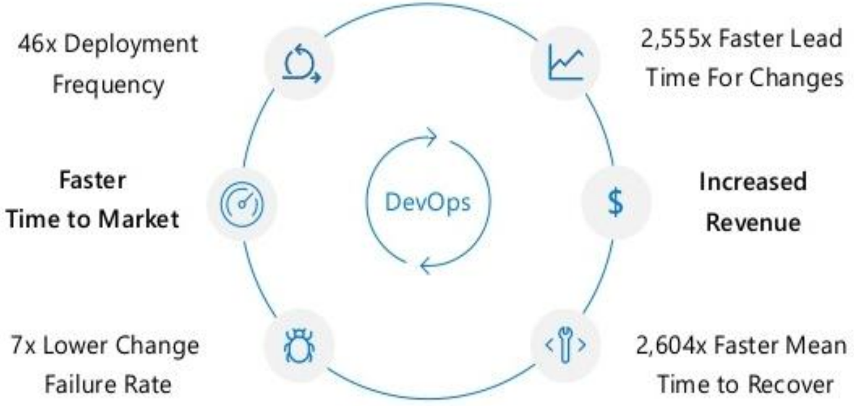

Continuous Business Planning
Continuous Release & Deployment
Collaborative Development
Continuous User Feedback & Optimization
Continuous Testing
Continuous Monitoring



DevSecOps - BizDevOp



- Philosophical, No Single approach or methods.
- Developers Powerful Tool - Automation.
- Transitions gets better with implementations
- Production-Ready state


46x Deployment Frequency
2,555x Faster Lead Time For Changes
Faster Time to Market
DevOps
Increased Revenue
7x Lower Change Failure Rate
2,604x Faster Mean Time to Recover

Product

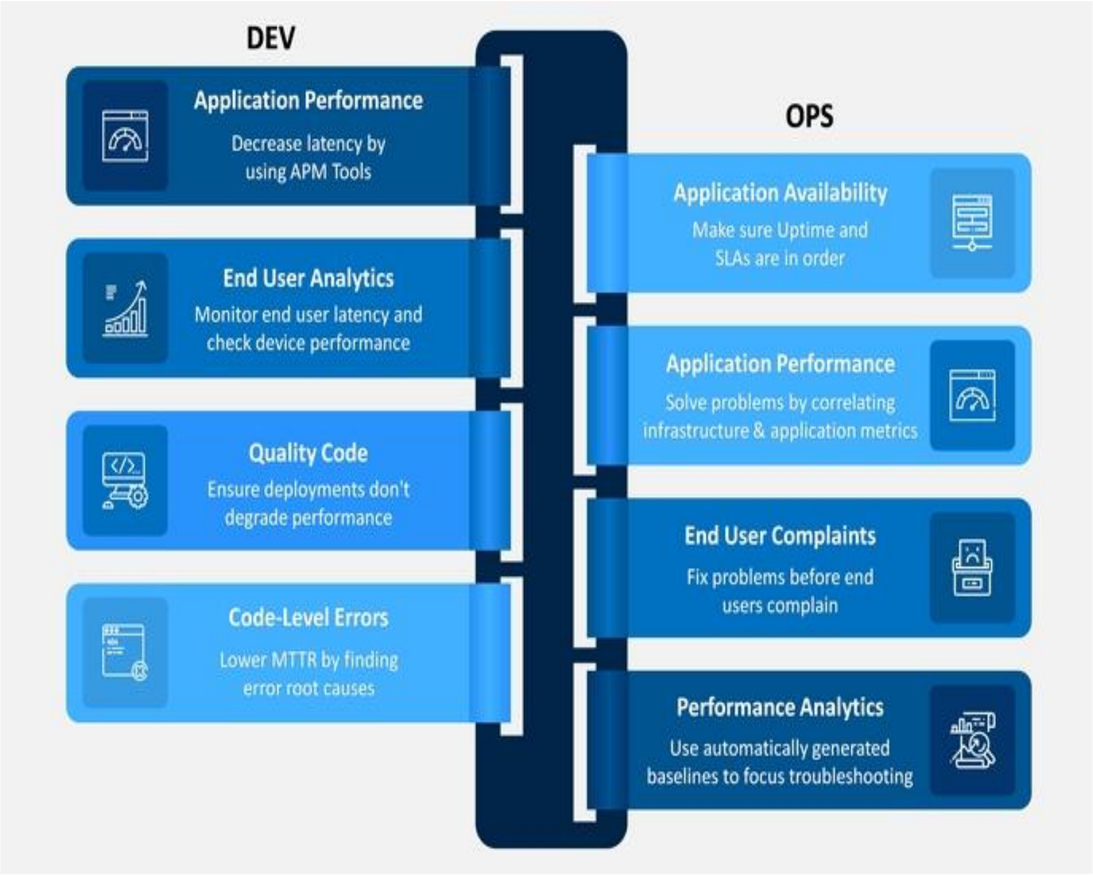
- Continuous Delivery
- Continuous Integration
- Version Control Systems
- Release Controls
- Analytics & Performance

Business

- Trust
- Cross-Functional Support
- Continuous Communication
- Feature LifeCycle Co-ordination
- Feedback Loops

Key Benefits

- Speed
- Rapid Delivery
- Reliability
- Scale
- Improved Collaboration
- Security



DEV

Application Performance
Decrease latency by using APM Tools

End User Analytics
Monitor end user latency and check device performance

Quality Code
Ensure deployments don't degrade performance

Code-Level Errors
Lower MTTR by finding error root causes

OPS

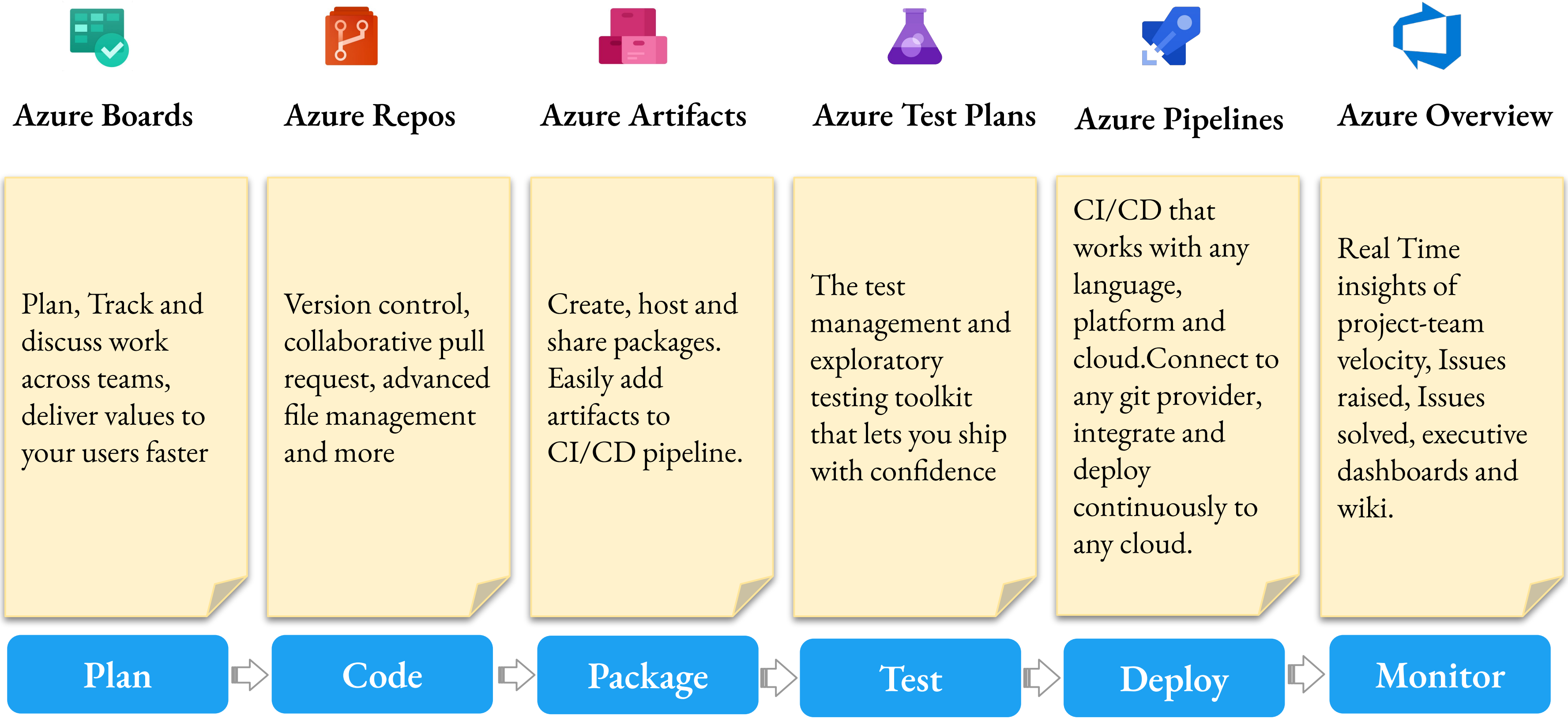
Application Availability
Make sure Uptime and SLAs are in order

Application Performance
Solve problems by correlating infrastructure & application metrics

End User Complaints
Fix problems before end users complain

Performance Analytics
Use automatically generated baselines to focus troubleshooting

Azure DevOps



Azure Boards

Plan, Track and discuss work across teams, deliver values to your users faster

Azure Repos

Version control, collaborative pull request, advanced file management and more

Azure Artifacts

Create, host and share packages. Easily add artifacts to CI/CD pipeline.

Azure Test Plans

The test management and exploratory testing toolkit that lets you ship with confidence

Azure Pipelines

CI/CD that works with any language, platform and cloud.Connect to any git provider, integrate and deploy continuously to any cloud.

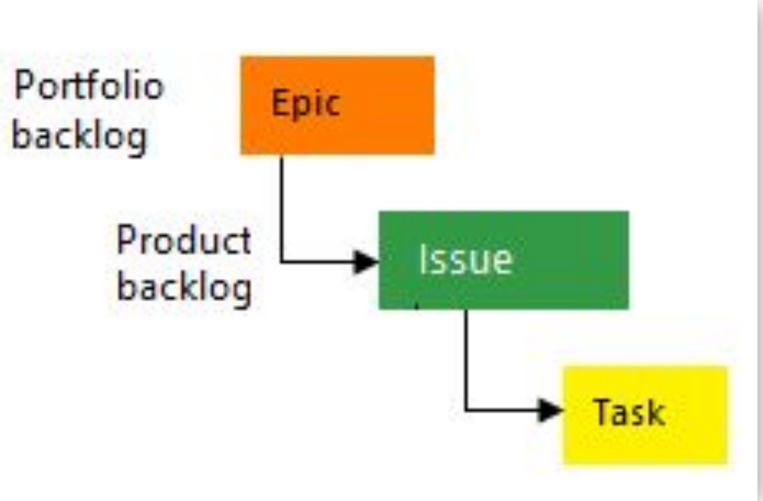
Azure Overview

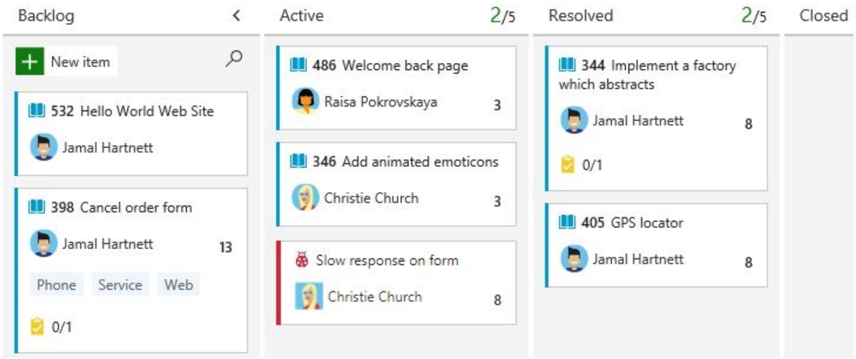
Real Time insights of project-team velocity, Issues raised, Issues solved, executive dashboards and wiki.

Plan → Code → Package → Test → Deploy → Monitor

Process - Basic

- Does not follow any formal methods.
- Simplest model that uses Issues, Tasks and Epics to track work.
- Tasks can be simple as Todo Model and Bug Tracker.




