Software Process Formative Assessment

Team	**I**

Marker	**Georgios Kampanos**
Matriculation	**2021259k**

Guidance

The purpose of the assessment is *formative*. The marks that you assign will *not* be used as part of the final assessment. The assessment provides the team with an opportunity to review their software process and identify opportunities for improvement. The assessment also gives *you* an opportunity to identify a process improvement activity with one of the PSD/TP teams.

- Each demonstrator in ASEP will be assigned one or two PSD/TP teams to provide formative assessment on the team's software process.
- The assessment comprises four categories worth 25 marks each. Each categories contains four questions worth 5 marks and a further 5 marks that can be used to give discretionary marks.
- Each assessment should take between 30 minutes and one hour.
- Conduct the assessment in person with the team where possible.
- Introduce yourself to the team and make sure that they have your contact details in case they have questions to follow up with you.
- Sit with the team and work through the assessment sheet collectively.
- Use evidence from the team's project artefacts (version control system, wiki, issue tracker, continuous integration environment and other.
- You are free to ask the teams to show you particular aspects of their project in order to answer certain questions. For example, you might want to review a recent selection of log messages.
- Do not worry about being over generous or cautious in marking: the mark should be indicative of your assessment of their performance.
- Try to make a short comment for each assessment category in the space provided.
- You can fill the assessment sheet in electronically (preferred) or on paper.

When you are finished, let the team make a copy of the mark sheet by taking a photograph, and upload it the ASEP Moodle submission slot. Let the team also take a copy.

Change Management

Did the team make small, frequent commits to the project SCM? Are all team members contributing to the SCM?	Available	Awarded
	5	4
Comment: **Team commits frequently on Git. While most team members make small and frequent commits some others commit only when a task it complete.**		

Did each commit have an informative log message that had a one sentence summary, explained the rationale for the change in more detail, with a reference to the relevant ticket in the issue management system?	Available	Awarded
	5	5
Comment: **Messages are descriptive. Team has a log format as well and all commits reference a ticket.**		

Is the project SCM appropriately organised according to project/ language conventions? Are there non-configuration items in the repository (e.g. binaries, compiled documentation)?	Available	Awarded
	5	5
Comment: **Yes, .gitignore removes all the unnecessary files such as binaries or executables.**		

Does the team have established practices for managing concurrent development and conflict resolution, such as feature branching?	Available	Awarded
	5	3.5
Comment:		

No, every member of the Team works in the master branch. They told me that because of the nature of the project they can separate the tasks and work on master without affecting the system. While this could be true, it can still be dangerous and introduce bugs.

Other/discretion	Available	Awarded
	5	4.5

Comment:

Overall their Git is really good. All team members contribute and everything looks organized and consistent. Maybe look more into branching!

Project Planning

Are the details of ticket (ownership, milestones, issue type, priority etc.) complete in the issue management system? Are there milestones for each iteration of the project with associated tickets?	Available	Awarded
	5	5

Comment:

All tickets are assigned to someone and have a milestone, priority and is descriptive.

Are tickets frequently updated during the course of the iteration as tasks are completed, or mostly towards the end of the iteration? Are new tickets created as needed throughout the iteration?	Available	Awarded
	5	5

Comment:

Yes, the trac and all the tickets are frequently updated.

Did the team make realistic estimates of the effort required for the iteration (i.e. were all objectives achieved?) If not, did the team	Available	Awarded

actively manage the project plan and identify low priority tasks to be removed from the iteration?	5	5
Comment: **Features promised to the client have always been achieved. Due to the nature of the project and because the client is the first time that tries such platform, some features are constantly "in progress". But the team seems to deliver what is asked from them every time.**		

Does the project have a wiki page containing key details such as a summary of the project, the locations of issue trackers and SCM systems, and team member contact details and responsibilities?	**Available**	**Awarded**
	5	5
Comment: **Yes, wiki page is very descriptive and has all the information needed for someone to get familiarized with the project very quick.**		

Other/discretion	**Available**	**Awarded**
	5	5
Comment: **Trac is very organized and the team is obvious that has put a lot of effort.**		

Quality Assurance

Was every commit tested in a continuous integration environment by executing an automated regression test suite?	Available	Awarded
	5	5
Comment: **Yes, every commit is automatically tested in Jenkins**		

Do the team immediately fix any broken builds report by the CI environment, or are new features added despite broken builds?	Available	Awarded
	5	5
Comment: **In the begging of the project the team went back and fixed a broken build. Now, the team has assigned one person to manage Jenking, testing and builds**		

Do the team perform code reviews? Has the team tried pair programming to actively review code?	Available	Awarded
	5	3
Comment: **Since the team is not using branches is (probably) impossible to do any code reviews before the code makes it to "production". Some pair programming has been done which is good.**		

Is the test suite effective (look for evidence from code coverage and/or mutation testing metrics)? Does the build/test cycle execute quickly (less than 10 minutes).	Available	Awarded
	5	5
Comment:		

Yes. Build is fast as well (26s)

Other/Discretion	Available	Awarded
	5	4.5

Comment:

Everything looks good here as well. The only "issue" here is the code reviews. They can be useful to identify bugs early, somewhere else in the system. Moreover, they can enforce good code practices as well as make the team members have one coding style. Again this could easily be achieved with the use of branches.

Process Improvement

Has the team elicited and documented *software process* issues during a retrospective? Did all team members contribute to the retrospective?	Available	Awarded
	5	5

Comment:

Everything is documented on trac.

Have software process problems been thoroughly analysed by the team, e.g. by using a 5-whys or other root cause analysis technique?	Available	Awarded
	5	5

Comment:

All problems and resolution can be found on their iterations on the trac wiki.

Have the team identified process improvement actions? Have these been documented and assigned to a team member for monitoring/ completion?	Available	Awarded
	5	5
Comment: **Yes, team has good communication and everything is documented on trac.**		

Have results from previous software process improvement actions been evaluated and acted upon?	Available	Awarded
	5	5
Comment: **Yes, the team takes customer feedback and acts upon. Again everything is on trac**		

Other/discretion	Available	Awarded
	5	5
Comment: **Very good job here as well.**		

Other Comments

I did not find any issues in the team or their methods. Overall they are very organized and seem to have great communication. They are in the right tack!