Exposing High-School Students to Open Source Development with CodeYard

Adriaan de Groot* Sebastian Kügler* Donna Metzlar* Jasper Stein*
Marko van Eekelen*

March 29, 2007

Abstract

This paper shows our experiences with Open Source software development practices — the trinity of openness, distributed development and community — applied in the Dutch High School Computer Science curriculum. CodeYard is a project that offers support in the form of infrastructure, documentation, teaching materials and personal tutoring where needed to Dutch High School students. The programming projects undertaken within CodeYard may (but need not) be part of the course curriculum.

We describe the technical setup of CodeYard and the projects which have been undertaken within CodeYard. The results of these projects, along with some social observations on the work done by students within CodeYard are presented as conclusions.

CodeYard is about getting High School students to *produce* Open Source software, not use it.

The CodeYard [1] project was started in 2005 as a "SourceForge[2] for students," an Open Source collaboration environment for Dutch High School children. CodeYard was conceived when we noticed that at the High School level[3] there is very little familiarity with Open Source, either as a movement, a source of software or as a development method. This lack of familiarity was evident both among the students and the teachers. In order to address this (in our minds) failing, the CodeYard project was started.

The project aims to to show that computer science and software development can be *fun*. Many Open Source developers name "fun" as their motivation[4] for working on Open Source projects, although this intrinsic motivation can be split into altruism and community commitment [LW, cit01]. CodeYard introduces modern distributed Open Source software development into the classroom; this fits well with the increased emphasis on group projects in Dutch secondary schools and provides students with software engineering skills they may use after graduation.

Secondary purposes of the project are to increase the popularity of Computer Science in schools (and hence the number of students arriving at universities), increase awareness of Open Source in Dutch schools and to attract Open Source developers for other projects.

CodeYard provides infrastructure for the development of Open Source projects by students. This includes source repositories, forums, database servers, IRC, email lists. In short, CodeYard provides all the infrastructure of a typical Open Source project in one place. This is provided to students and schools free of charge. The initial reaction to CodeYard from teachers and students is positive. Now in its second year, CodeYard hosts some 30 active projects and 120 students.

*Supported by Stichting NLnet and the University of Nijmegen as part of the CodeYard project.

[1] `http://www.codeyard.net/`

[2] `http://www.sourceforge.net/`

[3] In the Dutch educational system, this typically means HAVO grades 4 and 5 and VWO grades 4–6, for secondary school children from the ages of 15-18.

[4] Consider the biography of Linus Torvalds [TD01], titled "Just for Fun."

An important consideration in the way CodeYard works is the "spirit" of Open Source. As much as possible, CodeYard itself operates openly and freely, using existing Open Source solutions, and it encourages students to do the same while respecting copyright and other restrictions on intellectual freedom.

1 Parts of CodeYard

The CodeYard infrastructure is relatively lightweight. After a year of work by 75 CodeYard projects (of which some 30 remain active and 20 are stillborn) only 1GB of disk space has been used. 4222 revisions have been made to the projects. This level of usage means that cheap commodity hardware can be used, although if CodeYard meets its long-term growth targets[5] much more infrastructure will be needed.

The CodeYard software consists of four main parts: the web portal, project repositories, a web forum and a news feed. Additional infrastructure is provided by e-mail and IRC networks[6], but these are not part of the CodeYard infrastructure itself.

The informational web portal, which includes statistics about active projects intended to give readers a sense of community, is a home-brew collection of PHP scripts. The project repositories use Subversion for version control. News is presented to the CodeYard community through a Serendipity weblog [7] installation, and the webforum is provided by PunBB[8]. By using as much existing Free software as possible, we reduce the amount of effort required to create the framework for CodeYard. Glue code is created to synchronize the login databases of the different parts to create a coherent whole.

The CodeYard infrastructure runs Free software exclusively: FreeBSD provides the operating system; Apache the webserver software, PHP a powerful scripting language and Postgres serves the needs of a relational database. Those are pars of the low-level infrastructure on which the visible parts of the system run.