Backlog

New item

532 Hello World Web Site
Jamal Hartnett

398 Cancel order form
Jamal Hartnett 13
Phone Service Web
0/1

Active 2/5

486 Welcome back page
Raisa Pokrovskaya 3

346 Add animated emoticons
Christie Church 3

Slow response on form
Christie Church 8

Resolved 2/5

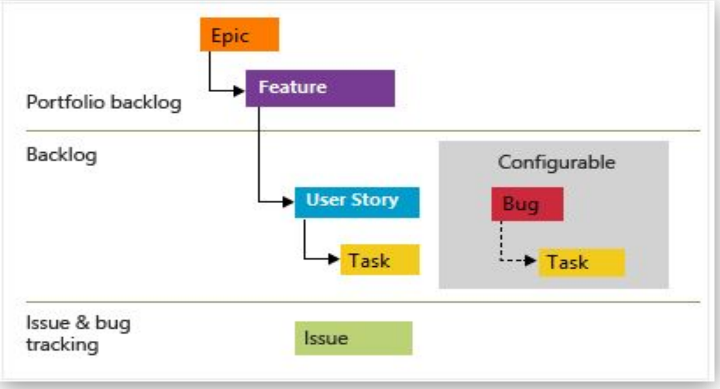
344 Implement a factory which abstracts
Jamal Hartnett 8
0/1

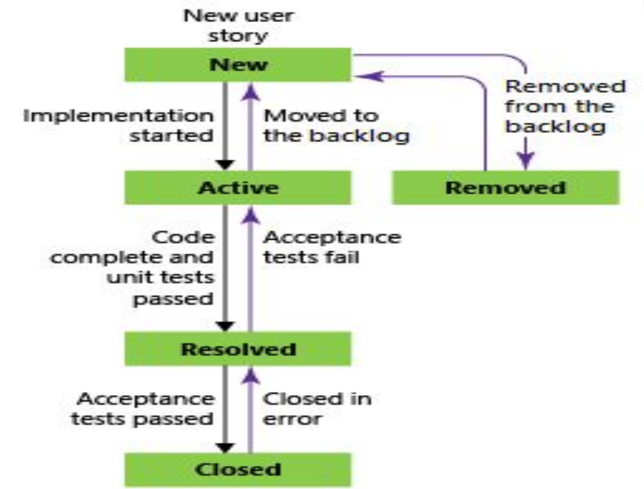
405 GPS locator
Jamal Hartnett 8

Closed

Process - Agile

- Great when your team uses Agile Planning methods for development and tests.
- Track User Stories and Bugs Separately
- Original Estimate, Remaining Work and completed work.





Business/Stakeholders

High level Epic and Feature management

Break down Features into User stories(A module)

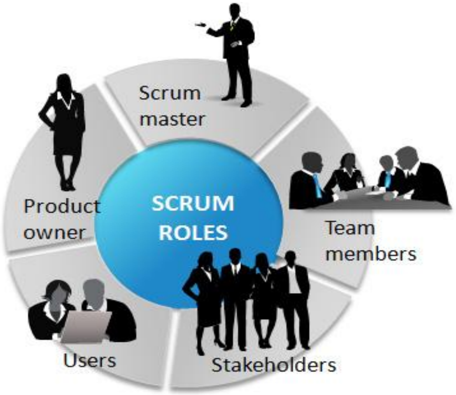
User Stories is broken down into tasks that are developed iteratively

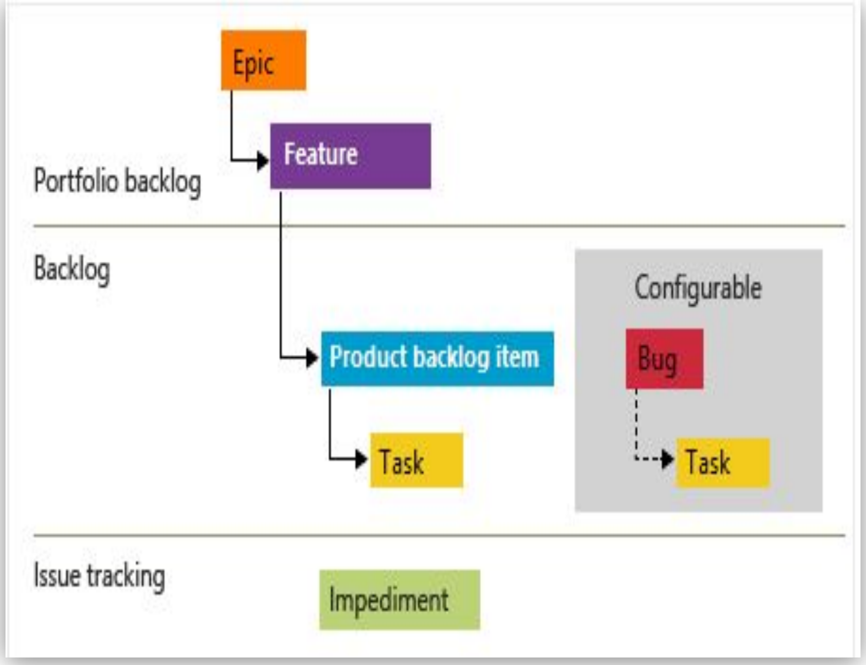
Quality Assurance Goes with development on each cycle

Separate Test suite to manage tests

Process - Scrum

- Emphasis on Scrum Process - Product backlog, Task boards, Daily Stand-Ups, Retrospectives, Definition of done, Sprint Reviews
- Product backlog Vs Sprint backlog
- “Inspect” and “Adapt” - Retrospective vs Review



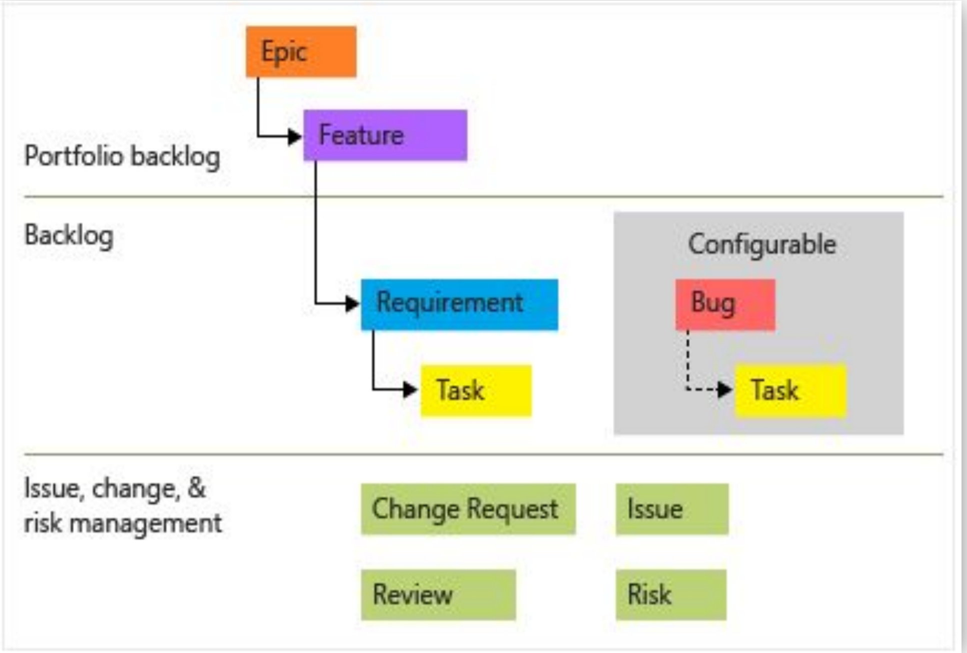


Process - CMMI

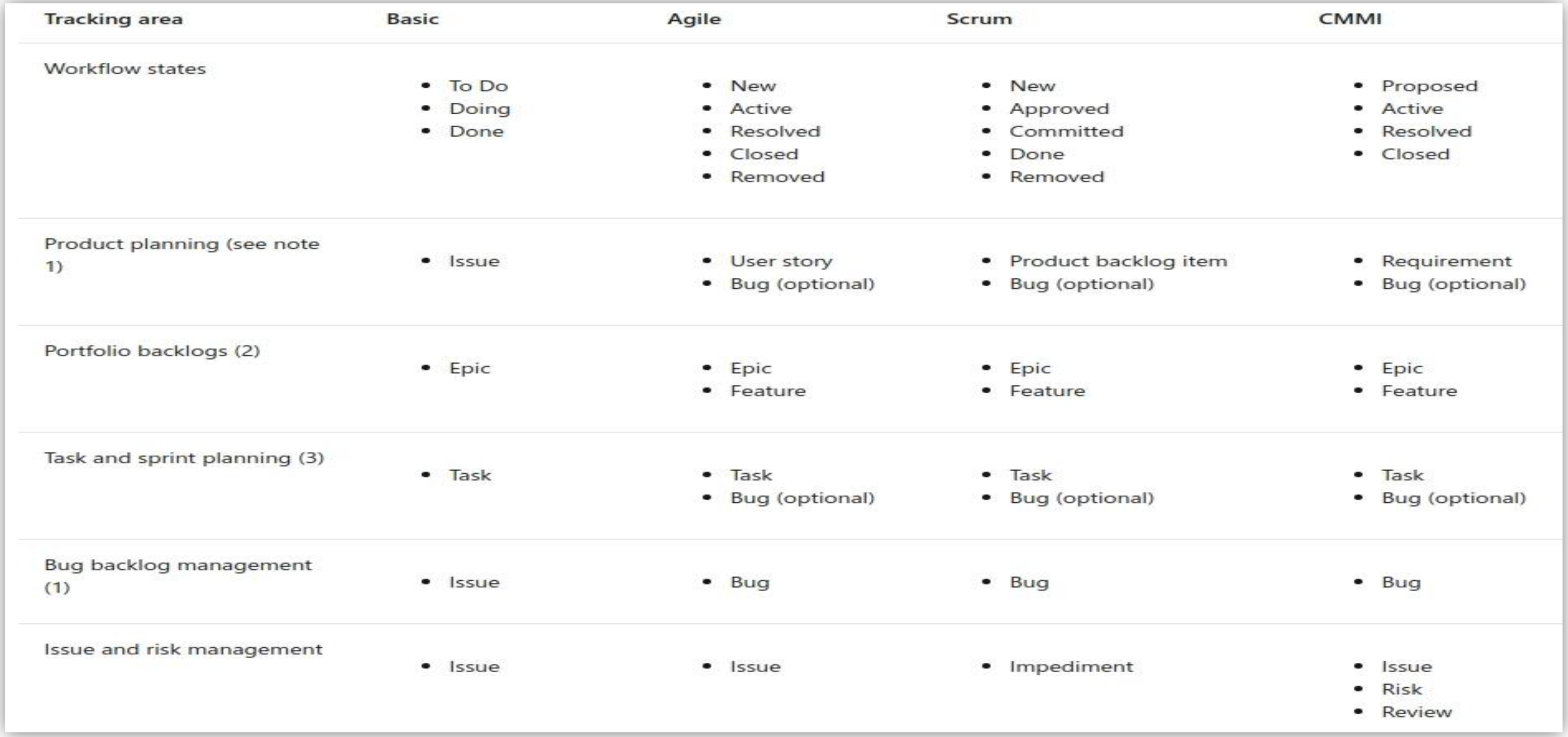
- More Formal methods of process with an auditable records of decision

- Formal change management process like CAR, IPM, OT,PMC,PP..

