Form - Bad practices of continuous integration

Identification

Email:	
System: ( )S1 ( )S2	Role: ( )Project manager ( )Developer
Experience time with system development (in years):	
Education level: ( )Graduate Degree ( )Master Degree ( )PhD	
Level of knowledge in continuous integration:	( )Basic ( )Intermediate ( )Advanced

1/7 - REPOSITORY

For each bad practice, check the corresponding alternative: 1 - I do not know how to inform, 2 - does not exist in the project, 3 - exists but does not impact, 4 - exists and impacts slightly, 5 - exists and impacts strongly

BAD PRACTICE	**1**	**2**	**3**	**4**	**5**
Project decomposition in the repository does not follow modularization principles					
Test cases are not organized in folders based on their purposes					
Local and remote workspace are not aligned					
Number of branches do not fit the project needs/characteristics					
A stable release branch is missing					
Feature branches are used instead of feature toggles					
Divergent Branches					
Generated artifacts are versioned, while they should not					
Blobs are unnecessarily checked-in at every build instead of being cached					
Pipeline related resources are not versioned					

1/7 - INFRASTRUCTURE CHOICES

For each bad practice, check the corresponding alternative: 1 - I do not know how to inform, 2 - does not exist in the project, 3 - exists but does not impact, 4 - exists and impacts slightly, 5 - exists and impacts strongly

BAD PRACTICE	**1**	**2**	**3**	**4**	**5**
Resources related to the same pipeline stage are distributed over several servers					
The CI server hardware is used for different purposes other than running the CI framework					
External tools are used with their default configurations					
Different releases of tools/plugins versions are installed on the same server					
Different plugins are used to perform the same task in the same build process					
A task is implemented using an unsuitable tool/plugin					
Use shell scripts for a task for which there is a suitable plugin available					

1/7 - BUILD PROCESS ORGANIZATION

For each bad practice, check the corresponding alternative: 1 - I do not know how to inform, 2 - does not exist in the project, 3 - exists but does not impact, 4 - exists and impacts slightly, 5 - exists and impacts strongly

BAD PRACTICE	**1**	**2**	**3**	**4**	**5**
Inappropriate build environment clean-up strategy					
Missing Package Management					
Wide and incohesive build jobs are used					
Monolithic builds are used in the pipeline					
Independent build jobs are not executed in parallel					
Only the last commit is built, aborting obsolete and queued builds					
Build steps are not properly ordered					
Pipeline steps/stages are skipped arbitrarily					
Tasks are not properly distributed among different build stages					
Incremental builds are used while never building the whole project from scratch					

Poor build triggering strategy					
Private builds are not used					
Some pipeline's tasks are started manually					
Use of nightly builds					
Inactive projects are being polled					
A build is succeeded when a task is failed or an error is thrown					
A build fails because of some flakiness in the execution, whereas it should not					
Dependency management is not used					
Including unneeded dependencies					
Some tasks are executed without clearly reporting their results in the build output					
The output of different build tasks are mixed in the build output					
Failures notifications are only sent to teams/developers that explicitly subscribed					
Missing notification mechanism					
Build reports contain verbose, irrelevant information					
Time-out is not properly configured					
Unneeded tasks are scheduled in the build process					
Build time for the commit stage overcomes the 10-minutes rule					
Unnecessary re-build steps are performed					
Authentication data is hardcoded (in clear) under VCS					

1/7 - BUILD MAINTAINABILITY

For each bad practice, check the corresponding alternative: 1 - I do not know how to inform, 2 - does not exist in the project, 3 - exists but does not impact, 4 - exists and impacts slightly, 5 - exists and impacts strongly

BAD PRACTICE	**1**	**2**	**3**	**4**	**5**
Absolute/machine-dependent paths are used					
Build scripts are highly dependent upon the IDE					
Environment variables are not used at all					
Build configurations are cloned in the different environments					
Build jobs are not parameterized					
Lengthy build scripts					
Missing smoke test, set of tests to verify the testability of the build					

Missing/Poor strict naming convention for build jobs					

1/7 - QUALITY ASSURANCE

For each bad practice, check the corresponding alternative: 1 - I do not know how to inform, 2 - does not exist in the project, 3 - exists but does not impact, 4 - exists and impacts slightly, 5 - exists and impacts strongly

BAD PRACTICE	**1**	**2**	**3**	**4**	**5**
Lack of testing in a production-like environment					
Code coverage tools are run only while performing testing different from unit and integration					
Coverage thresholds are fixed on what reached in previous builds					
Coverage thresholds are too high					
Missing tests on feature branches					
All permutations of feature toggles are tested					
Production resources are used for testing purposes					
Testing is not fully automated leading to a non-reproducible build					
Test suite contains flaky tests					
Bad choice on the subset of test cases to run on the CI server					
Failed tests are re-executed in the same build					
Quality gates are defined without developers considering only what dictated by the customer					
Use quality gates in order to monitor the activity of specific developers					
Unnecessary static analysis checks are included in the build process					

1/7 - DELIVERY PROCESS

For each bad practice, check the corresponding alternative: 1 - I do not know how to inform, 2 - does not exist in the project, 3 - exists but does not impact, 4 - exists and impacts slightly, 5 - exists and impacts strongly

BAD PRACTICE	**1**	**2**	**3**	**4**	**5**
Artifacts locally generated are deployed					
Missing artifacts repository					
Missing rollback strategy					
Release tag strategy is missing					

Missing check for deliverables					

1/7 - CULTURE

For each bad practice, check the corresponding alternative: 1 - I do not know how to inform, 2 - does not exist in the project, 3 - exists but does not impact, 4 - exists and impacts slightly, 5 - exists and impacts strongly

BAD PRACTICE	**1**	**2**	**3**	**4**	**5**
Changes are pulled before fixing a previous build failure					
Team meeting/discussion is performed just before pushing on the master branch					
Developers and operators are kept as separate roles					
Developers do not have a complete control of the environment					
Build failures are not fixed immediately giving priority to other changes					
Issue notifications are ignored					

NARRATIVE EVALUATION

Cite and describe your perception of the positive impacts of good CI practices in relation to software health. Exemplify for clarity of your answer.

Cite and describe your perception of the negative impacts of bad CI practices in relation to the health of the software. Exemplify for clarity of your answer.