When projects need specific additional resources such as a MySQL database, the CodeYard team evaluates the effort needed to implement them (for MySQL, the effort is not great) and provides the additional server software. This is an advantage for students because they can easily use whatever Open Source technology they need, but *also* for the teacher because he or she can instruct students to use that technology without having to deal with a local installation. We find that the IT departments in many schools are resistant to installing random Open Source software on their servers, so CodeYard provides that instead.

2 Open Source and Computer Science in CodeYard

Starting in 2007 students can take Computer Science as an examination-level elective course. This means that there are more hours available for Computer Science work and that project work is encouraged. With more time available for programming projects in the curriculum, there is some demand for innovative and interesting ways of using that time. Most of the school books used provide *programming* exercise, but very few deal with software development and software development processes as a project or as a management exercise.

Open Source is a philosophy of software development. The definition on the Open Source Initiative[9] website states:

> Open Source doesn't just mean access to the source code. The license shall not restrict any party from selling or giving away the software as a component

[5] Ideally, every school in the Netherlands, some 6000 institutions, would participate in CodeYard.
[6] CodeYard uses IRC channel #codeyard on FreeNode (`http://freenode.net/`).
[7] `http://www.s9y.org/`
[8] `http://www.punbb.org/`
[9] Excerpted from `http://www.opensource.org/`; the actual definition contains more conditions defining what is Open Source, but the conditions listed here are those most relevant for development work done in CodeYard.

> of an aggregate software distribution containing programs from several different sources. The program must include source code, and must allow distribution in source code as well as compiled form. The license must allow modifications and derived works, and must allow them to be distributed under the same terms as the license of the original software.

Because of these conditions, Open Source development means that the source code is shared; code is a form of knowledge. Sharing knowledge is fundamental to schooling, and we believe that showing students how to cooperate to share and create knowledge in distributed teams is part of a good education. The Open Source philosophy fits with education because the knowledge must be shared and must be able to be passed on to others, modified and applied for one's own purposes. In addition, most Open Source software is developed by a group of people, which stimulates the communication and social aspects of knowledge production; this fits well with modern theories of education which have steered the Dutch educational system towards a large proportion of group project work.

By adding a technical element, that of public software development through a distributed software revision control system, we add educational possibilities. Students can review each others work — even work from other projects if they like — because the code is accessible and public. In addition, teachers and coaches can review the work and give feedback even at a very early stage because the software is constructed incrementally in public view.

Finally, by making students realize that they may share code and use each other's code only because they give explicit permission through the license of the source code, we increase awareness of copyright issues; hopefully this translates into increased awareness of the students about copyright in other endeavors as well.

The CodeYard project makes it easy for High School students in the Netherlands and elsewhere[10] to set up an Open Source project for themselves or a project group. By presenting this work as part of the course work in the Computer Science courses the students take CodeYard provides students with a creative and useful way to share their schoolwork. CodeYard makes the projects independent of school or geographical location — this is normal for Open Source projects, but innovative for High School projects — so that students from different schools can work together, thus achieving more than any student could do individually.

3 CodeYard's First Year

In the first year of operation, the CodeYard team created the infrastructure and software system. A web portal with information and login management was developed and a workflow defined for administering projects and presenting them on the CodeYard website[11].

Public relations are important in order to present CodeYard to the press and most importantly to High School students who might be interested in using CodeYard resources. This led to a number of popular press articles and conference appearances [*I*&05, SAN06]. After one year, 75 students and 42 projects were active in CodeYard projects; that number has continued to grow.

The projects that have been started in CodeYard are very diverse. They range from the simplistic (an informative website about hairdressing) to commericial-like (online shops) to the esoteric (a LISP dialect using a virtual machine or a single-sign on protocol implementation).

[10] The original software was developed in Dutch, but is now being made ready for multi-lingual deployment.

[11] CodeYard itself is at `http://www.codeyard.net/` and the student project websites are at `http://cvs.codeyard.net/`.