- Supports tracking Original Estimate, Remaining work and completed work

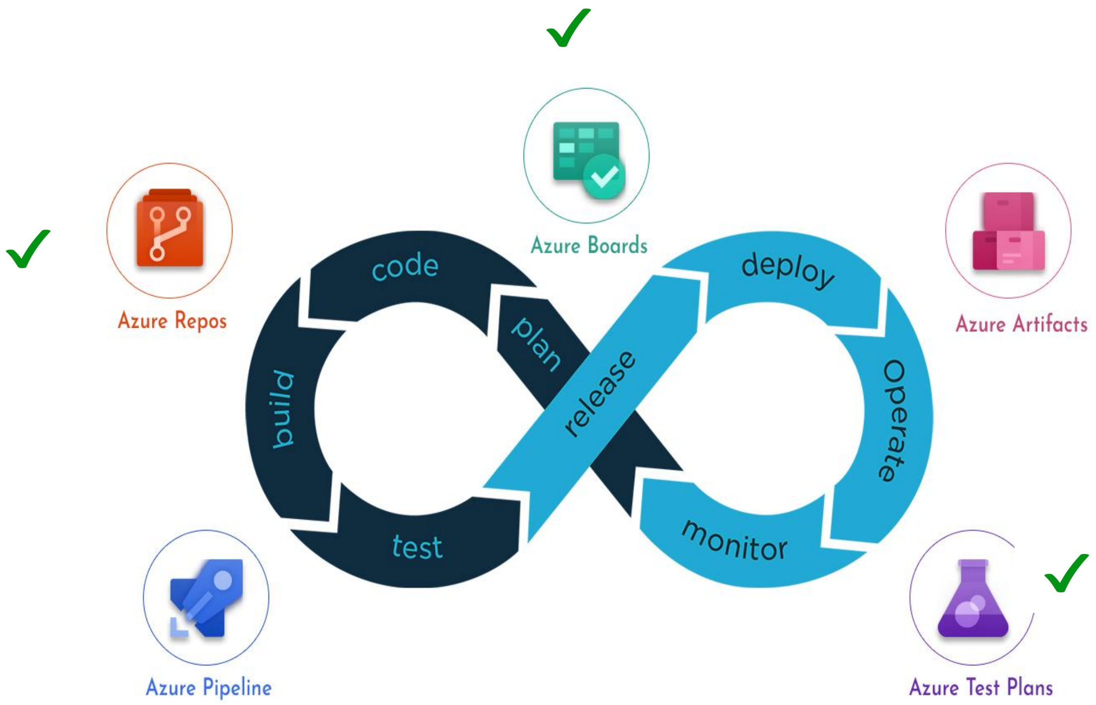


Process Comparison

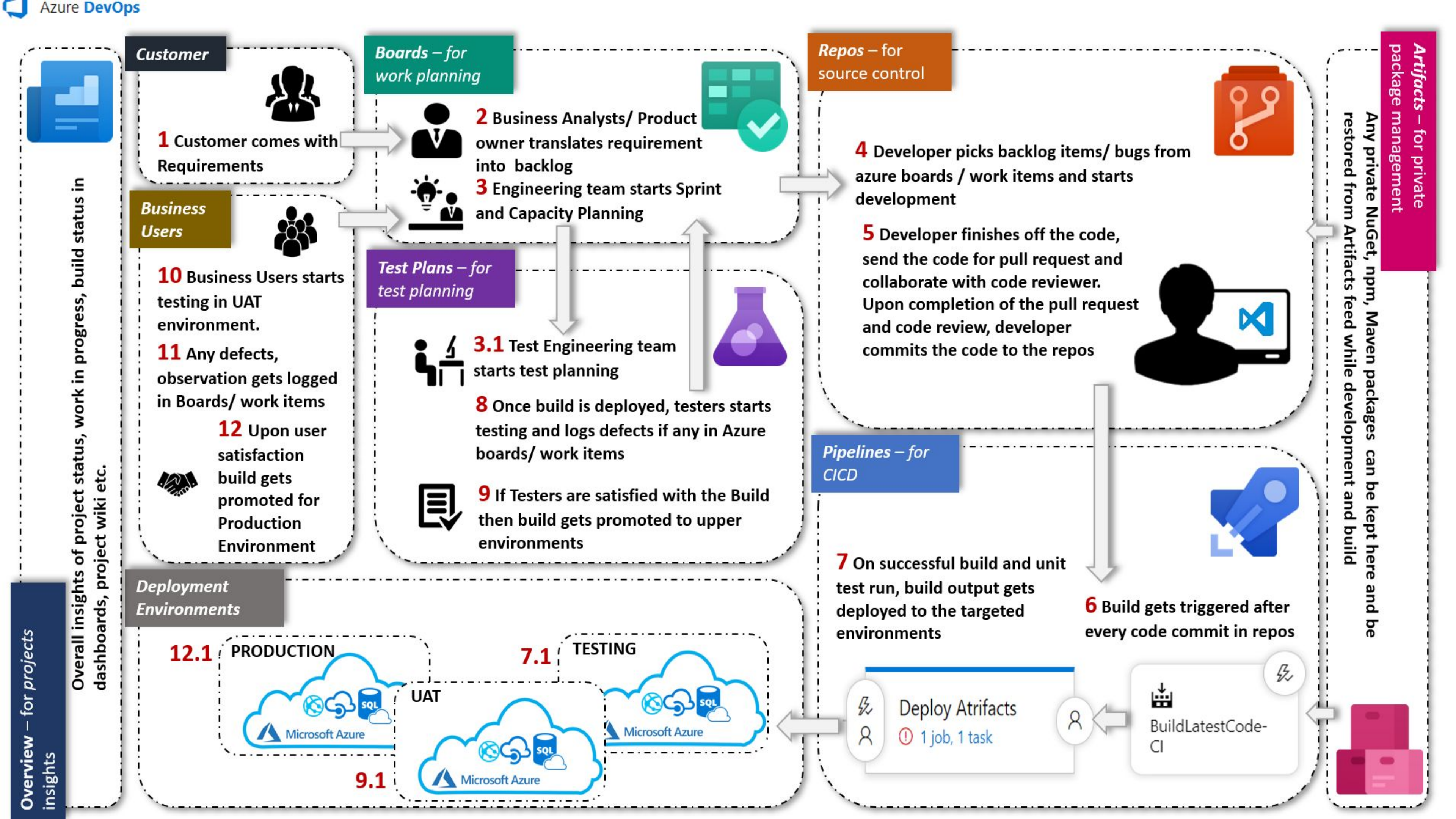


Tracking area	Basic	Agile	Scrum	CMMI
Workflow states	• To Do • Doing • Done	• New • Active • Resolved • Closed • Removed	• New • Approved • Committed • Done • Removed	• Proposed • Active • Resolved • Closed
Product planning (see note 1)	• Issue	• User story • Bug (optional)	• Product backlog item • Bug (optional)	• Requirement • Bug (optional)
Portfolio backlogs (2)	• Epic	• Epic • Feature	• Epic • Feature	• Epic • Feature
Task and sprint planning (3)	• Task	• Task • Bug (optional)	• Task • Bug (optional)	• Task • Bug (optional)
Bug backlog management (1)	• Issue	• Bug	• Bug	• Bug
Issue and risk management	• Issue	• Issue	• Impediment	• Issue • Risk • Review

The Basic Process



Azure **DevOps**



Overview *– for projects insights*

Overall insights of project status, work in progress, build status in dashboards, project wiki etc.

Customer

1 Customer comes with Requirements

Boards *– for work planning*

2 Business Analysts/ Product owner translates requirement into backlog

3 Engineering team starts Sprint and Capacity Planning

Repos – for source control

4 Developer picks backlog items/ bugs from azure boards / work items and starts development

5 Developer finishes off the code, send the code for pull request and collaborate with code reviewer. Upon completion of the pull request and code review, developer commits the code to the repos

Artifacts *– for private package management*

Any private NuGet, npm, Maven packages can be kept here and be restored from Artifacts feed while development and build

Business Users

10 Business Users starts testing in UAT environment.

11 Any defects, observation gets logged in Boards/ work items

12 Upon user satisfaction build gets promoted for Production Environment

Test Plans *– for test planning*

3.1 Test Engineering team starts test planning

8 Once build is deployed, testers starts testing and logs defects if any in Azure boards/ work items

9 If Testers are satisfied with the Build then build gets promoted to upper environments

Pipelines *– for CICD*

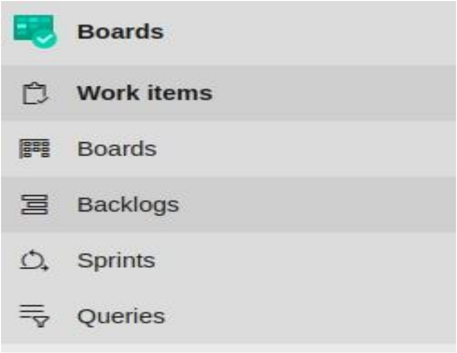
7 On successful build and unit test run, build output gets deployed to the targeted environments

6 Build gets triggered after every code commit in repos

Deployment Environments

Azure Boards

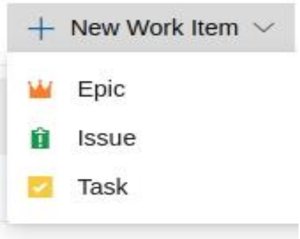
- Work Items are a part(small or large) of product development.
- Only 3 work items in basic(minimalistic)
- One Nested level
  1. Split Epics into Issues
  2. Split Issues into Tasks



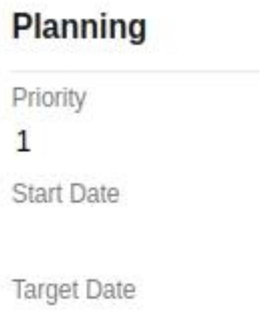
Boards	Service for managing work items
Backlogs	Product Backlog Item(PBI)
Sprints	Iterations
Queries	Custom Filter for Analytics

Azure Boards

- Epics, Issues and tasks can be interlinked
- Start and end date for EPIC
- Effort estimation for Issues
- Issues/Epics can be mapped to an Iteration/Sprint
- Discussions, Attachments, branches and tags



- □ Title
- □ Assigned To
- □ State
- □ Area
- □ Iteration

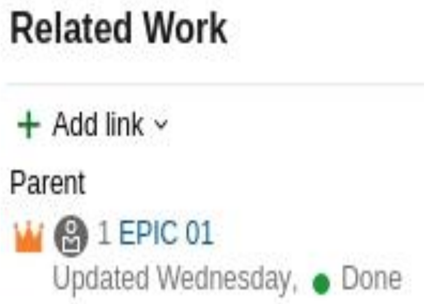


Planning

Priority
1

Start Date

Target Date



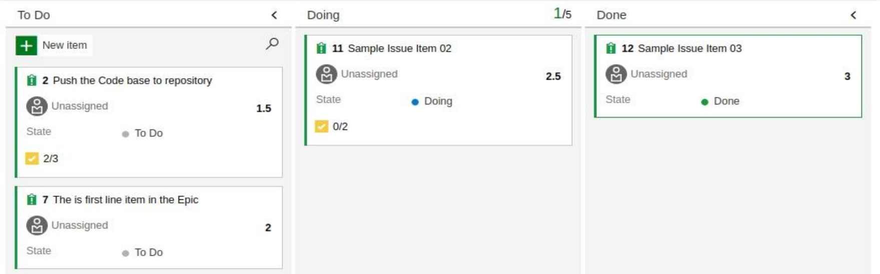
Related Work

Parent

1 EPIC 01
Updated Wednesday, ● Done

Azure Boards

- Board Items as Epics and Issues
- Movable between the lanes
- State changes are Automatically updated
- Lists current state, total issues, sum of effort and status
- Supports Filters



