Project Planner

Frameworks to Support Continuous Delivery

Not Started	In Progress	Completed	Blocked					

Critical Resolution	Critical Path	Incomplete	Critical (Incomplete)	Non-critical (Incomplete)				

DEPTHS	HIGHER/REQUIREMENT	DESCRIPTION	OWNER	START/SLIPPAGE	FINISH/SLIPPAGE	ACTUAL WORK	ACTUAL DURATION	FORECAST COMPLETION
0	0/1	Project scoping	Team	10/19	30	171	147	100%
0/1	0/1	Initial meeting with product for requirements	Team	10/5	7	10/19	5	100%
0/1	0/1	Design notes for team	Team	10/19	5	10/24	4	100%
0/1	0/1	Create project description and context diagram	Everyone	10/24	2	10/25	2	100%
0/1	0/1	Write system requirements	Team	10/26	4	10/31	2	100%
0/1	0/1	Create project planner with planned and actually completed work	Paul	10/26	2	10/31	2	100%
Feature 1.1	FR 1	Extract connections, grid directory		10/31	2			0%
Feature 1.2	FR 2	Analyze and record classes (interfaces, effected by such a change)		10/27	5			0%
Feature 1.3	FR 3	Process and record the amount of code changed with each commit		11/3	3			0%
Feature 1.4	FR 4	Record results of the last series each time it runs		11/6	2			0%
Feature 1.5	FR 5	Backlog to produce impact previous releases		11/7	3			0%
Feature 1.6	FR 6	Record issues (fixed for this code base		11/10	3			0%
Feature 1.7	FR 7	Connect issues to their resolution and make an inference from the		11/13	3			0%
Feature 1.8	FR 8	Allow parsing from diverse options (Jaxer, nasagenga, tscitlint) reports,		11/16	5			0%
Feature 1.9	FR 9	Normalize and combine SA data into a central data model		11/16	5			0%
Feature 1.10	FR 10	Normalize and combine test data into a central data model		11/23	3			0%
Feature 1.11	FR 11	Normalize and combine issue data into a central data model		11/3	3			0%
Feature 1.12	FR 12	In the datamodel Gauge out risks wards of the code and much those		1/3	3			0%
Feature 1.13	FR 13	Identify lines frequently that are higher risk (churn/churn)		1/7	7			0%
Feature 1.14	FR 14	Identify impact where a risk is change (file/module/code/ugly/reflecs)		1/14	7			0%
Feature 1.15	FR 15	Create a PoR report for stakeholders to see progress in codebase		1/21	10			0%
Feature 1.16	FR 16	Create a Metric Gate for exploring the risk within codebase		1/34	5			0%
Feature 1.17	FR 17	Create GitHub, JFFT and Slack integrations for viewing Framework data		1/14	5			0%
Feature 1.18	FR 18	Create tool integration on developer's an are / branching when		1/34	5			0%
Feature 1.19	FR 19	Develop command line interface to the Framework		1/34	7			0%
Feature 1.20	FR 20	Research deployment frameworks and integrate risk analysis / recipes		1/17	7			0%
Feature 1.21	FR 21	Develop VSM plugin for viewing risk rating (Fusade method) class		1/21	5			0%
Feature 1.22	FR 22	Develop JIRA plugin for viewing risk rating (Fusade method) class		1/24	5			0%
Feature 1.23	FR 23	Make the Framework easily deployable / into a microservice		1/17	5			0%
Feature 1.24	FR 24	Reduce dependencies and distribute a minimally intrusive product		1/34	5			0%