

ПРИЛОЖЕНИЕ НА СИСТЕМИТЕ ЗА УПРАВЛЕНИЕ НА КОД В ОБУЧЕНИЕТО ПО ИНФОРМАТИКА

Христо Лесев¹, Диляна Тоцева², Радослав Радев³

Пловдивски Университет „Паисий Хилендарски”

Факултет по Математика и Информатика

236 бул. "България", Пловдив, България

¹ hristo.lesev@gmail.com

² dilyana.totseva@gmail.com

³ radoslav_radev@gbg.bg

Резюме: В тази статия се обсъждат ползите от прилагането на системите за контрол на кода и управлението на проекти в обучението по информатика. Разглеждат се най-използваните в практиката системи.

Ключови думи: системи за контрол на кода, контрол и оценяване

1. Въведение

Системите за контрол на кода са мощен инструмент за професионална разработка на софтуер, използван във всички големи компании. Това е свързано с наложената практика, големи проекти да се разработват от множество екипи. В този случай е необходимо използването на система, която да улеснява екипната работа и да дава ефективни резултати. Разработката на софтуер би била немислима без тях. Те могат да се използват както от отделен потребител, така и от множество екипи, работещи по създаването на софтуер. Те оптимизират работата по проекта, подпомагат неговата разработка, проследяват направените в кода промени, ускоряват процеса на изпълнение, дават изключително добри резултати.

В раздел 2 се посочват някои проблеми, които стоят пред оценяването на практически задачи, поставяни в дисциплините по информатика. Раздел 3 прави кратък обзор на съществуващите системи за контрол на кода. В четвърти и пети раздел се описват евентуалните ползи от използването на такива системи и прилагането им във висшите учебни заведения. В раздел 6 са направени конкретни предложения за използването им в процеса на оценяване на студентите във Факултета по Математика и Информатика към Пловдивски университет „Паисий Хилендарски”.

2. Мотивация

Най-често практикуваният подход за проверяване и оценяване на студентите по време на семинарни занятия е разработването на проект. Често срещана практика е, възложените проекти да се проверяват и оценяват в последната учебна седмица на семестъра. По време на защитата на проекта,

студентът е длъжен да представи кратка презентация и да отговори на поставени въпроси върху програмния си код.

Съществуващият подход за проверяване и оценяване е твърде неефективен.

Основната му слабост е, че студентите предават своите проекти в последния момент. Така става физически невъзможно за един човек да ги провери и оцени. Често предадените проекти не отговарят напълно на всички изисквания, посочени в програмата на съответната дисциплина. Голям процент от студентите разчитат на заимствани проекти или намерени в Интернет. Оценяването в последния момент носи множество негативи, както за студента, така и за преподавателя и прави невъзможно да се прецени текущото състояние на даден проект. Преподавателят няма възможност да обърне внимание на съществуващите недостатъци във всеки проект и съответно студентът да бъде оценен най-точно. Почти невъзможно е да се разпознаят и евентуално заимстваните проекти.

Проведохме експеримент, при който студентите изпращат проектите си на електронната поща на преподавателя, няколко дни преди определената за защита дата. Целта е, той да има време да разгледа проекта, описвайки забележките си по него и да ги изпрати обратно на студента. Така се дава възможност на студентите да коригират проектите си и по този начин да повишат оценката си преди датата на същинския изпит. Също така е по-лесно да се забележат заимствани един от друг проекти, защото проверяващият е имал време да се запознае с тях по-подробно. Освен това той има проектите, съхранени при себе си и винаги може да направи по-подробен анализ за прилики в кода на тези от тях, в чието авторство се съмнява.

Това улеснение за студентите и преподавателя обаче се оказва на практика неефективно, защото те не спазват наложените изисквания и масово изпращат проектите си в последната вечер преди проверката, което отново довежда до физическа невъзможност на проверяващия да постави обективна оценка на изпълнението на поставената задача.

3. Системи за управление на кода

Едни от най-често използваните в практиката системи за контрол на кода, на които ще обърнем внимание в статията са CVS [2], Subversion (SVN) [1], Git [3], Mercurial [4]. Сред множеството възможности за подпомагане на разработката на софтуер, ще акцентираме на проследяване на промените в даден проект, разклоняването му на версии и евентуалната отмяна на промени в предишни версии.