To Do | Doing 1/5 | Done

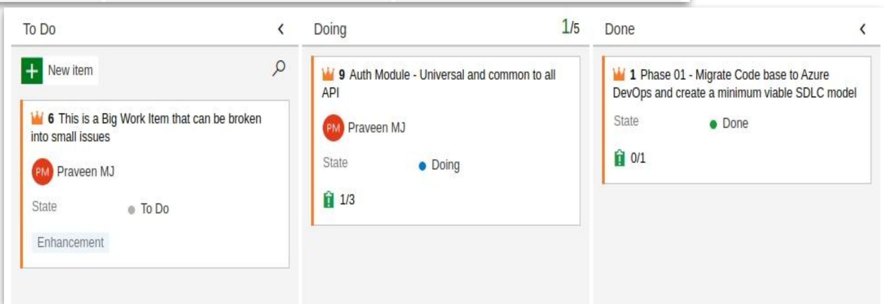
New item

2 Push the Code base to repository
Unassigned 1.5
State ● To Do
2/3

7 The is first line item in the Epic
Unassigned 2
State ● To Do

11 Sample Issue Item 02
Unassigned 2.5
State ● Doing
0/2

12 Sample Issue Item 03
Unassigned 3
State ● Done



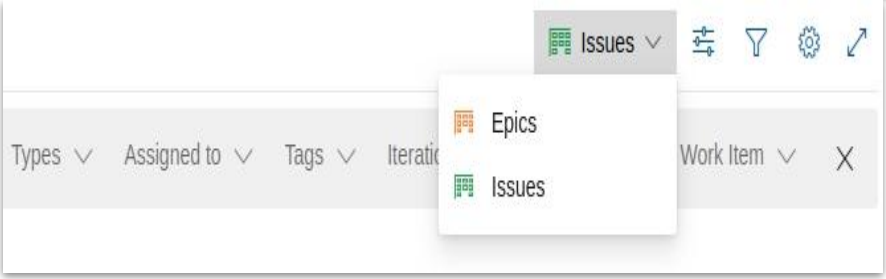
To Do | Doing 1/5 | Done

New item

6 This is a Big Work Item that can be broken into small issues
PM Praveen MJ
State ● To Do
Enhancement

9 Auth Module - Universal and common to all API
PM Praveen MJ
State ● Doing
1/3

1 Phase 01 - Migrate Code base to Azure DevOps and create a minimum viable SDLC model
State ● Done
0/1



Manage Boards

- Add work items directly from boards Or using **Boards** ➡ Work items ➡ + New Work Item ⌄
- To Manage boards, Select **Boards** ➡ Boards /Add work items directly from board + New item
- Create multiple teams using Project settings ➡ General ➡ Teams ➡ New Team
- Assign permission based on Role
  - Readers (*Guest*)
  - Project Administrators (*Admin*)
  - Contributors (*Coders/Developers*)
  - Stakeholders (*Product Owners*)
- Lists current state, total issues, sum of effort and status
- Supports Filters

Access Team profile using 👥

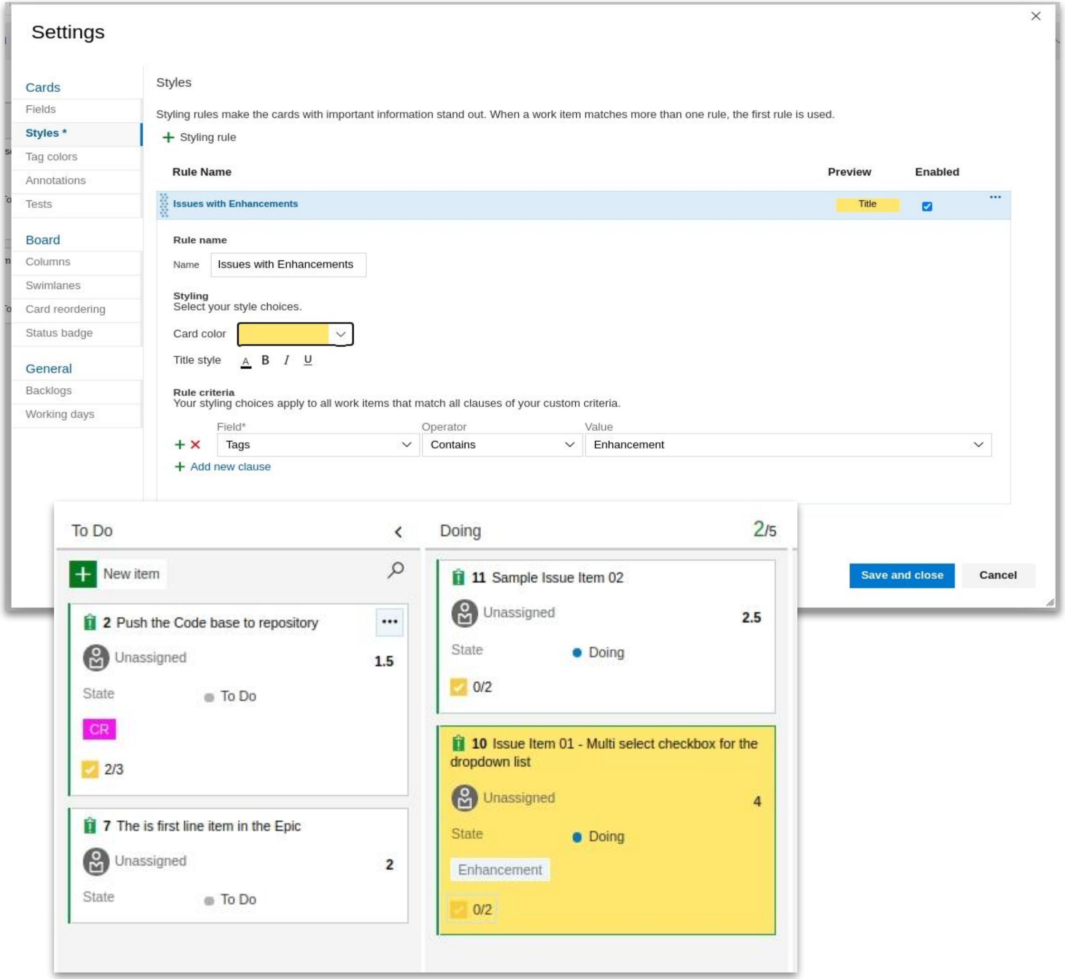
Toggle filter option using ▽

Customize team configurations, add styles and tag colors, field options ⚙

Select board work items by Issue type Issues ⌄

Customize Boards

- This sample customization applies a Style with rule name called ‘Issues with Enhancement tags’ for a tag called ‘Enhancement’
- Applies the style in boards once saved/Enabled
- Suitable to tags as well by editing the Tag colors



Styles

Styling rules make the cards with important information stand out. When a work item matches more than one rule, the first rule is used.

Rule Name	Preview	Enabled
Issues with Enhancements	Title	☑

Rule name

Name: Issues with Enhancements

Styling
Select your style choices.

Card color:

Title style: A B I U

Rule criteria
Your styling choices apply to all work items that match all clauses of your custom criteria.

Field*	Operator	Value
Tags	Contains	Enhancement

To Do

2 Push the Code base to repository
Unassigned 1.5
State: To Do
CR
2/3

7 The is first line item in the Epic
Unassigned 2
State: To Do

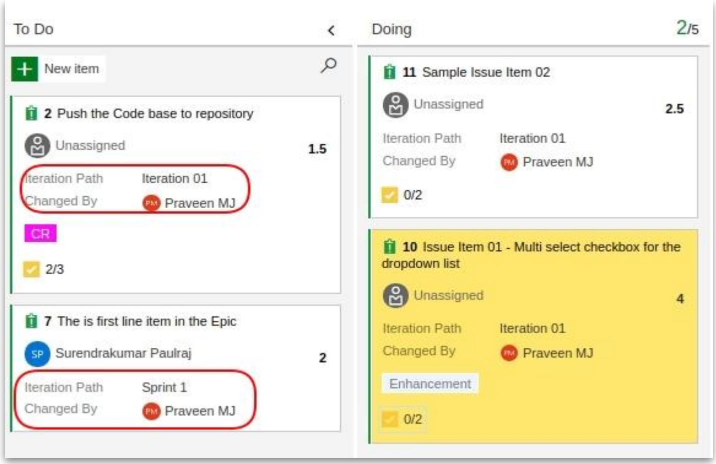
Doing 2/5

11 Sample Issue Item 02
Unassigned 2.5
State: Doing
0/2

10 Issue Item 01 - Multi select checkbox for the dropdown list
Unassigned 4
State: Doing
Enhancement
0/2

Customize Boards

- Update field property to customize Field related settings



To Do

New item

2 Push the Code base to repository

Unassigned 1.5

Iteration Path Iteration 01

Changed By Praveen MJ

CR

2/3

7 The is first line item in the Epic

Surendrakumar Paulraj 2

Iteration Path Sprint 1

Changed By Praveen MJ

Doing 2/5

11 Sample Issue Item 02

Unassigned 2.5

Iteration Path Iteration 01

Changed By Praveen MJ

0/2

10 Issue Item 01 - Multi select checkbox for the dropdown list

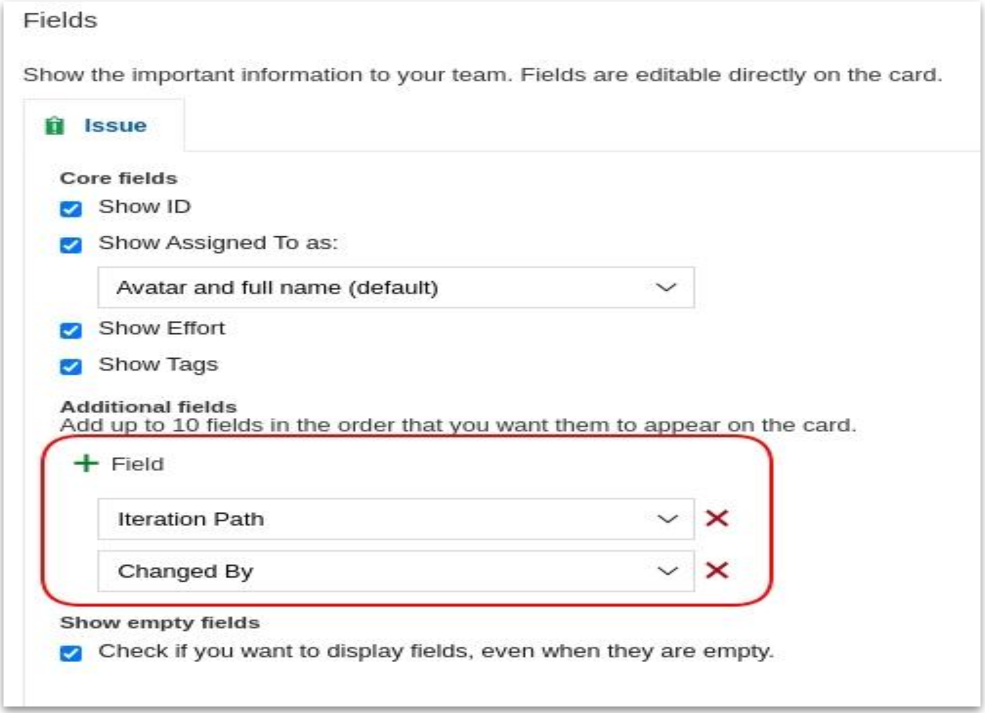
Unassigned 4

Iteration Path Iteration 01

Changed By Praveen MJ

Enhancement

0/2



Fields

Show the important information to your team. Fields are editable directly on the card.

Issue

Core fields

☑ Show ID

☑ Show Assigned To as:

Avatar and full name (default)

☑ Show Effort

☑ Show Tags

Additional fields

Add up to 10 fields in the order that you want them to appear on the card.

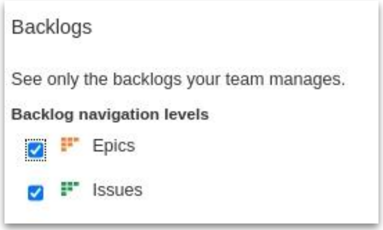
+ Field

Iteration Path ✕

Changed By ✕

Show empty fields

☑ Check if you want to display fields, even when they are empty.



Backlogs

See only the backlogs your team manages.

