



## Release Engineering Tech Debt Analysis - Overview - Software

<b>Software</b>	Level of use	Active maintainers	Code review	Code quality (coding standards, testability, refactorability, inline/API docs, etc.)	End-user doc	Release/deployment process	Test coverage	Test environment	Configuration management	Fault tolerance	Monitoring	Troubleshooting
	<i>Rank (0-5) 0 = almost no one 5 = just about everyone</i>	<i>Rank (0-5) 0 = worst 5 = best</i>	<i>Rank (0-5) 0 = worst 5 = best</i>	<i>Rank (0-5) 0 = worst 5 = best</i>	<i>Rank (0-5) 0 = worst 5 = best</i>	<i>Rank (0-5) 0 = worst 5 = best</i>	<i>Rank (0-5) 0 = worst 5 = best</i>	<i>Rank (0-5) 0 = worst 5 = best</i>	<i>Rank (0-5) 0 = worst 5 = best</i>	<i>Rank (0-5) 0 = worst 5 = best</i>	<i>Rank (0-5) 0 = worst 5 = best</i>	<i>Rank (0-5) 0 = worst 5 = best</i>
<b>Meta (applies to all of the above)</b>												



## Release Engineering Tech Debt Analysis - Overview - Services

	Fault tolerance	Documentation	Bus factor # of active maintainers	Minimum level of service (how does a service fail?)	Dependencies on other teams / upstreams (does an external dependency currently impede this process?)	Complexity of the process (Users and us maintaining)	Safety of the process	How automated is a process
<b>Services (not directly software)</b>	<i>Rank (0-5) 0 = worst 5 = best</i>	<i>Rank (0-5) 0 = worst 5 = best</i>	<i>Rank (0-5) 0 = worst 5 = best</i>	<i>Rank (0-5) 0 = worst 5 = best</i>	<i>Rank (0-5) 0 = worst 5 = best</i>	<i>Rank (0-5) 0 = worst 5 = best</i>	<i>Rank (0-5) 0 = worst 5 = best</i>	<i>Rank (0-5) 0 = worst 5 = best</i>
Repository administration (Phab+Gerrit)	N/A	3	2	N/A	5	2	3	1