



AZ-400 Domain 1: Processes and Communications

AZ-400 skills measured, domain 1 (about 10-15 %, as of 2026) | Work tracking in Azure Boards and GitHub, traceability, flow and DORA metrics, dashboards, wikis and documentation, notifications and chat integrations

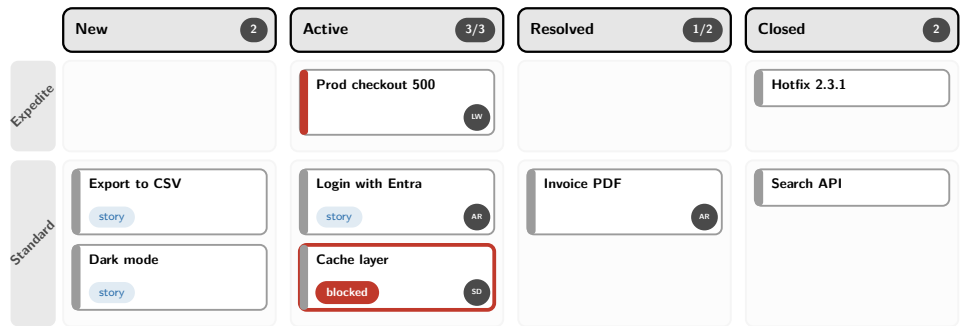
AZ-400 Domain 1 Azure Boards GitHub Projects DORA

Work tracking: Azure Boards and GitHub

Process	Backlog levels	Requirement item	States of the requirement	Choose it when
Basic	Epic → Issue → Task	Issue	To Do, Doing, Done	small team, simplest tracking
Agile	Epic → Feature → User Story → Task (+ Bug)	User Story	New, Active, Resolved, Closed	story points, flexible teams
Scrum	Epic → Feature → Product Backlog Item → Task	PBI (effort)	New, Approved, Committed, Done	Scrum teams, sprint commitment
CMMI	Epic → Feature → Requirement → Task (+ Change Request, Review, Risk)	Requirement	Proposed, Active, Resolved, Closed	audited, formal change control

EXAM TIP

Customise a system process? You cannot edit Agile/Scrum/CMMI directly: create an **inherited process**, add the field or state there, then change the project's process to it. A **custom rule** (when field X changes, make Y required) and **custom work item types** live in the inherited process too.



Azure Boards Kanban board: WIP limits per column, an Expedite swimlane for incidents, a blocked card flagged (tag Blocked)

Azure Boards features to know	GitHub planning features
Backlogs and sprints: capacity per person, sprint burndown	Issues with labels, assignees, milestones, sub-issues, issue types
Kanban board: columns map to states, WIP limits, split columns (Doing / Done), definition of done per column, swimlanes	GitHub Projects: table, board and roadmap views, custom fields (iteration, single select, number), built-in automations
Queries (WIQL): flat, tree, direct links; charts pinned to dashboards	Issue forms and templates in .github/ISSUE_TEMPLATE
Delivery Plans: several teams on a timeline, dependencies	Closes #12 in a PR closes the issue on merge
Area paths per team, iteration paths per sprint	Discussions for Q&A and announcements
Tags, custom fields, rules in an inherited process	Azure Boards app: AB#123 links GitHub commits and PRs to work items

Mention	Effect
git commit -m "Fix rounding AB#123"	GitHub commit linked to Azure Boards item 123 (Azure Boards app installed on the repo)
Fixes AB#123 (in PR description)	on merge of the GitHub PR, item 123 moves to its Done/Closed state
#123 (Azure Repos commit or PR)	links Azure Repos commit or PR to work item 123
Closes #45 (GitHub PR)	closes GitHub issue 45 when the PR merges into the default branch
az boards work-item create --type Bug --title "500 on checkout"	create a work item from the CLI (azure-devops extension)
az boards query --wiql "SELECT [System.Id] FROM WorkItems WHERE [System.State] = 'Active'"	run a WIQL query

Traceability end to end

- Work item** ↔ **branch** create the branch from the work item (Development section) or name it users/ana/123-fix
- Commit / PR** ↔ **work item** #123 (Azure Repos) or AB#123 (GitHub); branch policy *Check for linked work items* makes it mandatory
- Build** ↔ **work items** the build summary lists commits and associated work items since the last successful build
- Release / environment** ↔ **work items** environment deployment history shows the commits and work items deployed; YAML Integrated in build field set on work items

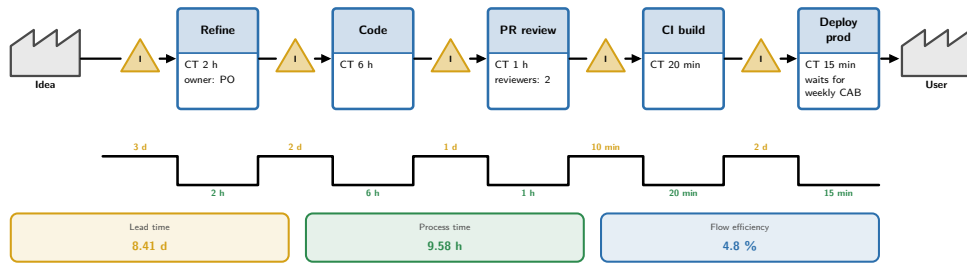
- 5 **Deployment** ↔ **telemetry** release annotations in Application Insights mark each deploy on the charts
- 6 **Test results** ↔ **requirement** Test Plans test cases linked to user stories; requirement-based suites show coverage