Backlog navigation levels

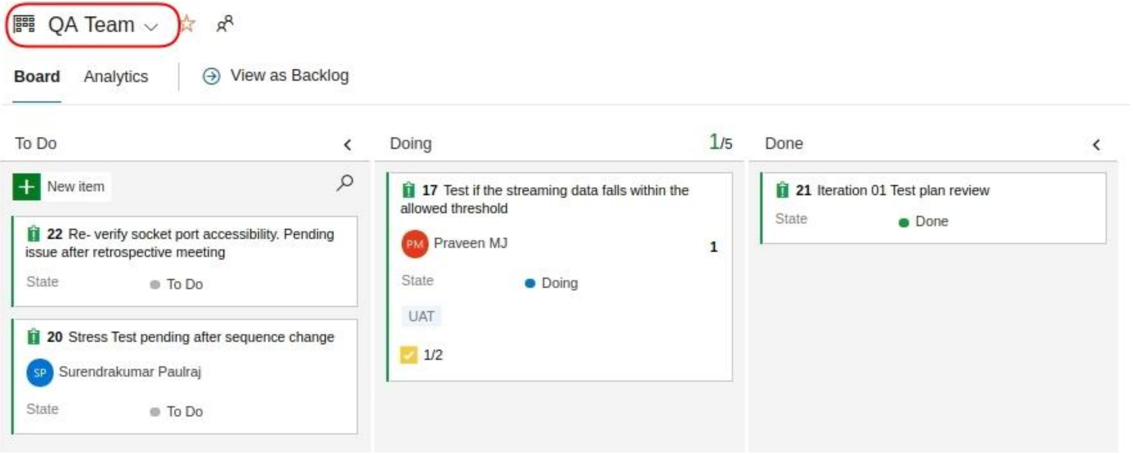
☑ Epics

☑ Issues

- Control what to display in board

Boards for Individual Teams

- Each team can have their own Managed boards



QA Team

Board Analytics View as Backlog

To Do

New item

22 Re- verify socket port accessibility. Pending issue after retrospective meeting
State ● To Do

20 Stress Test pending after sequence change
SP Surendrakumar Paulraj
State ● To Do

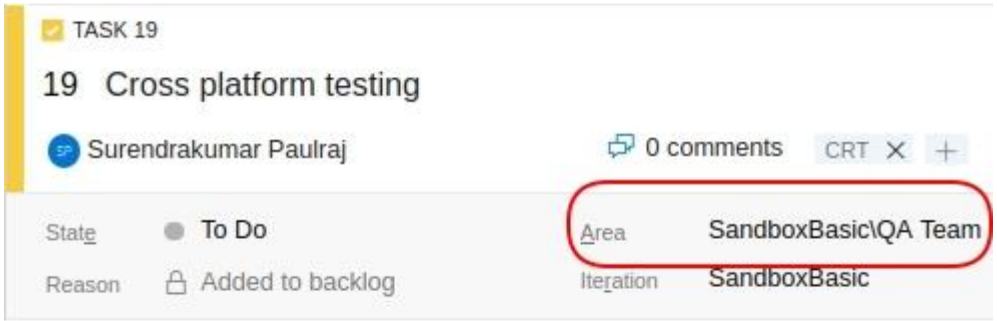
Doing 1/5

17 Test if the streaming data falls within the allowed threshold
PM Praveen MJ 1
State ● Doing
UAT
1/2

Done

21 Iteration 01 Test plan review
State ● Done

- Teams are assigned while creating an issue, mapping to Area path



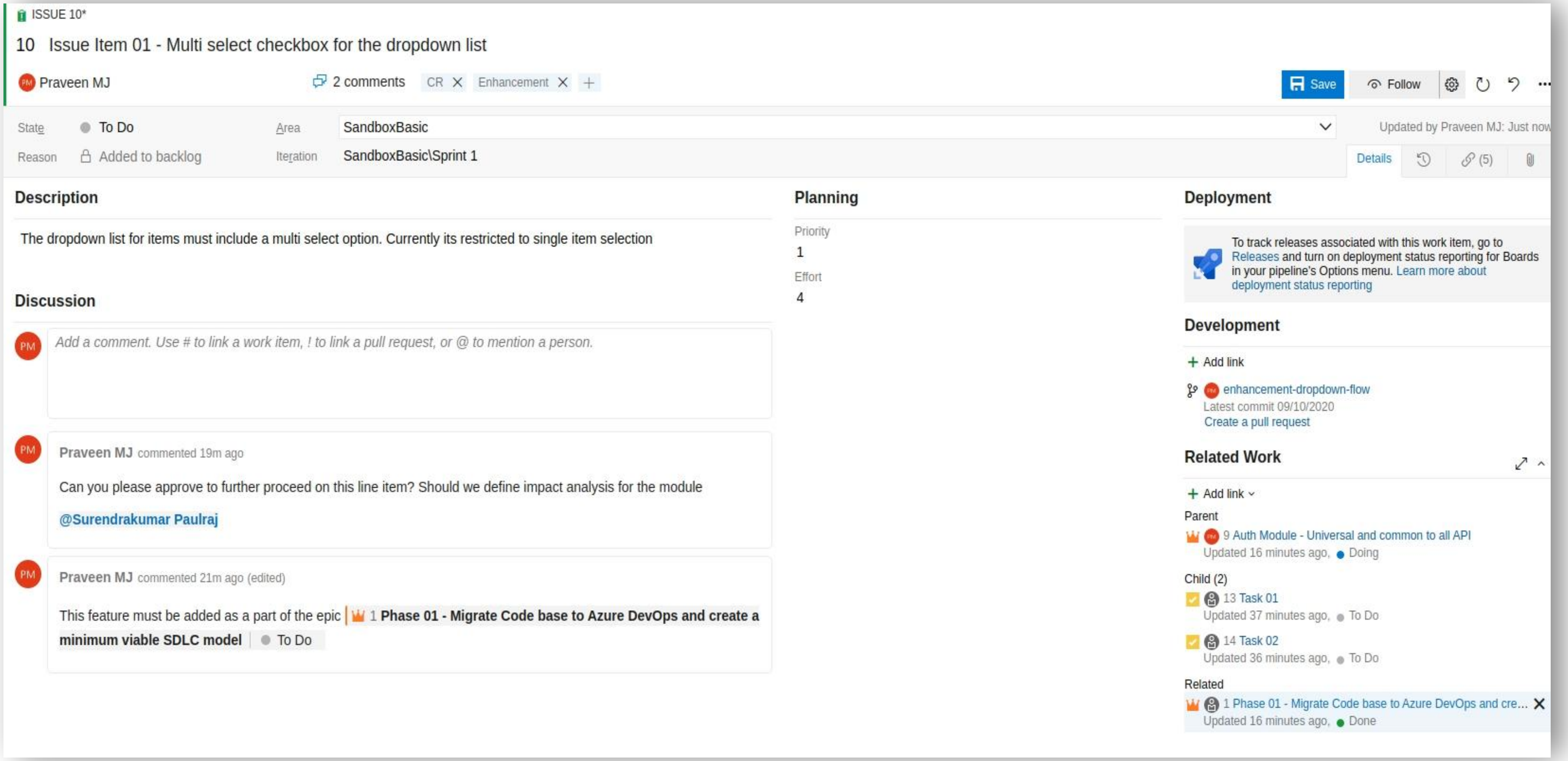
TASK 19

19 Cross platform testing

SP Surendrakumar Paulraj 0 comments CRT

State ● To Do
Reason Added to backlog
Area SandboxBasic\QA Team
Iteration SandboxBasic

Sample Work Item



10 Issue Item 01 - Multi select checkbox for the dropdown list

Praveen MJ

State: To Do

Reason: Added to backlog

Area: SandboxBasic

Iteration: SandboxBasic\Sprint 1

Updated by Praveen MJ: Just now

Description

The dropdown list for items must include a multi select option. Currently its restricted to single item selection

Discussion

Add a comment. Use # to link a work item, ! to link a pull request, or @ to mention a person.

Praveen MJ commented 19m ago

Can you please approve to further proceed on this line item? Should we define impact analysis for the module

@Surendrakumar Paulraj

Praveen MJ commented 21m ago (edited)

This feature must be added as a part of the epic | 1 **Phase 01 - Migrate Code base to Azure DevOps and create a minimum viable SDLC model** | ● To Do

Planning

Priority: 1

Effort: 4

Deployment

To track releases associated with this work item, go to Releases and turn on deployment status reporting for Boards in your pipeline's Options menu. Learn more about deployment status reporting

Development

enhancement-dropdown-flow
Latest commit 09/10/2020
Create a pull request

Related Work

Parent

9 Auth Module - Universal and common to all API
Updated 16 minutes ago, ● Doing

Child (2)

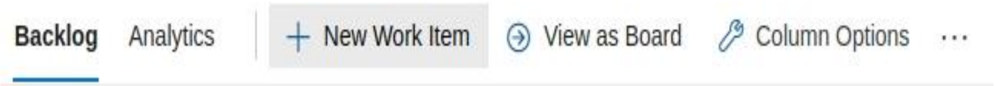
13 Task 01
Updated 37 minutes ago, ● To Do

14 Task 02
Updated 36 minutes ago, ● To Do

Related

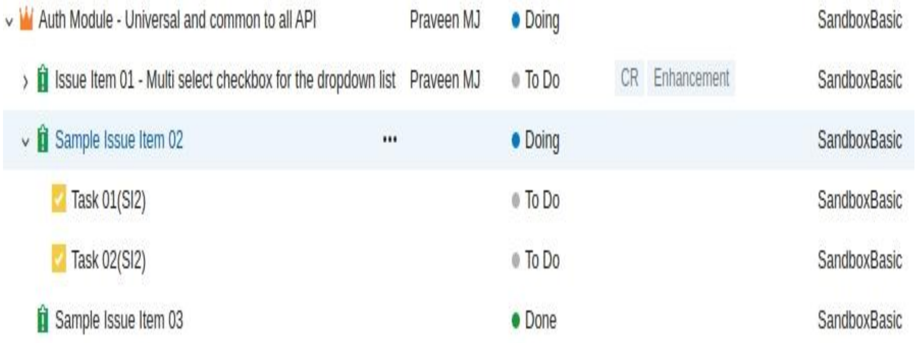
1 Phase 01 - Migrate Code base to Azure DevOps and cre...
Updated 16 minutes ago, ● Done

Azure Backlogs



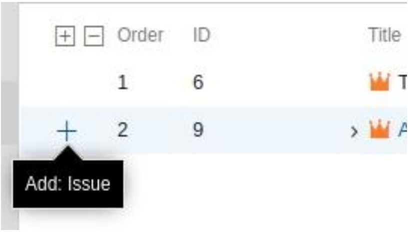
Backlog Analytics + New Work Item View as Board Column Options ···

- Backlogs denotes PBI
- PBIs can be moved between Sprints
- Supports Filters
- Custom Columns



⌄ Auth Module - Universal and common to all API	Praveen MJ	● Doing		SandboxBasic
› Issue Item 01 - Multi select checkbox for the dropdown list	Praveen MJ	● To Do	CR Enhancement	SandboxBasic
⌄ Sample Issue Item 02 ···		● Doing		SandboxBasic
Task 01(SI2)		● To Do		SandboxBasic
Task 02(SI2)		● To Do		SandboxBasic
Sample Issue Item 03		● Done		SandboxBasic

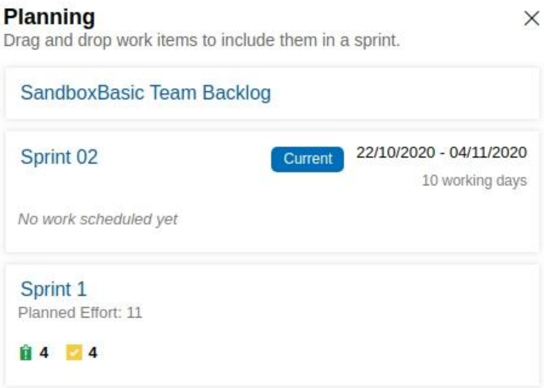
Hierarchical Display with stats



	Order	ID	Title
	1	6	T
+	2	9	› A

Add: Issue

Add Issue/Navigate between boards



Planning

Drag and drop work items to include them in a sprint.

