AREA	BAD PRACTICE	# OF OCCURRENCES	EXIST BUT HAVE NO IMPACT	EXISTS AND IMPACTS LIGHTLY	EXISTS AND IMPACTS STRONGLY
REPOSITORY	Project decomposition in the repository does not follow modularization principles	2	50,00%		50,00%
	Test cases are not organized in folders based on their purposes	4	25,00%	50,00%	25,00%
	Local and remote workspace are not aligned	1		100,00%	
	Number of branches do not fit the project needs/characteristics	3	~33,33%	~33,33%	~33,33%
	A stable release branch is missing	0			
	Feature branches are used instead of feature toggles	7	~28,57%	~42,85%	~14,28%
	Divergent Branches	7		~42,85%	~57,14%
	Generated artifacts are versioned, while they should not	2			100,00%
	Blobs are unnecessarily checked-in at every build instead of being cached	1	100,00%		
	Pipeline related resources are not versioned	0			
INFRASTRUCTURE CHOICES	Resources related to the same pipeline stage are distributed over several servers	1		100,00%	
	The CI server hardware is used for different purposes other than running the CI framework	4	25,00%	75,00%	
	External tools are used with their default configurations	4	50,00%	25,00%	25,00%
	Different releases of tools/plugins versions are installed on the same server	1	100,00%		
	Different plugins are used to perform the same task in the same build process	1	100,00%		
	A task is implemented using an unsuitable tool/plugin	3		~66,66%	~33,33%
	Use shell scripts for a task for which there is a suitable plugin available	0			
BUILD PROCESS ORGANIZATION	Inappropriate build environment clean-up strategy	1		100,00%	
	Missing Package Management	1			100,00%
	Wide and incohesive build jobs are used	1		100,00%	
	Monolithic builds are used in the pipeline	1			
	Independent build jobs are not executed in parallel	0			
	Only the last commit is built, aborting obsolete and queued builds	2	50,00%		50,00%
	Build steps are not properly ordered	1			100,00%
	Pipeline steps/stages are skipped arbitrarily	1		100,00%	
	Tasks are not properly distributed among different build stages	0			
	Incremental builds are used while never building the whole project from scratch	0			
	Poor build triggering strategy	1	100,00%		
	Private builds are not used	2	50,00%	50,00%	
	Some pipeline's tasks are started manually	5	80,00%	20,00%	
	Use of nightly builds	0			
	Inactive projects are being polled	0			
	A build is succeeded when a task is failed or an error is thrown	2	50,00%		50,00%
	A build fails because of some flakiness in the execution, whereas it should not	4	25,00%	75,00%	
	Dependency management is not used	1			100,00%
	Including unneeded dependencies	2	100,00%		
	Some tasks are executed without clearly reporting their results in the build output	3		100,00%	
	The output of different build tasks are mixed in the build output	1		100,00%	
	Failures notifications are only sent to teams/developers that explicitly subscribed	6	~33,33%	~33,33%	~33,33%
	Missing notification mechanism	2		50,00%	50,00%
	Build reports contain verbose, irrelevant information	3	~33,33%	~33,33%	~33,33%
	Time-out is not properly configured	0			
	Unneeded tasks are scheduled in the build process	1	100,00%		
	Build time for the commit stage overcomes the 10-minutes rule	1		100,00%	
	Unnecessary re-build steps are performed	2	100,00%		
	Authentication data is hardcoded (in clear) under VCS	1		100,00%	
ILD NABILITY	Absolute/machine-dependent paths are used	2		50,00%	50,00%
	Build scripts are highly dependent upon the IDE	2	50,00%		50,00%
	Environment variables are not used at all	3	~33,33%		~66,66%
	Build configurations are cloned in the different environments	3	100,00%		

AREA	BAD PRACTICE	# OF OCCURRENCES	EXIST BUT HAVE NO IMPACT	EXISTS AND IMPACTS LIGHTLY	EXISTS AND IMPACTS STRONGLY
BU MAINTAI	Build jobs are not parameterized	0			
	Lengthy build scripts	4	75,00%	25,00%	
	Missing smoke test, set of tests to verify the testability of the build	3		~66,66%	~33,33%
	Missing/Poor strict naming convention for build jobs	2	50,00%	50,00%	
QUALITY ASSURANCE	Lack of testing in a production-like environment	5		40,00%	60,00%
	Code coverage tools are run only while performing testing different from unit and integration	2		50,00%	50,00%
	Coverage thresholds are fixed on what reached in previous builds	0			
	Coverage thresholds are too high	2	50,00%	50,00%	
	Missing tests on feature branches	6		~33,33%	~66,66%
	All permutations of feature toggles are tested	1			100,00%
	Production resources are used for testing purposes	2			100,00%
	Testing is not fully automated leading to a non-reproducible build	1		50,00%	50,00%
	Test suite contains flaky tests	0			
	Bad choice on the subset of test cases to run on the CI server	0			
	Failed tests are re-executed in the same build	0			
	Quality gates are defined without developers considering only what dictated by the customer	4	25,00%	50,00%	25,00%
	Use quality gates in order to monitor the activity of specific developers	2		50,00%	50,00%
	Unnecessary static analysis checks are included in the build process	3		~66,66%	~33,33%
DELIVERY PROCESS	Artifacts locally generated are deployed	2	50,00%		50,00%
	Missing artifacts repository	1			100,00%
	Missing rollback strategy	3		100,00%	
	Release tag strategy is missing	2		50,00%	50,00%
	Missing check for deliverables	1		100,00%	
CULTURE	Changes are pulled before fixing a previous build failure	3		~66,66%	~33,33%
	Team meeting/discussion is performed just before pushing on the master branch	3		~66,66%	~33,33%
	Developers and operators are kept as separate roles	4		75,00%	25,00%
	Developers do not have a complete control of the environment	7	~42,85%	~42,85%	~14,28%
	Build failures are not fixed immediately giving priority to other changes	2	100,00%		
	Issue notifications are ignored	1			100,00%