What links to what (the exam's traceability chain)

💡 Release notes from traceability

Generate release notes in the pipeline from the work items and PRs between two builds: the Azure DevOps REST API (`_apis/build/builds/{id}/workitems`), a marketplace task, or on GitHub the **generated release notes** (`gh release create v2.4.0 --generate-notes`) driven by `.github/release.yml` label categories.

☉ Flow and DORA metrics



Value stream of one change: most of the lead time is waiting (queues before refinement, review and release)

Flow efficiency of the stream above

FORMULA **FE = PT / LT × 100**

WHERE

- FE = flow efficiency, share of lead time spent working (%)
- PT = process (touch) time: sum of cycle times of the steps (hours)
- LT = lead time: process time plus every wait (hours)

WORKED

PT = 9.58
 LT = 201.75
 $FE = 9.58 / 201.8 \times 100$
 FE = 4.748 %

RESULT FE = 4.748 %

READ IT Under 5 %: the change waits 95 % of the time; remove queues (weekly CAB → automated approval, smaller PRs) before speeding up the build.

MISTAKES

- Lead time and cycle time are not the same: cycle time starts when work starts
- Measure medians or 85th percentiles, not averages skewed by one old item

📄 Worked example: the numbers behind the card

Process time PT = 2 + 6 + 1 + 0.33 + 0.25 = 9.58 h. Waits = 3 d + 2 d + 1 d + 0.17 h + 2 d = 8 d = 192 h (24-hour days) + 0.17 h. LT = 9.58 + 192.17 = 201.75 h ≈ 8.4 days. FE = 9.58 / 201.75 × 100 = 4.75 %. Replacing the weekly CAB with an automated approval check cuts 2 days: LT = 153.75 h, FE = 6.2 %, lead time down 24 %.

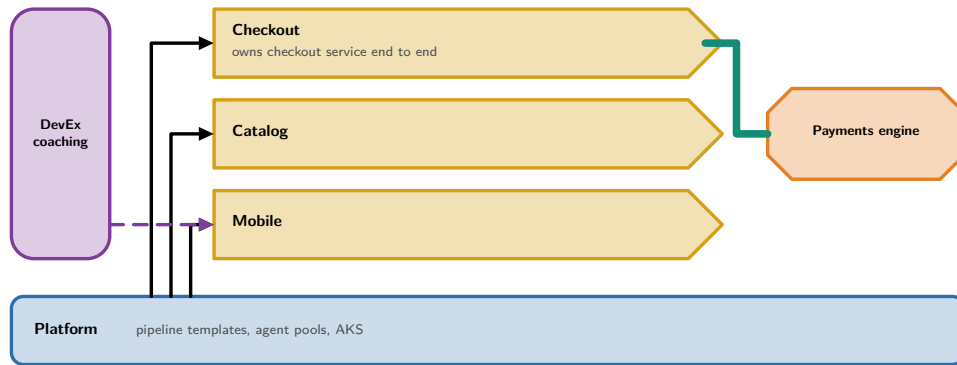
DORA metric	What it measures	Elite (DORA re-port)	Where to get it on Azure
Deployment frequency	how often production gets a deploy	on demand, several per day	environment deployment history, pipeline runs to prod
Lead time for changes	commit → running in production	less than one day	commit time vs deployment job end time
Change failure rate	share of deployments causing a failure (rollback, hotfix, incident)	low, about 0-15 %	deployments linked to bugs / incidents
Time to restore service	incident start → service restored	less than one hour	incident tool, Azure Monitor alert to resolve

Azure Boards chart or widget	Answers
Cumulative flow diagram	Where does work pile up? Widening band = bottleneck; WIP = vertical distance
Lead time / cycle time widgets	How long items take, scatter with 85th percentile trend
Velocity	How much the team completes per sprint (story points or count)
Sprint burndown	Will we finish the sprint scope? Remaining work vs ideal line
Burnup	Scope growth and completed work in the same chart (Analytics view)
Test results trend, pipeline pass rate	Quality and pipeline reliability over time
Deployment status	Which build is in which environment

📄 INFO

Dashboards are per team or per project; widgets read Analytics (OData). Power BI connects to Azure DevOps Analytics through **Analytics views** or OData queries (https://analytics.dev.azure.com/{org}/{project}/_odata/v4.0-preview/WorkItems) for custom DORA reports. GitHub: Insights (pulse, contributors, traffic), Projects charts, and the REST/GraphQL API for custom metrics.

