

Azure DevOps - Theoretical Assignment

Problem Statement:

You are part of a software development team working on a new internal tool named “TeamPulse” — a web-based project management dashboard that helps organizations track productivity, project milestones, and collaboration efficiency.

Your task is to structure this project in Azure DevOps, defining a clear hierarchy of

Epics → Features → User Stories → Tasks.

Objective:

To demonstrate your understanding of Agile Project Structuring and how Azure DevOps supports:

- Work item management
- Team collaboration
- Sprint planning
- Traceability and reporting

Submission:

2. Screenshots:

The screenshot shows the Azure DevOps interface for the 'TeamPulse Project'. The left sidebar contains navigation links: TeamPulse Project, Overview, Summary, Dashboards, Wiki, Boards, Repos, Pipelines, Test Plans, Artifacts, and Project settings. The main content area is titled 'TeamPulse Project' and includes sections for 'About this project' (with a 'Add Project Description' button), 'Project stats' (showing 51 work items created and 28 completed), and 'Members' (listing one member). A cartoon character is running on clouds in the background of the stats section.

Epic Screenshot:

Epic

- TeamPulse – Project Management Dashboard

Azure DevOps jasminem1 / TeamPulse Project / Boards / Work items

Work items

Recently updated | + New Work Item | Open filtered view in Queries | Column Options | Import Work Items | ...

| ID | Title | Assigned To | State | Area Path |
|-----|--|-------------|-------|-------------------|
| 292 | TeamPulse – Project Management Dashboard | JAS M | New | TeamPulse Project |

Filter by keyword

Epic | Assigned to | States | Area | Tags | X

Boards | Backlogs | Sprints | Queries | Delivery Plans | Analytics views | Repos | Pipelines | Test Plans | Artifacts | Project settings

Features Screenshot:

Features (Under Epic)

1. User Authentication & Role Management
2. Dashboard Analytics & KPIs
3. Task & Sprint Management
4. Team Chat & Notifications

Azure DevOps jasminem1 / TeamPulse Project / Boards / Work items

Work items

Recently updated | + New Work Item | Open filtered view in Queries | Column Options | Import Work Items | ...

| ID | Title | Assigned To | State | Area Path |
|-----|---------------------------------------|-------------|-------|-------------------|
| 296 | Team Chat & Notifications | JAS M | New | TeamPulse Project |
| 295 | Task & Sprint Management | JAS M | New | TeamPulse Project |
| 294 | Dashboard Analytics & KPIs | JAS M | New | TeamPulse Project |
| 293 | User Authentication & Role Management | JAS M | New | TeamPulse Project |

Filter by keyword

Feature | Assigned to | States | Area | Tags | X

Boards | Backlogs | Sprints | Queries | Delivery Plans | Analytics views | Repos | Pipelines | Test Plans | Artifacts | Project settings

User Story Screenshot:

Parent Feature: User Authentication & Role Management

- Admin: Create User Accounts
- Manager: Assign/Change User Roles
- User: Secure Account Login

Parent Feature: Dashboard Analytics & KPIs

- Manager: View Team Velocity KPI
- Executive: View Multi-Project Burndown
- User: View Personal Task Summary

Parent Feature: Task & Sprint Management

- Project Lead: Create & Define Sprints
- User: Update Task Status on Board
- Project Lead: Estimate Effort (Points/Hours)

Parent Feature: Team Chat & Notifications

- Team: Persistent Chat Sidebar
- User: In-App Notifications
- User: '@Mention' in Comments

The screenshot shows the Azure DevOps interface for the 'TeamPulse Project'. The left sidebar is collapsed, and the main area displays a table of work items under the 'Work items' tab. The table has the following columns: ID, Title, Assigned To, State, Area Path, Tags, and Comments. The data in the table is as follows:

| ID | Title | Assigned To | State | Area Path | Tags | Comments |
|-----|--|-------------|----------|-------------------|---------------------|----------|
| 302 | User: View Personal Task Summary | JAS M | Resolved | TeamPulse Project | Frontend, Usability | |
| 298 | Manager: Assign/Change User Roles | JAS M | Resolved | TeamPulse Project | Backend, Database | ... |
| 303 | Project Lead: Create & Define Sprints | JAS M | Active | TeamPulse Project | Backend, Planning | |
| 299 | User: Secure Account Login | JAS M | Resolved | TeamPulse Project | Frontend, Security | |
| 305 | Project Lead: Estimate Effort (Points/Hours) | JAS M | Active | TeamPulse Project | Database, Planning | |
| 304 | User: Update Task Status on Board | JAS M | Active | TeamPulse Project | Frontend, Usability | |
| 306 | Team: Persistent Chat Sidebar | JAS M | New | TeamPulse Project | Frontend, Real-time | |
| 307 | User: In-App Notifications | JAS M | New | TeamPulse Project | Backend, Frontend | ... |
| 308 | User: '@Mention' in Comments | JAS M | New | TeamPulse Project | Frontend, Usability | |
| 301 | Executive: View Multi-Project Burndown. | JAS M | Resolved | TeamPulse Project | Analytics, Frontend | ... |
| 300 | Manager: View Team Velocity KPI | JAS M | Resolved | TeamPulse Project | Analytics, Frontend | ... |
| 297 | Admin: Create User Accounts | JAS M | Resolved | TeamPulse Project | Backend, Frontend | ... |

Tasks Screenshot:

Parent User Story: Admin: Create User Accounts

- Define user roles and permissions.
- Design 'Create User' UI.
- Implement User Creation API.
- Configure Email Service.

Parent User Story: Manager: Assign/Change User Roles

- Update DB Schema for Roles.
- Develop Role Editing UI.
- Implement Role Change Logic.

Parent User Story: User: Secure Account Login

- Design Login/Reg Forms.
- Implement Auth Token Generation.
- Create Auth Data Flow Diagram.
- Document Password Reset Flow.

Parent User Story: Manager: View Team Velocity KPI

- Define/Calculate Velocity Metric.
- Develop KPI Data API.
- Implement Velocity Chart.

Parent User Story: Executive: View Multi-Project Burndown

- Aggregate Burndown Data.
- Design Interactive Burndown Chart.
- Optimize Historical Data Fetch.

Parent User Story: User: View Personal Task Summary

- Create User-Specific Tasks API.
- Develop 'My Work' Widget.

Parent User Story: Project Lead: Create & Define Sprints

- Design 'Sprint Settings' UI.
- Implement Sprint Date Logic.
- Link Work Items to Iterations.

Parent User Story: User: Update Task Status on Board

- Implement Kanban Drag/Drop.
- Develop Status Update API.
- Add Status Change to History.

Parent User Story: Project Lead: Estimate Effort (Points/Hours)

- Add Estimation Fields to Templates.
- Create Capacity View.

Parent User Story: Team: Persistent Chat Sidebar

- Integrate WebSocket Service.
- Design/Implement Chat UI.
- Implement Message Storage.

Parent User Story: User: In-App Notifications

- Implement Notification Service.
- Develop Notification Data Model/API.
- Create Notification UI/Badge.

Parent User Story: User: '@Mention' in Comments

- Implement '@' Autocomplete.
- Link '@Mention' to Notification Service.

Azure DevOps jasminem1 / TeamPulse Project / Boards / Work items ☰ 🖨️ 🔗 ✖ JM

| Work items | | | | | |
|------------|--|---|-------|-------------|--|
| | | Recently updated + New Work Item Open filtered view in Queries Column Options ... | | | |
| | | Filter by keyword | | | |
| ID | Title | Assigned To | State | Area Path | |
| 319 | Define and calculate Velocity KPI metric | JAS M | New | TeamPulse F | |
| 318 | Document password reset logic and flow | JAS M | New | TeamPulse F | |
| 317 | Create theoretical data flow diagrams | JAS M | New | TeamPulse F | |
| 316 | Implement server-side password validation and token generation | JAS M | New | TeamPulse F | |
| 315 | Design the login and registration form UI/UX | JAS M | New | TeamPulse F | |
| 314 | Implement role change authorization logic | JAS M | New | TeamPulse F | |
| 313 | Develop Role Editing UI component | JAS M | New | TeamPulse F | |
| 312 | Update DB Schema for Roles | JAS M | New | TeamPulse F | |
| 311 | Configure Email Service | JAS M | New | TeamPulse F | |
| 310 | Implement User Creation API | JAS M | New | TeamPulse F | |
| 309 | Design 'Create User' UI | JAS M | New | TeamPulse F | |

