

# D8.1 Semester of Code guide for Universities

**Authors:** RayCom Bv, Edwin Veenendaal, Raymond Elferink

## Project Partners:

1. Universidad de Salamanca
2. OpenDirective Ltd.
3. Raycom B.V
4. University of Bolton
5. Università di Udine.
6. University of Cyprus
7. Mindshock S. L.
8. University of Oxford – OSS Watch

## Contents

1. Introduction.....	3
2. Experiences from the universities .....	4
3. Running the pilots.....	4
4. Process of Semester of Code.....	5
5. Roles in Semester of Code .....	7
6. The Implementation of Semester of Code.....	7
7. References.....	8

# I. Introduction

The original purpose of this report as stated in the project proposal was the following: “...describing the experiences carried out, specially focused in how business and foundations can obtain benefits from the exchange of knowledge between them and with HE institutions through the virtual placements. It includes a description of the virtual placements carried out from the point of view of the business and foundation, taking into account the problems addressed, the innovation open source projects selected, how the virtual placements solve those problems, how to find students interested on this kind of problems, how to find future candidates to work in the companies, how make possible to evolve open source software projects, etc. In addition a guide to solve the most common issues is included in order it can be taken into account for future Semesters of Code.”.

But as extensively described in the document “Understanding the barriers to virtual student placements in the Semester of Code”, we found lots of difficulties on the road for achieving a convincing number of student placements during the various pilot runs we did. So instead, after describing the experiences from the company point of view, we will lay out the process in general and how to implement Semester of Code, building upon the recommendations done in the above mentioned document.

## 2. Experiences from the universities

It seems an obvious benefit for universities to have the opportunity for students to enrich their theoretic career in the university with a practical assignment that will not only be judged on theoretical merits (the responsibility of the tutor), but also by the practical application of the created solution in the field (the responsibility of the mentor and the Open Source community that is behind it). So there is an extra pressure on students to deliver the best quality IT solution they can provide. Also – and this is maybe even more important- they come acquainted with the tools and way of working in the Open Source Software practice. This might be the environment of the student's future job after all. Finally, by implementing a Semester of Code participation, a university that has no IT business geographically nearby, still has the chance to offer their students a practical assignment by the very nature of virtuality in Semester of Code. If a practical assignment is wanted in the curriculum, this saves tutors time creating a relevant exercise.

Of course these benefits only hold when the university actually includes a practical assignment course in the curriculum or provides free space for students to fill in with a project of their choice. This could then very well be a Semester of Code project.

Unfortunately all participating universities had none of the two requirements in place. That meant that students participating, did that because they could find time outside their normal curriculum to do such a software project in Semester of Code. In some universities they could earn student points for that but it still kept its character of add-on to their regular curriculum. Because of that, all kind of barriers became more important and withheld students from participating. So that is what we experienced in the pilots in spite of the effort of the pioneers we had in the universities to promote participation and the change of approach in the Semester of Code process.

## 3. Running the pilots

It turned out that the different participating universities had a diverging timeline for the academic year. Students were not available in great numbers at the same time. So to enable the participation of a large group, we had to make a long period of registration and writing proposals. In the first instance some functionality was only made available during the appropriate phase of the matching process. Reminder emails and notification alerts were sent to the involved stakeholders per project to speed up the process of the matchmaking.

But with the larger periods of registration and proposal writing, this lost its momentum. For the second pilot, we introduced more universities, intensified the dissemination amongst students, student associations and so on, to get as many students on board as possible. This had some effect but not enough.

In the last pilot (starting in September 2015) we made the matching process flexible. Mentors could mark their project being available between a begin and an end date. We measured here no student participation at all. We think that the low participation was due to the minimised effort in the dissemination during this period. From September on we ran an extension of the project and the last pilot was merely to see what happens when we left it alone and also to ensure continuity of the Semester of Code concept. The fact that not one student wrote a proposal (from the 42 students who registered since september), perhaps tells us that the involvement of the universities and the tutor is essential. There were still 49 projects marked as being available after we asked the organisations to participate a third time. So that was promising.

## 4. Process of Semester of Code

The Semester of Code concept is about matching the need for performing non critical project ideas to the need to have practical work experience for students during the academic year. To enable this proces, students and their tutor register in an online system and enter a proposal for a project that was entered before by a mentor who offers to be a mentor for this project.

This matching process can either be guided by phases enforced by the system (like it was done by Melange in Google Summer of Code) or by a more loose matching approach around the projects which have an available status that depends on a preset time period –by the mentor. Both are possible; it is a matter of a setting. But of course one has to choose per Semester of Code instance which way to go.

Since we have experienced difficulties in getting enough students to participate in Semester of Code, we have to broaden the range of higher education institutes that we aim for. That could mean that we not only approach universities but also other computer science higher education institutes. So we also invite higher education institutes offering an engineering study in computer science to participate in Semester of Code. By having a large number of institutes there is more chance of competition between students and thereby perhaps a larger base of interesting companies and projects, but also the danger of incompatible curricula which makes it impossible to have a timeline that covers all. So a more flexible matching process is then desirable.

Basically we have then the following phases in the matching process for an institute.