SandboxBasic Team Backlog

Sprint 02 Current 22/10/2020 - 04/11/2020
10 working days

No work scheduled yet

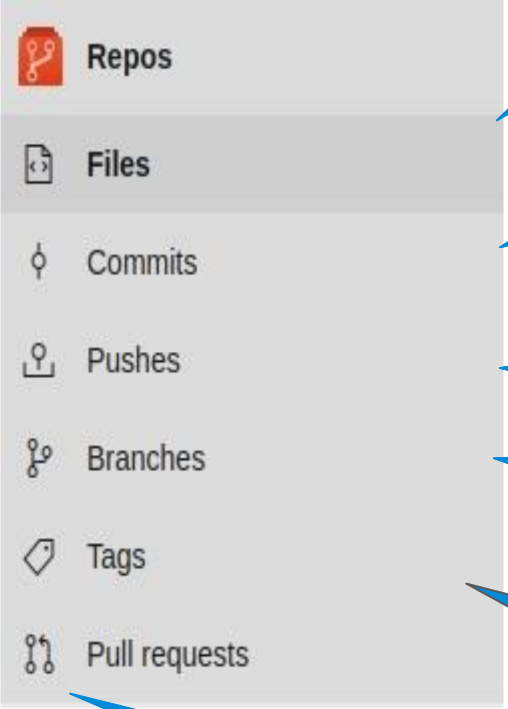
Sprint 1
Planned Effort: 11

4 4

Sprint Planning Sidebar with timeline

Azure Repo

- Provides set of version control tools used to manage your code.
- Save your work and coordinate code changes across your team
- Single Source of Truth for Product
- Track-Review-Approve
- Code Quality check with Continuous Integration



Display source files based on branch

Show changes with hash ID

Changes pushed to Cloud

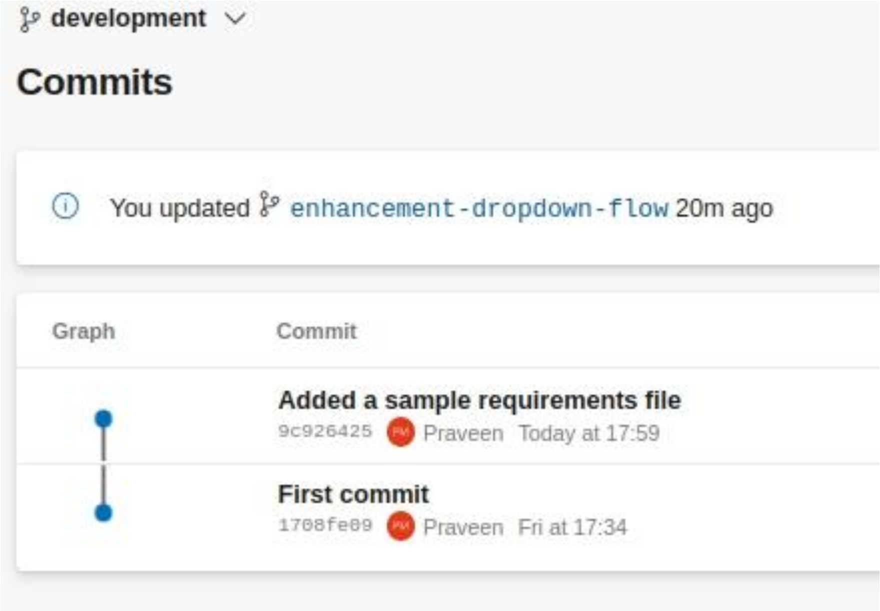
Manageable Source branches

Unique identifier in history

Merge Request/Approvals

Commits/Pushes

- A snapshot of the hierarchy (Git tree) and the contents of the files (Git blob)
- Local changes Vs Cloud Changes
- Maintains a history by default
- Move to any point in history with ID
- Tags, Link Issues
- Switch between branches

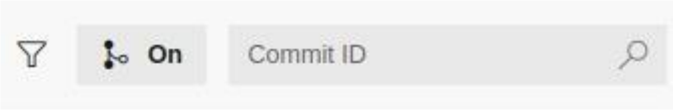


development

Commits

You updated enhancement-dropdown-flow 20m ago

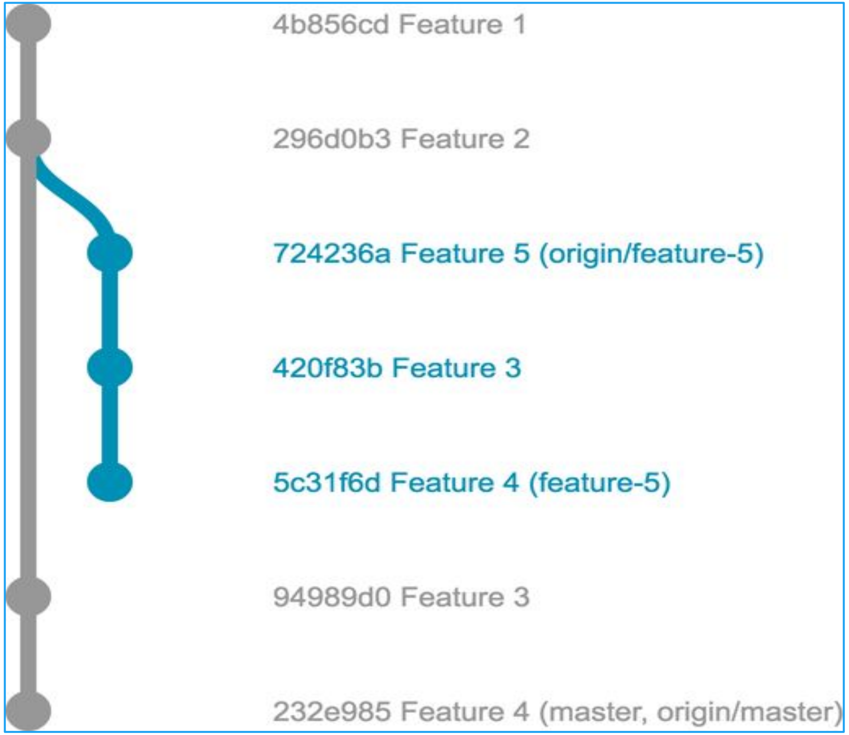
Graph	Commit
●	**Added a sample requirements file** 9c926425 Praveen Today at 17:59
●	**First commit** 1708fe09 Praveen Fri at 17:34



Filters

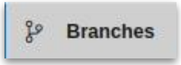
Branch Streams

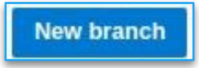
- A branch is essentially is a unique set of code changes with a unique name
- Each repository can have one or more branches
- The main branch — the one where all changes eventually get merged back into, and is called master

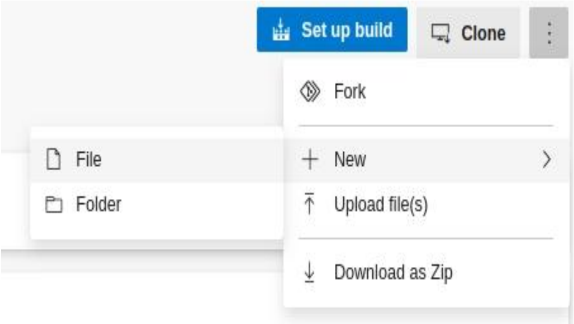


```
4b856cd Feature 1
296d0b3 Feature 2
724236a Feature 5 (origin/feature-5)
420f83b Feature 3
5c31f6d Feature 4 (feature-5)
94989d0 Feature 3
232e985 Feature 4 (master, origin/master)
```

Process Flow

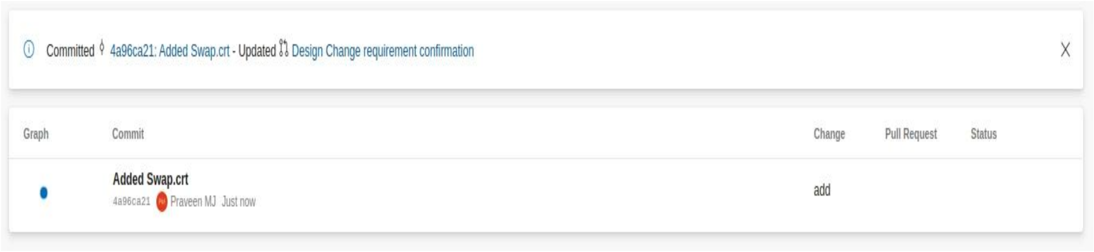
- A ReadMe file is created on repository in cloud(for clone). Click **clone**
- Create standalone branches or issue specific branches from 





- Add Files/Folders.
- Commit/save the added files in cloud. (Tagged to a branch, and issue)

- Changes are saved with commit id
- Issues are tagged as hyperlink near commits

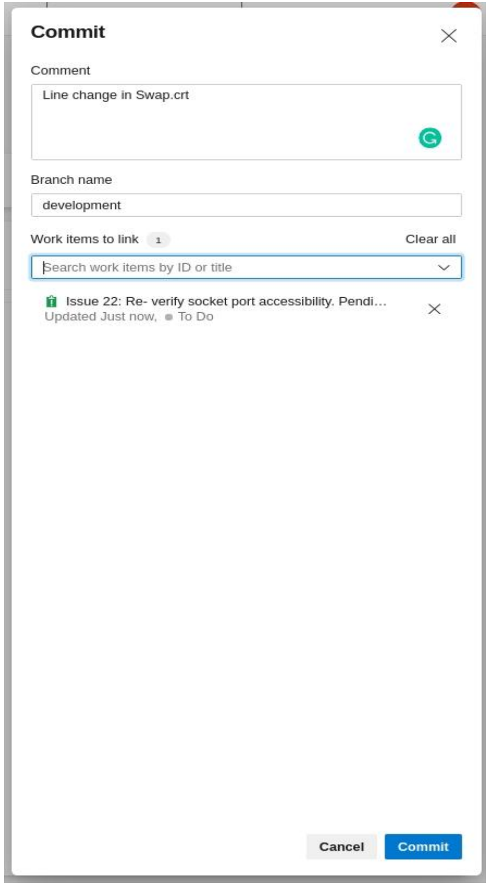


Committed 4a96ca21: Added Swap.crt - Updated Design Change requirement confirmation

Graph	Commit	Change	Pull Request	Status
●	Added Swap.crt 4a96ca21 Praveen MJ Just now	add		

Process Flow

Comment issue, tag users, issues and branches



Commit

Comment

Line change in Swap.crt

Branch name

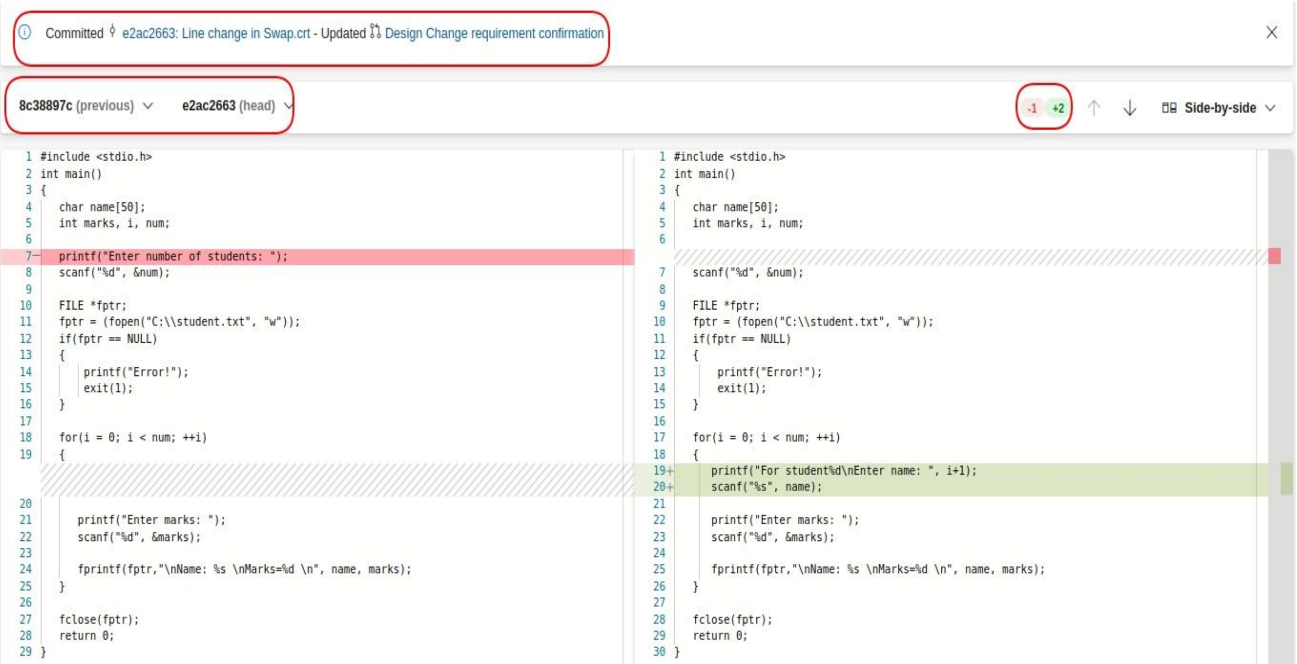
development

Work items to link 1 — Clear all

Search work items by ID or title

Issue 22: Re- verify socket port accessibility. Pendi…
Updated Just now, ● To Do

Cancel — Commit



Committed e2ac2663: Line change in Swap.crt - Updated Design Change requirement confirmation