Azure DevOps jasminem1 / TeamPulse Project / Boards / Work items ☰ 🖨️ 🔗 ✖ JM

| Work items | | | | | |
|------------|---|---|-------|-------------|--|
| | | Recently updated + New Work Item Open filtered view in Queries Column Options ... | | | |
| | | Filter by keyword | | | |
| ID | Title | Assigned To | State | Area Path | |
| 330 | Implement drag-and-drop for Kanban cards | JAS M | New | TeamPulse F | |
| 329 | Ensure work items can be assigned to the new sprint | JAS M | New | TeamPulse F | |
| 328 | Implement sprint date validation and persistence | JAS M | New | TeamPulse F | |
| 327 | Design the 'Sprint Settings' configuration interface. | JAS M | New | TeamPulse F | |
| 326 | Develop the 'My Work' dashboard widget and sorting logic | JAS M | New | TeamPulse F | |
| 325 | Create API endpoint to fetch user-specific tasks | JAS M | New | TeamPulse F | |
| 324 | Optimize data fetch for historical views | JAS M | New | TeamPulse F | |
| 323 | Design and style the interactive Burndown Chart component | JAS M | New | TeamPulse F | |
| 322 | Implement data aggregation for multi-project burndown | JAS M | New | TeamPulse F | |
| 321 | Implement charting component (Velocity/Throughput) | JAS M | New | TeamPulse F | |
| 320 | Develop API endpoint for dashboard data | JAS M | New | TeamPulse F | |

https://dev.azure.com/jasminem1/TeamPulse Project/_workitems/edit/3...

Azure DevOps jasminem1 / TeamPulse Project / Boards / Work items

Search

TeamPulse Project

Overview

Boards

Work items

Boards

Backlogs

Sprints

Queries

Delivery Plans

Analytics views

Repos

Pipelines

Test Plans

Project settings

Work items

Recently updated

+ New Work Item

Open filtered view in Queries

Column Options

Filter by keyword

| ID | Title | Assigned To | State | Area Path |
|-----|---|-------------|-------|-------------|
| 341 | Implement '@' symbol autocomplete functionality | JAS M | New | TeamPulse F |
| 340 | Create the in-app notification component and badge count | JAS M | New | TeamPulse F |
| 339 | Develop the notification data model and API | JAS M | New | TeamPulse F |
| 338 | Implement a notification service to track work item changes | JAS M | New | TeamPulse F |
| 337 | Implement message storage and retrieval | JAS M | New | TeamPulse F |
| 336 | Design and implement the chat sidebar UI/UX | JAS M | New | TeamPulse F |
| 335 | Integrate a WebSocket service for real-time chat | JAS M | New | TeamPulse F |
| 334 | Create capacity calculation view | JAS M | New | TeamPulse F |
| 333 | Add 'Story Points' and 'Remaining Work' fields to templates | JAS M | New | TeamPulse F |
| 332 | Add task status change to history log | JAS M | New | TeamPulse F |

Backlog View:

Azure DevOps jasminem1 / TeamPulse Project / Boards / Backlogs

Search

TeamPulse Project

Overview

Boards

Work items

Boards

Backlogs

Sprints

Queries

Delivery Plans

Analytics views

Repos

Pipelines

Test Plans

Project settings

TeamPulse Project Team

+ New Work Item

View as Board

Column Options

Backlog Analytics

Order

Work Item Ty...

Title

Epic

Feature

Feature

Feature

Feature

Planning

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

TeamPulse Project Team backlog

Iteration 1

No work scheduled yet

Iteration 2

No work scheduled yet

Iteration 3

No work scheduled yet

+ New Sprint

TeamPulse Project

- Overview
- Boards
- Work items
- Boards
- Backlogs**
- Sprints
- Queries
- Delivery Plans
- Analytics views
- Repos
- Pipelines
- Test Plans
- Project settings

