Planview vs GitLab

GitLab compared to other DevOps tools

Planview is an IT project planning and management tool designed to help enterprise teams organize, plan, priortize and excecute large agile projects and programs. It facilitates capturing and priortizing demand from business teams, allocating resources and managing execution through to completion.

FEATURES	PLANVIEW	
Portfolio Planning Establishing strategic priorities and direction in order to govern the allocation of corporate resources to support specific business/IT initiatives. Strategic planning evaluates in-flight projects and proposed future initiatves to shape and govern the ongoing investment in projects and discretionary work. Able to model and optimize different portfolio investment scenarios to determine the ideal funding combinations to meet strategic priorities. Specific features would include: Proposals, epics, backlog, strategic alignment, estimation, prioritization, what-if, monte-carlo simulation, optimization.	✓	−
Resource/Team management Tracks and manages the availability of team members by skill, experience, location, and cost, so they can support both planned and unplanned work. Specific features would include: individual capacity, individual skills, individual assignments, labor cost.	✓	✗
Financial management Tracking, managing and reporting on the budget and actual spend of projects and programs within specific portfolios. Able to allocate costs to OPex or CAPex depending on specific organization reporting rules. Time tracking information used to determine labor cost allocations. Specific features would include: budget, spend, time, resource cost.	✓	✗
Portfolio status management Tracks and reports on the overall status and health (scope, schedule, budget) of projects and programs within the portfolio to enable executives to support project execution. Specific features would include: forecasting, status tracking, release planning, roadmap, milestones, project/program hierarchy.	✓	−

SAFe - Lean Portfolio Provides support for the key elements of [Lean Portfolio Management(https://www.scaledagileframework.com/lean-portfolio-management/). Specific features would include: Release Train, Lean Portfolio Management.	⊖	✗
Portfolio Management Plan and track work at the project and portfolio level. Manage capacity and resources together with Portfolio Management. Learn more about Portfolio Management	✓	✓
Product Roadmap Establishing the **product** vision and strategy to organize, govern and shape the effort of the multi-disciplinary team building specific business services. The Product Roadmap is based on specific business goals and objectives, manages high level requirements, prioritizes future features, allocates resources, tracks progress and measures business results linked to the business strategy. Specific features would include: Product Strategy, Idea Gathering, Requirements Management, Resource Management, Feedback Collection, Milestone Tracking, Release Management, Workflow Management, Feature Tracking/Deviation, Roadmap Analytics, Centralized Notes, Report Generation, Imports, API/SDK, Conflict Management, and Customized Alerting.	✓	✗
Work Planning/ Management Able to define, schedule and assign specific tasks to team members and manage the sequence and interdependency of tasks with each other. This form of structured planning is needed when tasks are clearly defined and sequence of executution is critical. Specific features would include: WBS, Gantt Chart, Task Assignment, Scheduling, task sequence, task relationships.	✓	✗
Backlog Management Able to capture and track future features, capabilities and work in a consolodated and organized list which enables the team to organize, prioritize, accept, plan and start work on relevant items. The backlog is where future work is captured, defined, evaluated, and planned. Specific features would include: Backlog, user stories, issues, effort estimate, priority, backlog board.	✓	✓
Scrum Able to support the time-boxed (sprint) approach of the Agile -Scrum software delivery. Specific features would include: Issues, scrum boards, burndown charts, burn up.	✓	✓
Kanban Able to support the flow based approach of Agile - Kanban software delivery. Specific features would include: Issues, Kanban boards, burn up, cumulative flow diagram.	✓	✓

SAFe Able to support the key principles and practices of the [Essential SAFe configuration.](https://www.scaledagileframework.com/essential-safe/), Agile Release Train, (a combined CD pipeline where multiple projects allign to release), Planning Increment (PI) Planning (cadence driven project planning/ vision)	✓	✗
Project Financial Management Tracking, managing and reporting on the budget and acutal costs of the project. Able to allocate costs to OPex or CAPex depending on specific organization reporting rules. Time tracking information used to determine labor cost allocations. Specific features would include: budget, spend, time tracking, resource cost, Capex/Opex.	✓	✗
Risk/Issue Management Able to define and manage project RISK and ISSUE status and workflow to identify, track, mitigate and resolve potential risks and active issues facing the project. Risk/Issue status and resolution managed through online workfow that tracks assignment and actions to address the specific item. Specific features would include: workflow, risk severity, risk priority, Assigning Risk/Issue for action, Risk/Issue status.	✓	−
Project Time Tracking Able to capture individual time reporting for specific assigned tasks and then to allocate labor costs to the appropriate project. Specific features would include: estimate, actual, cost, reporting.	✓	−
DevOps Pipeline Able to establish visiblity into the end to end DevOps pipeline so the entire team is aware of pipeline status and can contribute to overall success. Specific features would include: visibility into status of pipeline	−	✓
Requirements Management Able to gather, document, refine and track approval of business and system requirements. Managing and tracking the relationships between requirments and other requirements, requirements and code, requirements and test cases for each version of requirements. Specific features would include: definition, traceability, requirement hierarchy, dependency.	✗	−
Quality Management Able to support test planning, definition of test cases, results of test execution and corresponding backlog of work resulting from failed tests. Specific features would include: Test case planning, test execution, defect tracking (backlog), severity, priority.	✗	−