✉ Teams, communication and documentation



■ Stream-aligned
 ■ Platform
 ■ Enabling
 ■ Complicated subsystem
→ Collaboration
 → X-as-a-service
 - - - Facilitating

Team Topologies view: the platform team serves pipeline templates as a product; a shared YAML template repo is its API

Integration	How to set it up	Typical events
Microsoft Teams	Azure Boards / Azure Pipelines apps for Teams; @azure pipelines subscribe <url>	run completed, approval pending (approve in chat)
Slack	Azure Pipelines / Boards apps for Slack; GitHub app /github subscribe org/repo	builds, PRs, issues
Service hooks	Project settings → Service hooks → subscription with filters	build completed, work item updated, PR created → webhook, Teams, Jenkins, Azure Service Bus
Notifications	Personal / team / project subscriptions, email or SOAP	work item assigned, build failed, PR reviewer added
GitHub webhooks	Repo / org settings → Webhooks, secret signs payload (X-Hub-Signature-256)	push, pull_request, workflow_run, release

Azure DevOps wikis	GitHub documentation
Project wiki: one per project, stored in a hidden Git repo, edited in the portal	README, CONTRIBUTING, CODEOWNERS, SECURITY.md in the repo
Code wiki (publish code as wiki): any repo folder of Markdown, per branch/version, reviewed by PR	Wiki per repository (a separate Git repo repo.wiki.git)
Mermaid diagrams with <code>::: mermaid ... ::;</code> <code>[[_TOC_]]</code> table of contents; @mention and #123 links	Mermaid in Markdown with a <code>'''mermaid'''</code> code block
Page order kept in a .order file per folder	GitHub Pages to publish docs sites from a branch or an Actions workflow

Wiki page with a Mermaid diagram (Azure DevOps syntax)

```
[[_TOC_]]
# Release process
::: mermaid
graph LR
  PR[Pull request] --> CI[CI build]
  CI --> DEV[Dev] --> PROD[Prod approval]
:::
Linked work item: #123 Owner: @Ana Ruiz
```

✗ Documentation that lives outside the code

Architecture notes in a portal wiki drift from the code. Keep design docs and ADRs (architecture decision records) in the repo (docs/adr/0007-use-trunk-based.md), publish the folder as a code wiki, and review them in the same PR as the change.

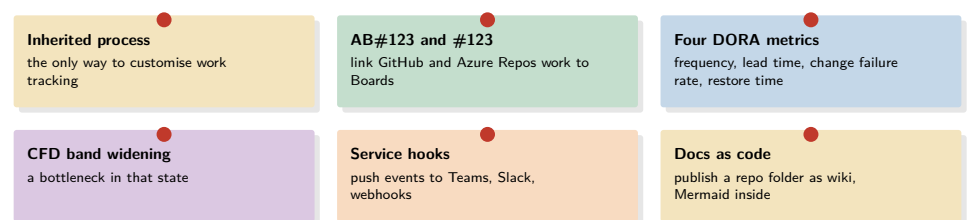
🚩 Recap and self-test

MYTH

- ✗ A higher velocity means a faster team
- ✗ Lead time and cycle time are synonyms
- ✗ Faster builds are the best lever on delivery speed
- ✗ You can edit the Scrum process to add a field

FACT

- ✓ Velocity is a planning aid for one team, not a productivity score
Story points are relative; comparing teams or targeting velocity inflates estimates.
- ✓ Lead time starts at creation, cycle time at start of work
Azure Boards widgets compute both; their gap is the backlog queue.
- ✓ Waits dominate lead time
Flow efficiency is often under 15 %: remove queues and approvals that wait for meetings first.
- ✓ System processes are locked; customise an inherited process
Create it from Scrum, add the field, switch the project to it.



Q1 A team wants a mandatory Risk field on User Stories. What do you do?

A Create an inherited process from Agile, add the field with a required rule, and assign the project to that process.

Q2 Which chart shows a bottleneck in the Active state?

A The cumulative flow diagram: the Active band widens over time.

Q3 How do you link a GitHub commit to Azure Boards item 512?

A Install the Azure Boards app on the repo and write AB#512 in the commit message or PR.

Q4 Which view shows several teams' backlogs with dependencies on one timeline?

A Azure Boards Delivery Plans.

Q5 Difference between lead time and cycle time?

A Lead time runs from creation to done; cycle time from first active state to done.

Q6 Name the four DORA metrics.

A Deployment frequency, lead time for changes, change failure rate, time to restore service.

Q7 How do you get pipeline approvals into Microsoft Teams?

A Install the Azure Pipelines app for Teams and subscribe the channel to the pipeline; approvals can be done from the card.

Q8 How do you draw a diagram in an Azure DevOps wiki page?

A A Mermaid block between ::: mermaid and :::.

Q9 How do you make linking a work item mandatory on PRs in Azure Repos?

A Branch policy Check for linked work items set to Required.

Q10 Which Azure DevOps feature posts events to an external webhook?

A A service hook subscription (Project settings → Service hooks).