TeamPulse Project Team

Backlog Analytics

| | Order | Work Item Ty... | Title |
|----|------------|-----------------|--|
| 1 | User Story | > | Admin: Create User Accounts |
| 2 | User Story | > | Manager: Assign/Change User Roles |
| 3 | User Story | > | User: Secure Account Login |
| 4 | User Story | > | Manager: View Team Velocity KPI |
| 5 | User Story | > | Executive: View Multi-Project Burndown. |
| 6 | User Story | > | User: View Personal Task Summary |
| 7 | User Story | > | Project Lead: Create & Define Sprints |
| 8 | User Story | > | User: Update Task Status on Board |
| 9 | User Story | > | Project Lead: Estimate Effort (Points/Hours) |
| 10 | User Story | > | Team: Persistent Chat Sidebar |
| 11 | User Story | > | User: In-App Notifications |
| 12 | User Story | > | User: '@Mention' in Comments |

Stories

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

- TeamPulse Project Team backlog
- Iteration 1**
No work scheduled yet
- Iteration 2**
No work scheduled yet
- Iteration 3**
No work scheduled yet

+ New Sprint

TeamPulse Project

- Overview
- Boards
- Work items
- Boards
- Backlogs**
- Sprints
- Queries
- Delivery Plans
- Analytics views
- Repos
- Pipelines
- Test Plans
- Project settings

TeamPulse Project Team

Backlog Analytics

| | Order | Work Item Ty... | Title |
|------------|---------|-----------------|--|
| Epic | | ✓ | TeamPulse – Project Management Dashboard |
| Feature | | ✓ | User Authentication & Role Management |
| User Story | | > | Admin: Create User Accounts |
| User Story | | > | Manager: Assign/Change User Roles |
| User Story | | > | User: Secure Account Login |
| + | Feature | ✓ | Dashboard Analytics & KPIs |
| User Story | | > | Manager: View Team Velocity KPI |
| User Story | | > | Executive: View Multi-Project Burndown. |
| User Story | | > | User: View Personal Task Summary |
| Feature | | ✓ | Task & Sprint Management |
| User Story | | > | Project Lead: Create & Define Sprints |
| User Story | | > | User: Update Task Status on Board |

Stories

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

- TeamPulse Project Team backlog
- Iteration 1**
No work scheduled yet
- Iteration 2**
No work scheduled yet
- Iteration 3**
No work scheduled yet

+ New Sprint

TeamPulse Project

- Overview
- Boards
- Work items
- Boards
- Backlogs**
- Sprints
- Queries
- Delivery Plans
- Analytics views
- Repos
- Pipelines
- Test Plans
- Project settings

TeamPulse Project Team

Backlog Analytics

| | Order | Work Item Ty... | Title |
|------------|---------|-----------------|--|
| Feature | | ✓ | Dashboard Analytics & KPIs |
| User Story | | > | Manager: View Team Velocity KPI |
| User Story | | > | Executive: View Multi-Project Burndown. |
| User Story | | > | User: View Personal Task Summary |
| Feature | | ✓ | Task & Sprint Management |
| User Story | | > | Project Lead: Create & Define Sprints |
| User Story | | > | User: Update Task Status on Board |
| User Story | | > | Project Lead: Estimate Effort (Points/Hours) |
| + | Feature | ✓ | Team Chat & Notifications |
| User Story | | > | Team: Persistent Chat Sidebar |
| User Story | | > | User: In-App Notifications |
| User Story | | > | User: '@Mention' in Comments |

Stories

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

- TeamPulse Project Team backlog
- Iteration 1**
No work scheduled yet
- Iteration 2**
No work scheduled yet
- Iteration 3**
No work scheduled yet



https://dev.azure.com/jasminem1/TeamPulse Project/_workitems/edit/2...

Backlog View for New states:

The screenshot shows the Azure DevOps Backlog View for the TeamPulse Project Team. The backlog table includes columns for Title, State, Story Points, Value Area, and Iteration Path. The backlog is organized by type (Epic, Feature, User Story, Task) and state (New). The sidebar on the right provides an overview of the project's planning, including sprints, iterations, and a planning section for dragging work items.