4 How CodeYard Works

CodeYard offers High School students and teachers a complete infrastructure for Open Source development — that is, distributed collaborative development. Project groups sign up on the website. A group is always associated with one or more teachers, typically the teacher at the school where most of the students come from. The teacher is not expected to provide any guidance to the project, although it is preferable if they do. There are two reasons for getting teachers involved at all: one is that the teachers must know what their students are up to if the students are to receive class credit for their work, two is that the teacher acts as a backup disciplinary system. In case a student misbehaves on the CodeYard website or disregards the rules of the community (for instance, rude behavior on the forum or misusing copyrighted materials) then our preferred resolution method is to go through the teacher rather than communicating through email or outright banning the student from CodeYard.

A project's life cycle seen from the point of view of a student in a Computer Science class which lasts until the end of the school year is roughly as follows:

1. A group of students is formed at school and finds a project topic (this may be dictated by the teacher, but need not be),
2. the group registers itself on the CodeYard website,
3. the CodeYard team checks back with the school as to the identity of the students and teacher, so that we have a relatively firm grip on who is participating,
4. project repositories and resources are allocated on the CodeYard servers and
5. project development begins.

The administrative steps take a day or two. The identity checks — a phone call to the school to verify that the students do actually go to school there and that the teacher exists — serve a secondary purpose which is relationship management. Regular communication with schools and teachers ensures continuing mindshare for CodeYard.

Every project has a Subversion repository[12]. Subversion is a software tool that makes it easy to share documents, and which keeps a history of the changes in them. Within this repository students do all of the development work for the project. This includes a website (if any) and supporting documents (for instance, some schools want to see timesheets for the project, which indicates a high level of process management for projects).

The website for each project may use the PHP resources available on the CodeYard server; the websites are automatically updated from the Subversion repository when students make a change. This allows the project groups to quickly develop a fancy website if that is within the scope of their project or to leave a default notice stating the name of the project only[13]. In practice the students need only one set of tools — a text editor for both program code and the website along with a Subversion client — to fully support their project and this simplifies the workflow both for them and for CodeYard's support team.

5 CodeYard in Practice

Initial expectations for CodeYard were for 30 or so students in the first year. Clearly these expectations have been exceeded, but the way in which they have been exceeded shows that a number of our assumptions about how High School students would participate in CodeYard projects were mistaken.

[12] All CodeYard projects are listed at `http://cvs.codeyard.net/` where, in spite of the name, all the repositories are Subversion and not CVS.

[13] Compare `http://cvs.codeyard.net/OPS/` with `http://cvs.codeyard.net/svntools/`.

- We assumed that most students would register as individuals and then go looking for a project to participate in, much like people do in the larger Open Source community. However, most of the students arrive in groups, pre-arranged at school. This shows that the organization *in school* is much more important than we had expected and that the teacher plays a greater role in promoting CodeYard than we assumed. As a consequence, we no longer aim for the individual student but concentrate on reaching teachers.

- Similarly, we assumed that the geographical spread of students would be larger. Instead, 19 schools, primarily from the middle of the Netherlands (and one in Belgium), yield all of the participants in CodeYard projects.

- Since we assumed that students would show up individually looking for a project to hack on — similar to the usual motivation for Open Source developers — we thought that projects would carry on for longer than the school term. Thus far, no projects have survived the end of term and no project has been revived in a subsequent term for expansion. This suggests that students are primarily learning to program (not a strange suggestion, since typically they participate in CodeYard during their first year of doing Computer Science at all) and not yet ready to move into doing maintenance on other code.

The kind of activity a project sees varies greatly as well. A project with intentions for a fancy website but with inexperienced PHP programmers may see hundreds of revisions (changes in one or more files) in a day as small changes are tried out and tested. Depending on the project management requirements imposed by the teacher at the school there may be a large amount of process documentation added to the project repository even when no real coding activity takes place. Some projects lay dormant for considerable time until the students have completed the waterfall model software engineering steps before actual coding — again this depends on the structure imposed by the school.

