

#	Type	Task (Description)	Motivation and Goal(s)	Artifacts involved				Task Character.			
				R	TC	I	S	P	C	E	F
39	Informative	Show architectural principles, plans, drawings & solution concepts; Technical roadmap. Interesting functionality: Locate software elements of the architecture; Telescope the architectural structures	Product management	-	-	-	-				
40	Control/Perfective	Statistics on system quality	Quality assurance and control	-	-	-	-				
41	Evaluative	Statistics on complexity (size) of system, services / components	Prioritization of services/components for implementation and testing.	-	-	-	✓				
TEST MANAGER											
42	Evaluative/Estimative	Show requirements info: structure & content, prioritization, state of requirements (in work, in test, accepted); number of changed, added, deleted, modified requirements.	To prioritize requirements for testing; Test effort (re)estimation.	✓	-	-	-				
43	Evaluative/Estimative	Show change requests	To prioritize change requests; Test effort (re)estimation	✓	-	✓	-				
44	Estimative/Evaluative	Show Risk-related information: requirements/components/services of a specific risk level at a specific point in time; requirements/features for which at least one bug of a specific severity level is open (e.g. blocker/critical/major); all bugs of requirements/component/service of a specific risk level at a specific point in time; Test/Bugs history of a requirement (show all bugs by severity)	Prediction of failure; To prioritize requirements for testing; Test effort (re) estimation	✓	-	✓	✓				
45	Evaluative/Estimative	Show criticality or business value of features in production.	To prioritize features for testing; Test effort (re)estimation.	✓	-	-	✓				
46	Estimative/Evaluative	Show Test-related information: test plans; test schedule; Top level test scenarios linked to requirements; Load testing plans; Test setup (from Software Architect);	Test effort and cost estimation; to derive Critical scenarios for Load Performance Tests;	-	✓	-	-				
47	Estimative	Test Cases overview: coverage of requirements/services/components, priority for tests, costs, duration, dates when test cases can be performed	To build cost estimation, also deadlines and effort.	✓	✓	-	-				
48	Control/Evaluative	Show: requirements covered by a specific test set, tests for a specific set of requirements (eg. requirements of one service);	Test management	✓	✓	-	✓				
49	Evaluative/Control	Show requirements/components/services for which at least one bug of a specific severity level (e.g. blocker/critical/major) is open.	To prioritize requirements for testing; Test prioritization.	✓	-	✓	✓				
50	Evaluative	Executable tests assigned to requirement(s).	Test prioritization	✓	✓	-	-				
51	Evaluative/Estimative	Show tests for a specific set of requirements (e.g. component, service);	Test prioritization, Effort estimation for components/services	✓	✓	-	✓				
52	Estimative/Evaluative	Show risk mitigation (by testing) over time, depending on the risk level (requirements are adequately covered by tests which should all pass) - (visualized in a risk burndown chart);	Test effort and cost (re)estimation; Reprioritization of tests to meet the deadlines; Evaluation of Risk mitigation and Test strategies.	✓	✓	-	-				
53	Estimative/Evaluative	Test history of a requirement (also called test progress report); Bug history of a requirement (open bugs, closed bugs, new bugs for each test cycle);	Test effort and cost (re)estimation; Test prioritization	✓	✓	✓	-				
54	Estimative/Evaluative	Tests that have to be executed after changes of specific requirement(s)/service/component.	Test effort and cost (re)estimation; Reprioritization of tests to meet the deadlines.	✓	✓	-	✓				
55	Estimative/Evaluative	Show: results of pretests (coming from IT analyst or software architect); test progress, results (failed, passed, blocked, executed), defects (by criticality level) (open, in work, resolved, tested, closed), number of test cases, specified vs. number of tests (cases) to be specified, progress of test case specifications, number of critical defects per component; test effectiveness, measured by number of failures (under consideration of severity and risk level) found by a specific requirements/component/service test.	Test management: Test effort and cost (re)estimation; Test prioritization; Evaluation of Risk mitigation and Test strategies.	✓	✓	✓	✓				