| Work Item Type | Title | State | Story Points | Value Area | Iteration Path |
|----------------|---|-------|--------------|------------|-------------------|
| Epic | TeamPulse – Project Management Dashboard | New | | Business | TeamPulse Project |
| Feature | Team Chat & Notifications | New | | Business | TeamPulse Project |
| User Story | Team: Persistent Chat Sidebar | New | 20 | Business | TeamPulse Project |
| Task | Design and implement the chat sidebar UI/UX | New | | | TeamPulse Project |
| Task | Implement message storage and retrieval | New | | | TeamPulse Project |
| User Story | User: In-App Notifications | New | 30 | Business | TeamPulse Project |
| Task | Develop the notification data model and API | New | | | TeamPulse Project |
| Task | Create the in-app notification component and ba... | New | | | TeamPulse Project |
| User Story | User: '@Mention' in Comments | New | 20 | Business | TeamPulse Project |
| Task | Update notification service to specifically handle... | New | | | TeamPulse Project |

3. Write a short explanation

1. How Azure DevOps supports Agile methodology

Azure DevOps is a comprehensive set of cloud-based services that provides an integrated toolchain to support the planning, development, testing, and deployment phases of the software development lifecycle, aligning perfectly with Agile and DevOps methodologies.

It supports Agile principles by offering tools for:

Agile Planning and Tracking (Azure Boards)

Azure Boards is the primary component for Agile project management and directly supports popular frameworks like Scrum and Kanban:

- Work Item Management: Teams use work item types like Epics, Features, User Stories (or Product Backlog Items), Tasks, and Bugs to capture and track work, enabling the Agile practice of splitting work into smaller, manageable units.
- Backlogs: Teams can create and prioritize Product Backlogs (lists of user stories) and Portfolio Backlogs (lists of Features and Epics) to maintain a clear, prioritized product roadmap.
- Sprint Planning: Azure DevOps provides dedicated Sprint Backlogs and Taskboards that facilitate sprint planning, allowing teams to assign work items to specific iterations, estimate effort (Story Points), and define team capacity.
- Visualization and Monitoring:
 - Kanban Boards: Customizable boards help teams visualize the workflow, limit Work in Progress (WIP), and identify bottlenecks—core principles of Kanban.
 - Burndown Charts and Velocity Charts: These built-in analytics and reports provide real-time progress monitoring, promoting transparency and allowing for continuous process improvement.

Continuous Integration and Delivery (CI/CD)

Azure Pipelines directly enables the Agile practice of delivering working software frequently by automating the build, test, and deployment process:

- Continuous Integration (CI): Automates the process of merging developer code changes frequently into a central repository and running automated tests, catching integration issues early.
- Continuous Delivery (CD): Automates the release process to deploy software to various environments (like staging or production) quickly and reliably after testing, ensuring shippable increments at the end of every sprint.
- Infrastructure as Code(IaC): Supports managing and provisioning infrastructure as a development task, which is key to repeatable and rapid deployments.

2.Benefits of Breaking Large Projects into Smaller Work Items

Breaking large projects (Epics) into smaller, manageable items (Features, Stories, Tasks) offers several key benefits. It drastically **improves estimation accuracy** and **reduces risk** by making work tangible. This structure enhances **traceability**, ensuring every small task directly contributes to a prioritized user goal. Most importantly, it allows the team to **deliver value incrementally** and gather early feedback, which is the core principle of agile development.

- **Improved Estimation and Risk Reduction:** It drastically improves **estimation accuracy** and **reduces risk** by making the work tangible and manageable.
- **Enhanced Traceability:** The structure enhances **traceability**, ensuring every small task directly contributes to a prioritized user goal.
- **Incremental Value Delivery:** It allows the team to deliver **value incrementally** and gather early feedback, which is the core principle of agile development.

Additional Sections:

1. Iteration/Sprint Planning

Sprint 1:

The screenshot shows the Azure DevOps Boards interface for the TeamPulse Project. The left sidebar navigation includes: Overview, Boards, Work items, Boards, Backlogs, Sprints (selected), Queries, Delivery Plans, Analytics views, Repos, Pipelines, Test Plans, and Artifacts. The main area displays the TeamPulse Project Team backlog under the Backlog tab for Sprint 1. The backlog table has columns for Order, Title, State, Assigned To, and Tags. The backlog items are:

| Order | Title | State | Assigned To | Tags |
|-------|---|------------|-------------|--------------|
| 1 | > User: Secure Account Login | ● Resolved | JAS M | Front... ... |
| 2 | > Manager: View Team Velocity KPI | ● Resolved | JAS M | Analy... ... |
| 3 | > Executive: View Multi-Project Burndown. | ● Resolved | JAS M | Analy... ... |
| 4 | User: View Personal Task Summary | ● Closed | JAS M | Front... ... |
| 5 | Manager: Assign/Change User Roles | ● Closed | JAS M | Back... ... |
| 6 | > Admin: Create User Accounts | ● Closed | JAS M | Back... ... |
| 7 | > Project Lead: Create & Define Sprints | ● Active | JAS M | Back... ... |