8c38897c (previous) — e2ac2663 (head) — -1 +2 — Side-by-side

```
#include <stdio.h>
int main()
{
    char name[50];
    int marks, i, num;

    printf("Enter number of students: ");
    scanf("%d", &num);

    FILE *fptr;
    fptr = (fopen("C:\\student.txt", "w"));
    if(fptr == NULL)
    {
        printf("Error!");
        exit(1);
    }

    for(i = 0; i < num; ++i)
    {

        printf("Enter marks: ");
        scanf("%d", &marks);

        fprintf(fptr,"\nName: %s \nMarks=%d \n", name, marks);
    }

    fclose(fptr);
    return 0;
}
```

```
#include <stdio.h>
int main()
{
    char name[50];
    int marks, i, num;

    scanf("%d", &num);

    FILE *fptr;
    fptr = (fopen("C:\\student.txt", "w"));
    if(fptr == NULL)
    {
        printf("Error!");
        exit(1);
    }

    for(i = 0; i < num; ++i)
    {
        printf("For student%d\nEnter name: ", i+1);
        scanf("%s", name);

        printf("Enter marks: ");
        scanf("%d", &marks);

        fprintf(fptr,"\nName: %s \nMarks=%d \n", name, marks);
    }

    fclose(fptr);
    return 0;
}
```

Compare between two commits

Process Flow from Local to Remote

1. Initialize a .git file to track configurations
2. Do code changes.
3. Add changes.

```
> git add .
```

4. Compare tracked and untracked changes.

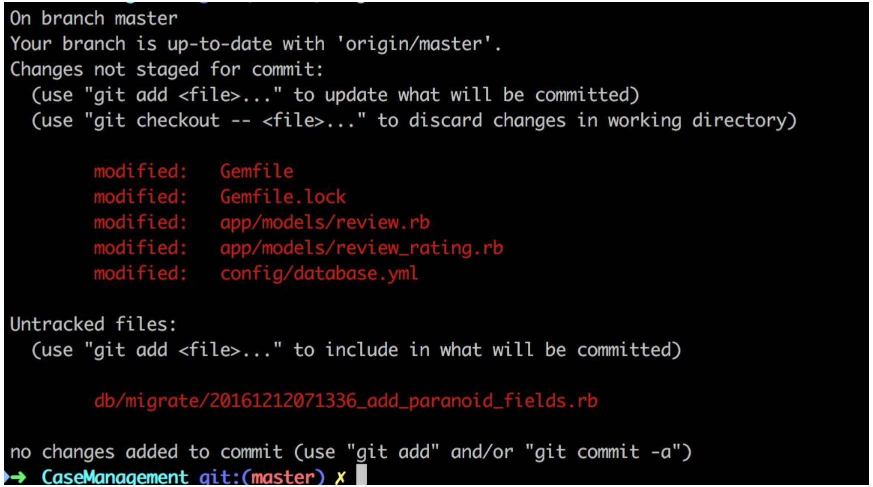
```
> git status
```

5. Commit the changes to local git history with a commit message

```
> git commit -m “Commit message”
```

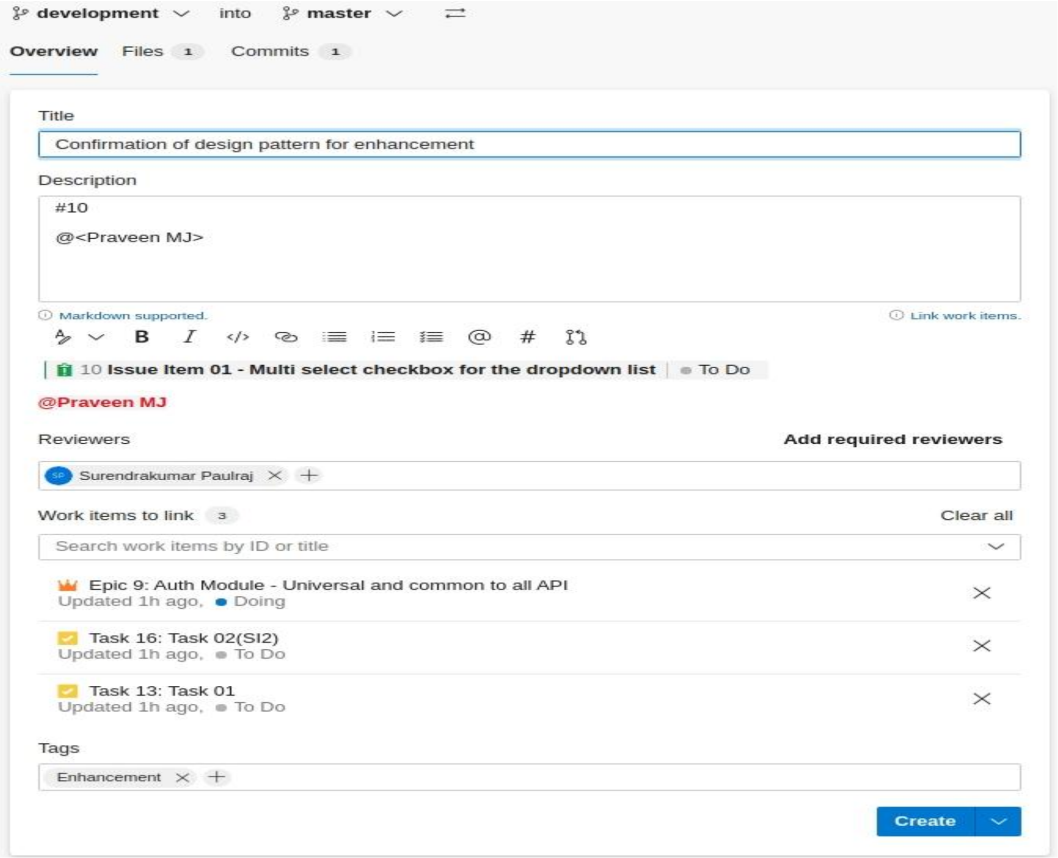
6. Push the changes to remote repository

```
> git push origin development
```



```
On branch master
Your branch is up-to-date with 'origin/master'.
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   Gemfile
        modified:   Gemfile.lock
        modified:   app/models/review.rb
        modified:   app/models/review_rating.rb
        modified:   config/database.yml

Untracked files:
  (use "git add <file>..." to include in what will be committed)

        db/migrate/20161212071336_add_paranoid_fields.rb

no changes added to commit (use "git add" and/or "git commit -a")
➜ CaseManagement git:(master) ✗
```

Pull Request/Merge



Title: Confirmation of design pattern for enhancement

Description:
#10
@<Praveen MJ>

10 Issue Item 01 - Multi select checkbox for the dropdown list | To Do

@Praveen MJ

Reviewers: Surendrakumar Paulraj

Work items to link 3

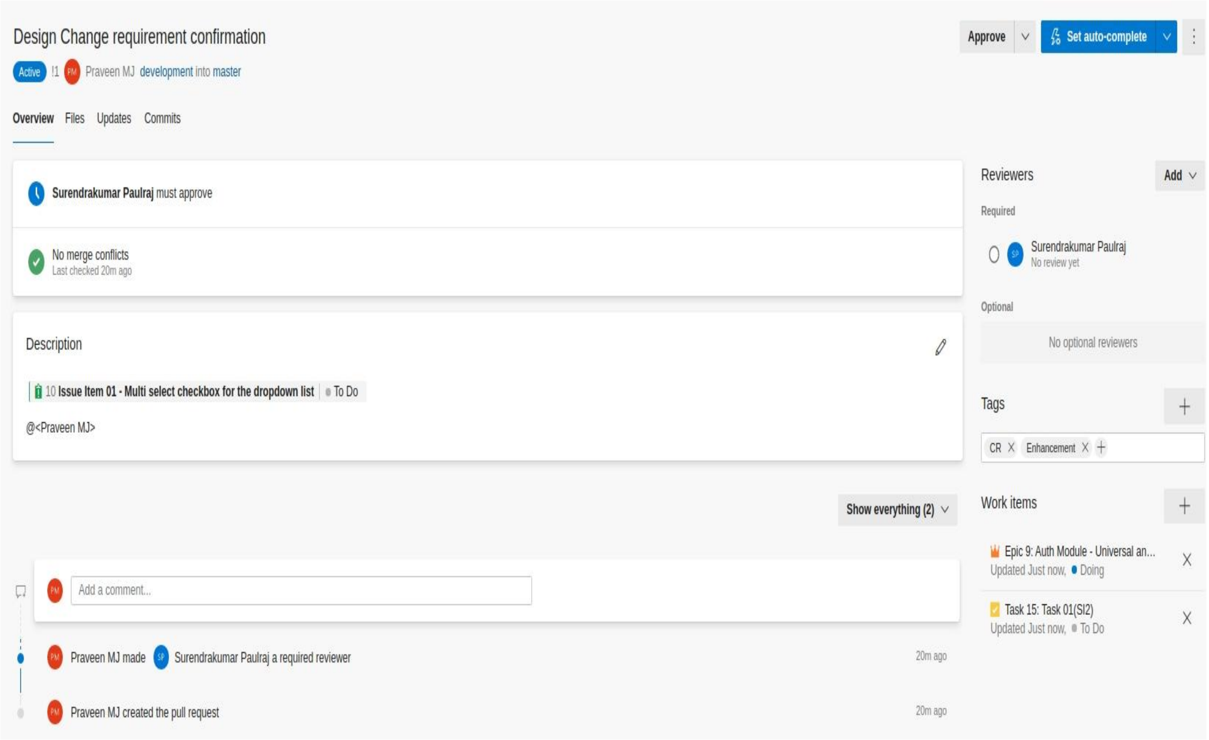
Search work items by ID or title

- Epic 9: Auth Module - Universal and common to all API
  Updated 1h ago, ● Doing
- Task 16: Task 02(SI2)
  Updated 1h ago, ● To Do
- Task 13: Task 01
  Updated 1h ago, ● To Do

Tags: Enhancement

- Create pull request to merge changes from branch1(development) to branch2(master)
- Info of Source Branch -> Target Branch
- Link Tasks, Issues, Epics
- Assign/Request-Required and optional reviewer
- Supports markdown and attachments

Pull Request/Approve



Design Change requirement confirmation

Active !1 PM Praveen MJ development into master

Surendrakumar Paulraj must approve

No merge conflicts
Last checked 20m ago

Description

10 Issue Item 01 - Multi select checkbox for the dropdown list · To Do

@<Praveen MJ>

Add a comment...

PM Praveen MJ made SP Surendrakumar Paulraj a required reviewer 20m ago

PM Praveen MJ created the pull request 20m ago

Reviewers

Required

SP Surendrakumar Paulraj
No review yet

Optional

No optional reviewers

Tags

CR, Enhancement

Work items

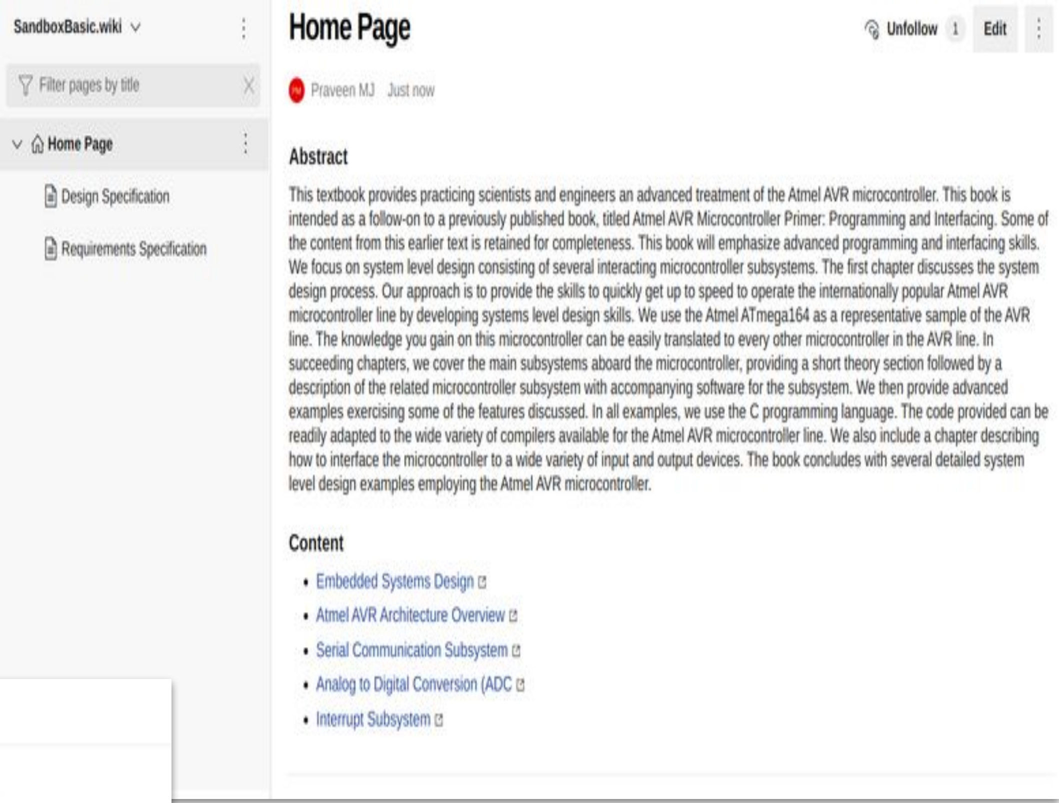
Epic 9: Auth Module - Universal an...
Updated Just now, · Doing