Students have been generally positive about the experience of doing software development with CodeYard. The freedom afforded them in choosing technologies, projects and work forms is refreshing while the support given to them by the infrastructure helps them get started quickly.

6 Revising at School

One of the big eye-openers for the students who join CodeYard is the use of a version control system. This is not covered in the standard textbooks on Computer Science used in Dutch High Schools, and most of the students take to it like ducks to water. Two aspects are most attractive to students:

- Automatic distribution of materials. Because each Subversion user keeps a local copy of the project materials and updates them regularly across the internet, there is no need to be in the same room or to share disks or USB sticks. This opens up the hours in which a student can work on a project greatly and saves them considerable overhead.

- Automatic backups. We use the term “save game” to describe how Subversion repositories work because the students tend to be familiar with video games and the way in which you can use save games to progress through difficult levels of such a game. Since each revision is a backup of the project state at some point in time, the students are relieved from having to worry about destroying the entire project or losing work when some other project participant inadvertently deletes material.

As a means for keeping history for students[14] and for distributed backup and for enabling students to work with their own materials from any location, Subversion — or any revision control system — seems ideal, and several students have indicated to us that they now use Subversion for more than only their CodeYard projects.

7 Building a Community

A large part of CodeYard is *community building*. By creating project teams from different locations and schools and from a diverse population — something that is in the stated goals of CodeYard but which has not been achieved, as described in section 5 — we can create a community of High School students that support each other in creating Open Source software. The CodeYard webforums form the central point of this endeavor, although the IRC channel for CodeYard is gaining in popularity[15].

By building a community where older, more experienced students can help newcomers and where projects can share experiences of software development we hope to make CodeYard a more valuable resource for students and teachers alike. Checking the state of CodeYard should help students learn "the ropes" of Open Source software development.

While the electronic forms of communication are efficient, they have only a limited bandwidth and do not necessarily foster a real sense of belonging (to the CodeYard community). Therefore we also organize "CodeYard community days" in which we invite all active students to the University of Nijmegen for a session of talks on a Computer Science related topic, lunch and discussion on the state of the CodeYard projects. This kind of meeting has a much higher bandwidth than electronic communications and allows for better development of that sense of community which is needed in order for students to help each other and help themselves. Studies (such as Crowston, *The role of face-to-face meetings in technology-supported self-organizing distributed teams*) show that the value of face-to-face meetings like the community days, developer meetings ("sprints") or conferences have a great positive impact on the productivity of Open Source projects.

The student participants in CodeYard projects are very positive about the effect of the CodeYard community days — which also attract students who are *not yet* part of CodeYard but who are interested; most recently six newcomers showed up and all left attached to some project — which encourages us to organize more of such events aimed specifically at community building.

8 CodeYard as an Atypical Open Source Community

CodeYard is an atypical Open Source community because it is considerably more structured — with the CodeYard team overseeing activities and teachers as additional observers and enforcers of social order — than most Open Source communities. In his thesis Ruben van Wendel de Joode [vWdJ05] writes about the organizing factors in Open Source communities. In business, certain organizing factors are assumed to be necessary for success; these are usually lacking in (even successful) Open Source projects. We examine briefly the organizing factors here and contrast CodeYard with typical projects.

The most obvious difference between CodeYard and most Open Source projects is the presence of a central administration in CodeYard which is lacking in most (but not all) Open Source projects. In addition, CodeYard has barriers to entry that do not exist for real Open Source projects: participants in CodeYard projects must (in principle, which means that there are a half-dozen projects where this does not apply) be High School students.

[14] Unfortunately the use of proprietary binary formats for data is fairly common, which remove some the benefits of keeping change logs — individual changesets are harder to review for binary file formats

[15] A downside to IRC is the tendency towards off-topic rambling, which can quickly obscure the purpose of a channel.