To the right of the backlog, a planning board shows four sprints:

- Sprint 1:** Planned Effort: 200, 10/24/2025 - 10/30/2025 (7 tasks)
- Sprint 2:** Planned Effort: 40, 10/30/2025 - 11/3/2025 (2 tasks)
- Iteration 1:** No work scheduled yet
- Iteration 2:** No work scheduled yet

A planning overlay at the top right indicates the period from October 24 to October 30, with 4 work days remaining.

Sprint 1 Board:

The screenshot shows the Azure DevOps Boards interface for the 'TeamPulse Project Team'. The board is divided into four columns: New, Active, Resolved, and Closed. Each column contains a list of user stories with their respective story points, assignees, and tags.

| New | Active | Resolved | Closed |
|------------|--|--|---|
| + New item | <ul style="list-style-type: none">303 Project Lead: Create & Define Sprints Story Points: 20 Tags: Backend, Planning Assignee: JAS M | <ul style="list-style-type: none">299 User: Secure Account Login Story Points: 30 Tags: Frontend, Security Assignee: JAS M300 Manager: View Team Velocity KPI Story Points: 30 Tags: Analytics, Frontend, Reporting Assignee: JAS M301 Executive: View Multi-Project Burndown. Story Points: 30 Tags: Analytics, Frontend, Reporting Assignee: JAS M | <ul style="list-style-type: none">302 User: View Personal Task Summary Story Points: 30 Tags: Frontend, Usability Assignee: JAS M297 Admin: Create User Accounts Story Points: 30 Tags: Backend, Frontend, Security Assignee: JAS M298 Manager: Assign/Change User Roles Story Points: 30 Tags: Backend, Database, Security Assignee: JAS M |

2. Tags and Priorities

User Stories (Priority and Tags)

Parent Feature: User Authentication & Role Management

- Admin: Create User Accounts | **P1** | Tags: **Frontend, Backend, Security**
- Manager: Assign/Change User Roles | **P1** | Tags: **Backend, Database, Security**
- User: Secure Account Login | **P1** | Tags: **Frontend, Security**

Parent Feature: Dashboard Analytics & KPIs

- Manager: View Team Velocity KPI | **P1** | Tags: **Frontend, Analytics, Reporting**
- Executive: View Multi-Project Burndown | **P2** | Tags: **Frontend, Analytics, Reporting**
- User: View Personal Task Summary | **P2** | Tags: **Frontend, Usability**

Parent Feature: Task & Sprint Management

- Project Lead: Create & Define Sprints | **P1** | Tags: **Backend, Planning**
- User: Update Task Status on Board | **P1** | Tags: **Frontend, Usability**
- Project Lead: Estimate Effort (Points/Hours) | **P2** | Tags: **Planning, Database**

Parent Feature: Team Chat & Notifications

- Team: Persistent Chat Sidebar | **P2** | Tags: **Frontend, Real-time**
- User: In-App Notifications | **P1** | Tags: **Frontend, Backend, Real-time**
- User: '@Mention' in Comments | **P2** | Tags: **Frontend, Usability**