Task 15: Task 01(SI2)
Updated Just now, · To Do

- Mandatest approval from all required reviewers
- Tag, comment, Add suggestion on approval
- Reject/Restart PR
- Pending PR will be shown in Pull requests

Azure Wiki

- Modern Digital Documentation
- Supports Markdown and attachments
- Collaborative document management
- Create and manage sub sections
- Tag/Link users and issues



Home Page

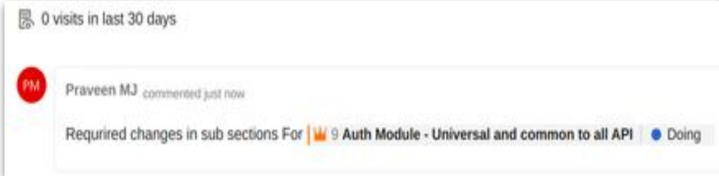
Praveen MJ Just now

Abstract

This textbook provides practicing scientists and engineers an advanced treatment of the Atmel AVR microcontroller. This book is intended as a follow-on to a previously published book, titled Atmel AVR Microcontroller Primer: Programming and Interfacing. Some of the content from this earlier text is retained for completeness. This book will emphasize advanced programming and interfacing skills. We focus on system level design consisting of several interacting microcontroller subsystems. The first chapter discusses the system design process. Our approach is to provide the skills to quickly get up to speed to operate the internationally popular Atmel AVR microcontroller line by developing systems level design skills. We use the Atmel ATmega164 as a representative sample of the AVR line. The knowledge you gain on this microcontroller can be easily translated to every other microcontroller in the AVR line. In succeeding chapters, we cover the main subsystems aboard the microcontroller, providing a short theory section followed by a description of the related microcontroller subsystem with accompanying software for the subsystem. We then provide advanced examples exercising some of the features discussed. In all examples, we use the C programming language. The code provided can be readily adapted to the wide variety of compilers available for the Atmel AVR microcontroller line. We also include a chapter describing how to interface the microcontroller to a wide variety of input and output devices. The book concludes with several detailed system level design examples employing the Atmel AVR microcontroller.

Content

- Embedded Systems Design
- Atmel AVR Architecture Overview
- Serial Communication Subsystem
- Analog to Digital Conversion (ADC
- Interrupt Subsystem

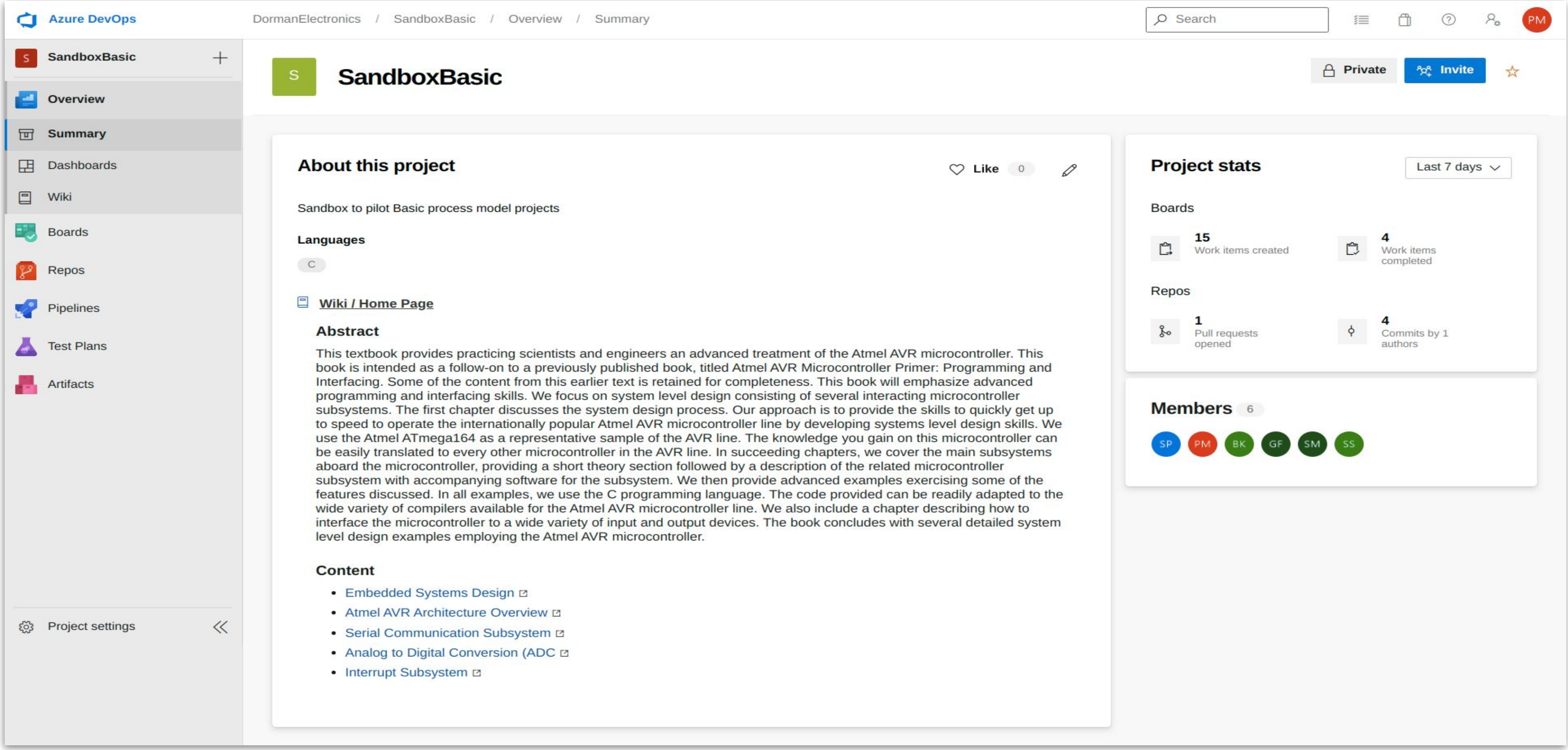


0 visits in last 30 days

Praveen MJ commented just now

Required changes in sub sections For | 9 Auth Module - Universal and common to all API ● Doing

Azure Dashboard Overview



SandboxBasic

About this project

Sandbox to pilot Basic process model projects

Languages

C

Wiki / Home Page

Abstract

This textbook provides practicing scientists and engineers an advanced treatment of the Atmel AVR microcontroller. This book is intended as a follow-on to a previously published book, titled Atmel AVR Microcontroller Primer: Programming and Interfacing. Some of the content from this earlier text is retained for completeness. This book will emphasize advanced programming and interfacing skills. We focus on system level design consisting of several interacting microcontroller subsystems. The first chapter discusses the system design process. Our approach is to provide the skills to quickly get up to speed to operate the internationally popular Atmel AVR microcontroller line by developing systems level design skills. We use the Atmel ATmega164 as a representative sample of the AVR line. The knowledge you gain on this microcontroller can be easily translated to every other microcontroller in the AVR line. In succeeding chapters, we cover the main subsystems aboard the microcontroller, providing a short theory section followed by a description of the related microcontroller subsystem with accompanying software for the subsystem. We then provide advanced examples exercising some of the features discussed. In all examples, we use the C programming language. The code provided can be readily adapted to the wide variety of compilers available for the Atmel AVR microcontroller line. We also include a chapter describing how to interface the microcontroller to a wide variety of input and output devices. The book concludes with several detailed system level design examples employing the Atmel AVR microcontroller.

Content

- Embedded Systems Design
- Atmel AVR Architecture Overview
- Serial Communication Subsystem
- Analog to Digital Conversion (ADC
- Interrupt Subsystem

Project stats

Last 7 days

Boards

15 Work items created

4 Work items completed

Repos

1 Pull requests opened

4 Commits by 1 authors

Members 6

SP PM BK GF SM SS

Azure Dashboard Overview

- Overall Overview Dashboard
- Customizable widget dashboards
- Query Generators/Widgets
- Specific to Teams/Users

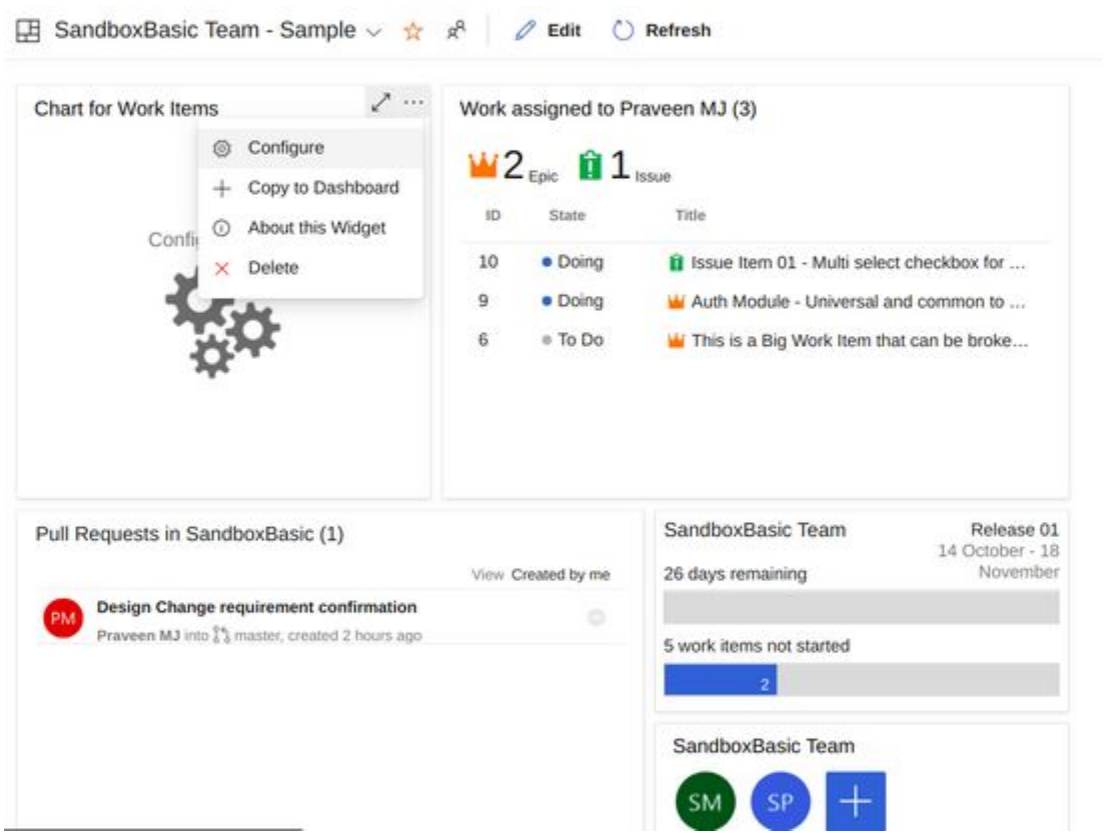


Chart for Work Items

Work assigned to Praveen MJ (3)

2 Epic 1 Issue

ID	State	Title
10	● Doing	Issue Item 01 - Multi select checkbox for ...
9	● Doing	Auth Module - Universal and common to ...
6	● To Do	This is a Big Work Item that can be broke...

Pull Requests in SandboxBasic (1)

Design Change requirement confirmation
Praveen MJ into master, created 2 hours ago

SandboxBasic Team
Release 01
14 October - 18 November
26 days remaining
5 work items not started
2

SandboxBasic Team

Thank You