- Decide if there is space in the curriculum to offer student points in exchange for work in the Semester of Code or perhaps have an enquiry amongst students to measure their enthusiasm to participate.
- Find a coordinator in the university that can act as contactsperson to the Semester of Code organising party and to the students in case of issues and questions.
- Find one or more tutors that are willing to spend some time on supervising a student's work in the Semester of Code.

- Contact the contact person of the Semester of Code organisation to get a registration code to be able to register your institute. This will be done by the coordinator (institute administrator).
- With that code, register the institute and create register codes to distribute to your supervisors.
- The supervisors create their own student groups, giving it a name to remember.
- They then let the students register themselves with the code that was automatically generated for their group. The supervisor can invite students automatically by filling in a list of email addresses.
- Then the supervisor can mark certain projects as favourite, meaning he/she is willing to be a supervisor for such a project. Students can filter the projects by the preference of their favourite tutor.
- Then the student writes a proposal in a couple of iterations until the student is satisfied and publishes the project proposal as open (visible for mentor also) or as published (open for final judgement of the mentor). Once the proposal is published or made more visible, the stakeholders receive an email notification.
- Once the mentor marks the project as accepted, he/she and the student and the tutor have to click an agreement for acceptance. In this agreement additional appointments can be written about contact frequency, time scheduling etc. But this is not obligatory.
- Then the actual work takes place. The tutor should monitor the process to some extend.
- After the work is finished to satisfaction, the mentor contacts the tutor and the student to mark the work as done. This should also be done in the VPS-system.

The work done is open source so it stays visible in the online version management system of the community. The tutor should check the work there and communicate with the mentor to get an idea of the work done and possible give credits in terms of student points when applicable.

## 5. Roles in Semester of Code

There are a number of roles fulfilled in the matching process.

There are students, supervisors and institute administrators for the academic site. In the system there is only one independent role: the administrator. This is the technical person within the university or organisation setting up a Semester of Code instance. He/she installs the VPS and is able to monitor the database and solve issues. Then there is the Semester of Code moderator –also called soc- who has almost the same rights as the administrator, but has no direct access to the resources of the system.

On the organisation side of the roles, we have two more roles: the organisation administrator and the mentor. In a lot of cases, the roles coincide. So in practice the organisation admin is also the mentor and will be the one to contact.

To enable for tutors organisation of students, they can divide them into student groups or classes. There is no limit to the number of student groups and the number of members. It is just a way to organise students. Student groups have no influence at all on the matching process: all students that are registered are treated the same in the system. It is a way though to find the original supervisor of a student. Students are free to choose any of the supervisors of the participating department of their university.

## 6. The Implementation of Semester of Code

We suppose there is an organisation in place to run an instance of the Semester of Code.

If your institute is the one acting as such an organisation, you need to set up an instance. You can download the Drupal module and place it on an Apache server. Under the Drupal module you install the vals\_soc module which can be found at Github. Follow the installation instructions in the Whitepages of the VPS system.

You also have to have a number of institutes that want to be part of the project. If you are running an instance on your own, there is less chance that organisations are willing to put effort in your initiative.

So since large numbers are important to make it attractive for organisations to participate, we try to bundle the initiatives into one organisation that tries to organise this Semester of Code event once a year. This initiative is currently being setup by the University of Salamanca. It gets its own board and enthusiastic members to run it.

So suppose you contact that organisation, they will likely use the same vps as was run in the pilot and so the same registration procedure will be used. You will be handed

a registration key to register your organisation with some description etc. by email. The person doing the registration will be Institute admin automatically. Next, you can produce a registration key yourself for all the supervisors that want to participate. You can also create other institute admins to share the responsibility. Note that institute admins are able to fulfill a supervisor role too.

A more detailed description of the registration procedure can be found in the document “*HowToVPS\_SemesterofCode\_VPS2*”<sup>2</sup>. How the vps functions, can be found at the user manual incorporated in the online VPS itself.

The most important in implementing the Semester of Code in your university or higher education institute, is to start long before the actual start of the instance, with the mobilisation of tutors and students. We found that the effort and enthusiasm of supervisors to participate in the Semester of Code was essential. It is more likely that students will follow, if supervisors stress the added value of a project in Semester of Code and fully support it, even if it has no direct link to the curriculum or a very strong theoretical basis. The supervisors act as ambassadors of the concept and should promote the opportunity in their lectures about half a year in advance, so that students can anticipate on it in their planning of their studies.

Students can be reached by organising seminars to explain the concept and to motivate students. It is a good idea to offer plenty of time for questions and show examples of both Google Summer of Code and previous projects of Semester of Code. If applicable, invite students who did a project in previous instances of Semester of Code to share their experiences.

## 7. References

García-Peñalvo, F. J., Griffiths, D., Cruz-Benito, J., Veenendaal, E., Achilleos, A. P., Wilson, S., & Kapitsaki, G. (2016). Understanding the barriers to virtual student placements in the Semester of Code. *Education in the Knowledge Society*, 17(1). *In press*

Veenendaal, E., (2015). *HowToVPS\_SemesterofCode\_VPS2*. An extract can be found in the online manual:  
[http://vps1516.semesterofcode.com/sites/all/modules/vals\\_soc/help/index.php?id=10](http://vps1516.semesterofcode.com/sites/all/modules/vals_soc/help/index.php?id=10)