Azure DevOps TeamPulse Project / Boards / Backlogs

TeamPulse Project Team Backlog Analytics

Filter by keyword

| Order | Type | Title | State | Story Points | Value Area | Iteration Path | Tags |
|-------|------------|---|----------|--------------|------------|----------------------------|------------------------------|
| 1 | Epic | TeamPulse - Project Management Dashboard | New | | Business | TeamPulse Project | |
| 2 | Feature | User Authentication & Role Management | New | | Business | TeamPulse Project | |
| 3 | User Story | User: Secure Account Login | Resolved | 30 | Business | TeamPulse Project\Sprint 1 | Frontend Security |
| 4 | Task | Document password reset logic and flow | New | | | TeamPulse Project\Sprint 1 | Documentation |
| 5 | Feature | Dashboard Analytics & KPIs | New | | Business | TeamPulse Project | |
| 6 | User Story | Manager: View Team Velocity KPI | Resolved | 30 | Business | TeamPulse Project\Sprint 1 | Analytics Frontend Reporting |
| 7 | User Story | Executive: View Multi-Project Burndown | Resolved | 30 | Business | TeamPulse Project\Sprint 1 | Analytics Frontend Reporting |
| 8 | Feature | Task & Sprint Management | New | | Business | TeamPulse Project | |
| 9 | User Story | Project Lead: Create & Define Sprints | Active | 20 | Business | TeamPulse Project\Sprint 1 | Backend Planning Database |
| 10 | Task | Ensure work items can be assigned to the new s... | New | | | TeamPulse Project | Database Planning |
| 11 | User Story | Project Lead: Estimate Effort (Points/Hours) | Active | 20 | Business | TeamPulse Project\Sprint 2 | Database Planning |
| 12 | User Story | User: Update Task Status on Board | Active | 20 | Business | TeamPulse Project\Sprint 2 | Frontend Usability |
| 13 | Task | Develop API endpoint for status updates | New | | | TeamPulse Project | Backend |
| 14 | Task | Add task status change to history log | New | | | TeamPulse Project | Database |
| 15 | Feature | Team Chat & Notifications | New | | Business | TeamPulse Project | |
| 16 | User Story | Team: Persistent Chat Sidebar | New | 20 | Business | TeamPulse Project | Frontend Real-time |
| 17 | Task | Design and implement the chat sidebar UI/UX | New | | | TeamPulse Project | Frontend |
| 18 | Task | Implement message storage and retrieval | New | | | TeamPulse Project | Database |
| 19 | User Story | User: In-App Notifications | New | 30 | Business | TeamPulse Project | Backend Frontend Reporting |
| 20 | Task | Develop the notification data model and API | New | | | TeamPulse Project | Database |
| 21 | Task | Create the in-app notification component and b... | New | | | TeamPulse Project | Frontend Real-time |
| 22 | User Story | User: '@Mention' in Comments | New | 20 | Business | TeamPulse Project | Frontend Usability |
| 23 | Task | Update notification service to specifically handle... | New | | | TeamPulse Project | Backend |

Azure DevOps TeamPulse Project / Boards / Boards

TeamPulse Project Team Board Stories

View as backlog

| New | Active | Resolved | Closed |
|--|---|---|--|
| 306 Team: Persistent Chat Sidebar | 303 Project Lead: Create & Define Sprints | 299 User: Secure Account Login | 302 User: View Personal Task Summary |
| New | Active | Resolved | Closed |
| JAS M | JAS M | JAS M | JAS M |
| Story Points 20 | Story Points 20 | Story Points 30 | Story Points 30 |
| Frontend Real-time | Backend Planning | Frontend Security | Frontend Usability |
| 1/3 | 2/3 | 3/4 | 2/2 |
| 307 User: In-App Notifications | 305 Project Lead: Estimate Effort (Points/Hours) | 300 Manager: View Team Velocity KPI | 297 Admin: Create User Accounts |
| New | Active | Resolved | Closed |
| JAS M | JAS M | JAS M | JAS M |
| Story Points 30 | Story Points 20 | Story Points 30 | Story Points 30 |
| Backend Frontend Real-time | Database Planning | Analytics Frontend Reporting | Backend Frontend Security |
| 1/3 | 2/2 | 3/3 | 3/3 |
| 308 User: '@Mention' in Comments | 304 User: Update Task Status on Board | 301 Executive: View Multi-Project Burndown | 298 Manager: Assign/Change User Roles |
| New | Active | Resolved | Closed |
| JAS M | JAS M | JAS M | JAS M |
| Story Points 20 | Story Points 20 | Story Points 30 | Story Points 30 |
| Frontend Usability | Frontend Usability | Analytics Frontend Reporting | Backend Database Security |
| 1/2 | 1/3 | 3/3 | 3/3 |

3.Acceptance Criteria

For each user story, define short acceptance criteria:

Acceptance Criteria Table – TeamPulse Project

| Story ID | User Story | Acceptance Criteria (AC) |
|----------|-----------------------------------|--|
| 1.1 | Admin: Create User Accounts | <ol style="list-style-type: none">Only users with the 'Admin' role can access the user creation interface.System validates that required fields (Name, Email, Password) are not empty.A confirmation email is successfully sent to the new user upon creation. |
| 1.2 | Manager: Assign/Change User Roles | <ol style="list-style-type: none">Managers can only assign roles of 'Manager' or 'User' (cannot elevate to Admin).Role changes take effect immediately upon saving the user profile.The Role drop-down selection defaults to the user's current role. |
| 1.3 | User: Secure Account Login | <ol style="list-style-type: none">User is redirected to the main dashboard upon successful login.User must receive a generic error message if login fails (e.g., "Invalid credentials").Successful and failed login attempts are recorded for audit purposes. |
| 2.1 | Manager: View Team Velocity KPI | <ol style="list-style-type: none">Dashboard displays the Velocity (Story Points/Sprint) metric accurately.Data visualization shows the trend over the last 5 completed sprints.Data is filtered to show results for the current project or team. |
| 2.2 | Exec: View Multi-Project Burndown | <ol style="list-style-type: none">A dedicated view displays a burndown chart comparing ideal vs. actual remaining work.Chart accurately aggregates and displays data from all selected ongoing projects.Tooltips show the exact remaining work for any point on the chart. |
| 2.3 | User: View Personal Task Summary | <ol style="list-style-type: none">A "My Work" widget lists all tasks assigned to the current user.Tasks are prioritized and sorted by the closest due date.Completed tasks are hidden by default, with a clear toggle to view them. |

| | | |
|-----|--------------------------------------|--|
| 3.1 | Lead: Create & Define Sprints | <ul style="list-style-type: none"> 1. Interface allows setting Sprint Start Date, End Date, and Name. 2. Sprint dates are validated to prevent overlap with existing sprints. 3. The new sprint appears immediately in the iteration path selection for work items. |
| 3.2 | User: Update Task Status on Board | <ul style="list-style-type: none"> 1. Tasks can be dragged and dropped between all active Kanban columns. 2. Dragging a task updates its status field in the database immediately. 3. The status change is visible in the task's history log. |
| 3.3 | Lead: Estimate Effort (Points/Hours) | <ul style="list-style-type: none"> 1. Story Points field is visible and editable on the Story work item form. 2. Remaining Work (hours) field is visible and editable on the Task work item form. 3. The capacity view correctly sums the 'Remaining Work' for all tasks in a sprint. |
| 4.1 | Team: Persistent Chat Sidebar | <ul style="list-style-type: none"> 1. A real-time chat sidebar is accessible from all primary dashboard views. 2. Users can send and receive text messages with immediate display. 3. The chat history for the last 24 hours is preserved upon page refresh. |
| 4.2 | User: In-App Notifications | <ul style="list-style-type: none"> 1. A bell icon displays a badge count of unread notifications. 2. Notifications trigger for new assignments or status changes to a user's task. 3. Clicking the notification icon opens a list where items are marked as 'read' upon viewing. |
| 4.3 | User: '@Mention' in Comments | <ul style="list-style-type: none"> 1. Typing '@' in a comment field activates an autocomplete list of team members. 2. The tagged user receives a specific '@mention' notification. 3. The tag in the comment is clickable, linking to the mentioned user's profile. |

4. Definition of Done (DoD)

Definition of Done (DoD) for TeamPulse

For the TeamPulse Project, a work item (User Story or Task) is marked as Done only when it has satisfied all quality gate requirements. Specifically, this means the code has been peer-reviewed and approved by at least one other developer, all automated unit and integration tests are passing, and the feature meets all Acceptance Criteria during dedicated QA testing. Finally, any relevant technical or user documentation must be updated and approved, and the functionality must be successfully deployed and confirmed stable in the Staging environment. This comprehensive definition ensures high quality and readiness for production release.

DoD is comprehensive, covering:

1. **Code Quality:** Peer Review
2. **Testing:** Automated Tests and QA Acceptance
3. **Documentation:** Updated Technical/User Docs
4. **Deployment Readiness:** Stable in Staging Environment

5. Burndown Chart