56	Evaluative/Diagnostic/Corrective	Show runtime fail probability: probability that a deployed requirement/component/ service fails at runtime.	To derive weak requirements/services/components for refactoring; Test prioritization.	✓	-	-	✓				
57	Evaluative/Control	Show traceability to/from: use case to test cases, test cases to code, code to bug reports, code coverage reports, requirement/ test case/ MKS.	Test management	✓	✓	✓	-				
58	Estimative/Evaluative	Show Test/Bugs history: bug history of a requirement/component/service (open bugs, closed bugs, new bugs - for each test cycle, prioritize by severity).	Test effort and cost (re)estimation; Test prioritization	✓	✓	✓	✓				
59	Evaluative/Diagnostic	Show all bugs (by severity levels) of a requirement/component/service of a specific risk level at a specific point in time.	Test prioritization	✓	-	✓	✓				
60	Estimative/Corrective	Show list of services with: deployments, state of service, number of critical defects per service (so to know whether service is OK or not) - only critical defects are important.	Prediction of failure; Effort (re)estimation.	-	-	-	✓				
61	Evaluative/Estimative	Show list of Issues with: tickets severity, priority, status of the issue (open, assigned, tested, resolved, closed), assigned person for the issue	Test prioritization; Effort (re)estimation	-	-	✓	-				
62	Diagnostic/Corrective	Show CRs & P(A)Rs (Change requests & problem reports)	Test/Issue Management	✓	-	✓	-				
63	Control/Evaluative	Show QA goals – acceptable number of defects or risk of BV – and QA budget	Quality Assurance	-	-	-	-				
64	Diagnostic/Evaluative/Corrective	Show Bug reports from Clients	Test management, test (re)prioritization	-	-	✓	-				
65	Estimative	Show features of the final (structured by business functionality from perspective of user roles).	Test management, Effort estimation	-	-	-	✓				
66	Estimative	Show: Project and Development plans (based on functional features); Milestones, Budget; Work pages list; Risk list; Status reports; State of the product (aggregated or state of the components based on defects).	Test manager produces and sends Test/Implementation/Customization effort and cost estimation, deadlines and plan on test cycles.	✓	-	-	✓				
67	Estimative/Evaluative	Show details on hardware architecture: hardware architecture with performance statistics; traceability link to/from architecture/coverage reports.	Test manager needs the architectural plans as well for the integration and acceptance testing.	✓	-	-	✓				
68	Estimative/Corrective	Show System reliability and Runtime fail probability: the reliability (measured by mean time between two runtime failures) of a system/feature/ requirement; the probability that a deployed requirement or component/service fails at runtime;	Prediction of failure.	✓	-	-	✓				
SOFTWARE ARCHITECT											
69	Estimative/Consultive	Show: List of requirements (customer, architecture, maintainability,...)*, QA goals, NF requirements, prioritized requirements.	SA derives: (Rough) Estimates on Effort & Feasibility (feasibility of the Architecture (levels)); Vision for future development budget (how large to build?).	✓	-	-	-				
70	Diagnostic/Groomative	Show: requirements change requests and orders to Software architect; test concept - from technical perspective.	Helps to test non functional requirements, sometimes also to automate tests.	✓	✓	✓	-				
71	Diagnostic/Estimative/Corrective	Show: test plans; load testing reports.	To derive Failure scenarios (when will fail, what kind of failure stress?)	-	✓	✓	-				
72	Diagnostic/Evaluative/Corrective	Show report on defects by architectural component (defects grouped/sorted by architectural component).	To derive weak components for refactoring; To prioritize components for testing.	-	-	✓	✓				

Legend: R=Requirements; TC=Test Cases; I=Issues (Bugs, Change Requests, New Feature Requests...); S=Services (and Components) / P=Priority; C=Complexity; E=Effort; F=Frequency

Notes: * V&V - Validation and Verification ** Component / Service fulfills requirements.

*** Details on Requirement(s) (when visualized for any stakeholder) can include: name, ID, type (functional, non-functional, business, security, system, performance, safety...), priority, risk level, state of requirements (in work, in test, accepted, ready for release), business criticality, authors (sources), structure & content, description/specification/definition, rationale, fit criterion, dependencies, conflicts, attached files, history of changes, change requests and other issues, test cases attached, implementation deadlines; associated product or product line...