While CodeYard as a whole is atypical in being centrally organized and having barriers to entry, each individual CodeYard project has its own decentralized organization. This may be more or less hierarchical; we do not have any information about how the groups actually work at school. Since the project groups thus far have been geographically compact (i.e. bound to a single school per project) we do not have any experience with how truly distributed CodeYard projects might work. With the little history we have available we cannot judge whether the projects have barriers to entry other than those of a meritocratic nature.

Another organizing factor mentioned in [vWdJ05] are the rules of production: a project needs a goal and should create a product. In Open Source projects the product is often a little nebulous, and with no central organization to define the product Open Source communities differ from businesses. CodeYard projects are somewhere in between because in practice the projects are organized as coursework for a class, which means that there must be results delivered and a clear product defined. Here the teacher's role within the CodeYard universe pushes the projects back towards a centrally organized paradigm.

A community of a certain size needs conflict resolution mechanisms. Within CodeYard we have not seen any conflicts *yet*, so our conflict resolution mechanisms remain untested. We prefer to leave that to the local groups and the teachers who are associated with each project. As long as the CodeYard projects remain geographically compact conflicts within a project group are best resolved that way. There have been no conflicts between groups about the direction of CodeYard.

Since CodeYard is a confederation of small projects, the community is large but the scale at which each project operates is in the 1–5 participant range where Open Source projects are born. This means that ownership of code and the project is very personal. None of the CodeYard projects have undergone the "phase change" to a larger distributed project with shared ownership. CodeYard projects so far have not attracted many new participants; typically a project starts, runs with one constant group of participants until it is done and then falls asleep. This removes the need — so far — to incorporate newcomers into a project.

A small project with more than three participants, however, does already suffer from the need to communicate and coordinate changes. Some teachers within CodeYard insist on four or more participants in a project to ensure that students learn about working together as a group; some other projects make do with a single participant. Because the participants in a CodeYard project are usually together at a single school, coordination *within* the group is relatively straightforward.

While projects are within the CodeYard confederacy, the CodeYard team does watch over their progress and makes suggestions as to best practices to the project teams. This includes using Subversion commit log messages, PHP coding style, documentation policies and usability and accessibility information. Here the role of the CodeYard team is as an *active* source of information to improve processes for each project, but it is not normative. Projects are free to take the advice or leave it. Here CodeYard is close to normal Open Source projects, although the level of oversight is most likely greater in CodeYard.

All in all we see that CodeYard hovers somewhere between regular Open Source projects and strict central control for its development practices; this is caused by the central control being limited in its power by the confederate nature of the participating projects.

9 The CodeYard Vision

CodeYard will take its place as a lively low-barrier-to-entry Open Source community for High School students where projects of varying size and ambition will be developed. These projects will often, but not always, be tied in with the Computer Science classes at school. Computer Science will be taught in a way that cannot come from a book in an inspiring and innovative fashion.

In a classroom situation CodeYard can help illustrate modern innovative software development. The motivation of students participating in CodeYard projects is high, and they

appreciate the additional challenges posed by Open Source development as well as the rewards they reap. CodeYard will certainly remain an atypical Open Source community due to the relatively strong outside influences — teachers, the school year and the CodeYard team – on it.

References

[cit01] *Working for Free? - Motivations of Participating in Open Source Projects*. IEEE Computer Society, 2001.

[*I*&05] *I*&I. *I*&I-*conferentie*. November 2005. `http://www.ieni.org/`.

[LW] Karim Lakhani and Robert Wolf. *Why Hackers Do What They Do: Understanding Motivation and Effort in Free/Open Source Software Projects*.

[SAN06] SANE Open Source Bazaar. May 2006. `http://www.sane.nl/`.

[TD01] Linus Torvalds and David Diamond. *Just for Fun: The Story of an Accidental Revolutionary*. HarperCollins, New York, 2001.

[vWdJ05] Ruben van Wendel de Joode. *Open Source Communities; an organizational perspective*. PhD thesis, Delft University of Technology, Delft, 2005. 270pp.