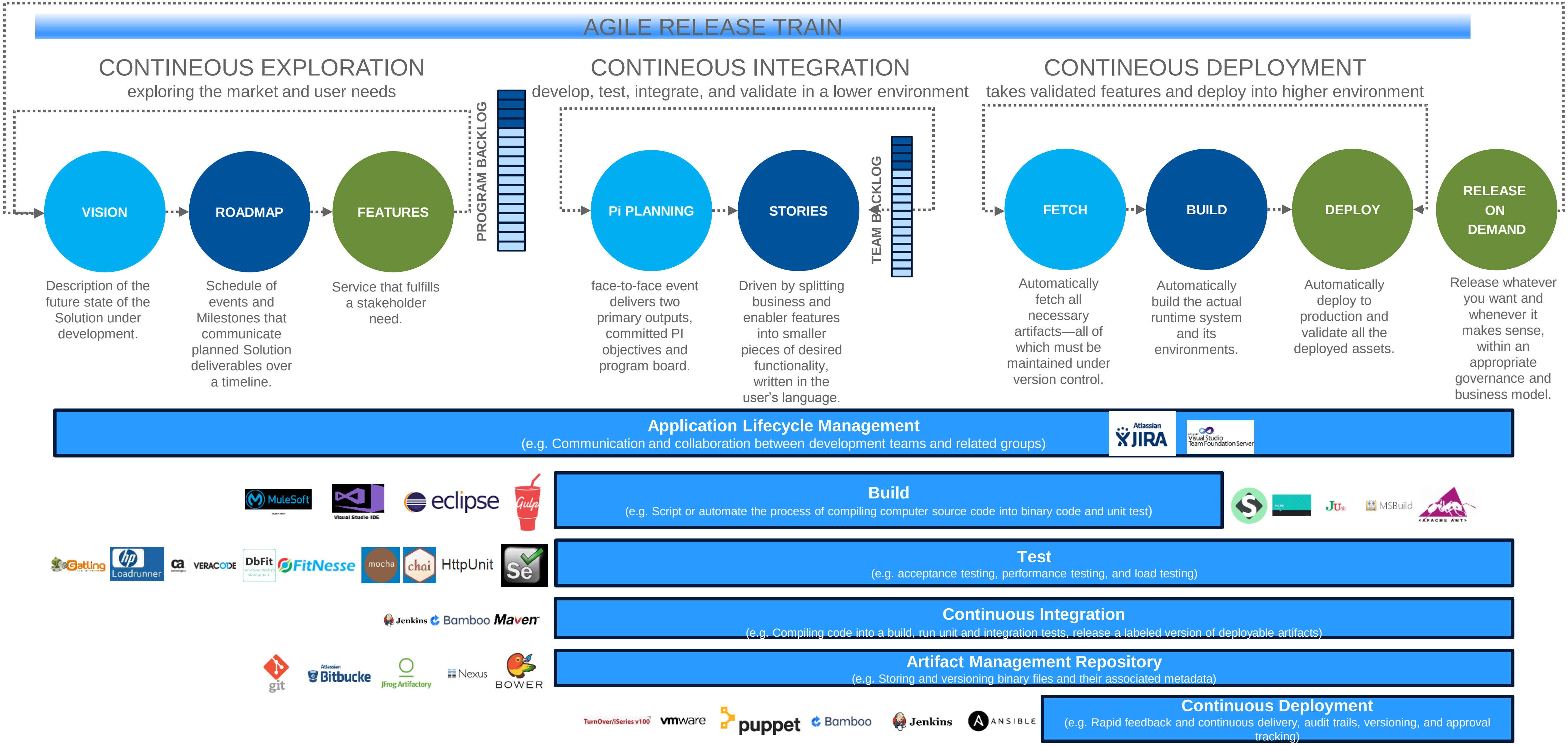
CONTINEOUS DELIVERY PIPELINE

Workflows, activities, and automation needed to provide a continuous release of value to the end user.



AGILE RELEASE TRAIN

CONTINEOUS EXPLORATION

exploring the market and user needs

VISION: Description of the future state of the Solution under development.

ROADMAP: Schedule of events and Milestones that communicate planned Solution deliverables over a timeline.

FEATURES: Service that fulfills a stakeholder need.

PROGRAM BACKLOG

CONTINEOUS INTEGRATION

develop, test, integrate, and validate in a lower environment

Pi PLANNING: face-to-face event delivers two primary outputs, committed PI objectives and program board.

STORIES: Driven by splitting business and enabler features into smaller pieces of desired functionality, written in the user's language.

TEAM BACKLOG

CONTINEOUS DEPLOYMENT

takes validated features and deploy into higher environment

FETCH: Automatically fetch all necessary artifacts—all of which must be maintained under version control.

BUILD: Automatically build the actual runtime system and its environments.

DEPLOY: Automatically deploy to production and validate all the deployed assets.

RELEASE ON DEMAND: Release whatever you want and whenever it makes sense, within an appropriate governance and business model.

Application Lifecycle Management
(e.g. Communication and collaboration between development teams and related groups)

Build
(e.g. Script or automate the process of compiling computer source code into binary code and unit test)

Test
(e.g. acceptance testing, performance testing, and load testing)

Continuous Integration
(e.g. Compiling code into a build, run unit and integration tests, release a labeled version of deployable artifacts)

Artifact Management Repository
(e.g. Storing and versioning binary files and their associated metadata)

Continuous Deployment
(e.g. Rapid feedback and continuous delivery, audit trails, versioning, and approval tracking)