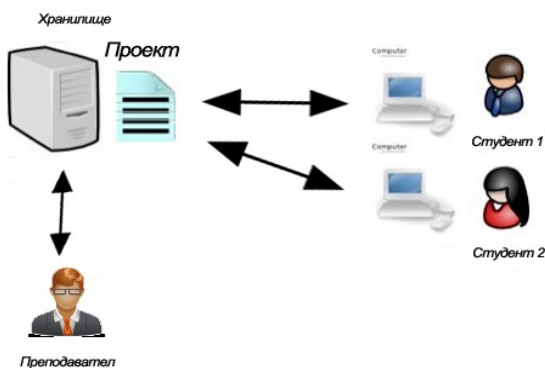
Тези системи се делят на централизирани и децентрализирани. Към централизираните се отнасят CVS [2] и Subversion (SVN) [1]. При тях кодът на проекта се съхранява в централно хранилище – сървър. Студентът може да изтегли свое работно копие, да направи промени по него и да върне

промененото копие в централното хранилище. Проектът се обновява и направените промени са видими за преподавателя. Основната разлика между двете системи е тази, че SVN има мощен механизъм за контрол на версиите, което улеснява значително разработката на по-сложни проекти, осъществяващи се от по-голям брой хора. Друго предимство на SVN е поддръжката на вътрешна информация дали файлът е текстов или двоичен.

Към децентрализираните системи се отнасят Git [3] и Mercurial [4]. При тях работното копие на даден проект е и хранилище. Почти всички операции върху проекта могат да се извършват локално, без връзка към основното хранилище (разположено на отделен сървър). При завършване на дадена задача, студента може да публикува своите промени в основното хранилище. Така те стават достъпни за преподавателя. Тук предимствата са очевидни при разработване на съвместен проект. Чрез този подход студентът има възможност да разработва своя модул, да приложи необходимите тестове и да отстрани всички възможни проблеми и конфликти още в своето локално хранилище. Когато приключи работата по даден етап от проекта, той може да го качи в главното хранилище и съответно то да бъде видимо за преподавателя и евентуално за колегите му, с които е работил в екип.

4. Ползи за обучението от системите за управление на кода

Улесненията, които подобен вид системи ще внесат в процеса на оценяването на студентите, са многобройни. На първо място е осигурена еднородна среда за съхраняването на самите проекти. Проверяващият може по всяко време да достъпи хранилищата през уеб интерфейс и да бъде уведомяван своевременно за настъпили промени в тях, например чрез автоматично изпращане на писмо от самата система. Същността на системите за контрол на кода е да запазват всяка настъпила в него промяна, което ще позволи на преподавателя да проследи развитието на студента и неговия проект по време на обучението и ще е добра база за набиране на статистически данни за уменията на студента и на обучаваната група като цяло. Могат да се проследяват степента на разбиране и грешките, които студентите допускат, и да се акцентира основно на тях в следващите семинарни занятия. От страна на проверяващия, системата дава възможност за сравняване на код от два различни проекта. Съществуващият потребителски интерфейс предлага множество визуални улеснения при преглеждането на функционалността на проекта. Ако към системата за контрол на кода се добави проследяващ модул, преподавателят може да съобщава на студента за забелязани нередности по кода. Проследяващите модули добавят тази информация като известия в системата, които се отнасят до дадена версия на кода. Така студентът може да проследи историята на дефекта и да го коригира своевременно. В зависимост от броя на отстранените дефекти, преподавателят може по-точно да прецени, каква оценка да постави.



Фиг. 1

Когато студента прецени, че е създад завършен код, той трябва да го качи в своето хранилище.

Всички системи за контрол на кода, разгледани в предишния раздел, могат да работят под управлението на най-разпространените операционни системи. Повечето инструменти за разработка разполагат с модули, които интегрират управлението на кода в самата среда. Това е допълнителен плюс за студентите, защото не ги ограничава изкуствено при евентуалния избор на операционна система и среда за разработка.

Основно предимство на предлагания подход е, че студентите ще се научат да работят със система за контрол на кода, която се използва от почти всички фирми в софтуерния бизнес и това ще е от голяма практическа полза при започването на работа. Този подход ще допринесе за засилване на връзката между процеса на обучение и сферата на бизнеса. Друг плюс е че студентът ще се запознае с всички етапи от процеса на реализиране на даден проект. В допълнение той ще добие представа за инфраструктурата и комуникациите при разпределяне на задачите между екип, работещ върху голям проект.

5. Прилагане на системи за контрол на кода в обучението

Практика в много чуждестранни университети е използването на система за контрол на кода. Като пример може да посочим University of Bath [7], както и University of Monash [8].

Приложеният от тях подход е следният: На студенти, приети в техническа специалност или посещаващи курс в областта на компютърните науки, е предоставена възможност да подадат заявка за създаване на собствено хранилище. Това се осъществява автоматично чрез попълване на форма в официалния сайт на университета. В тази заявка те посочват потребителските имена, които да имат достъп до хранилището, дата до която желаят то да е

активно, както и проекта, към който ще работят в посочения период, име на провеждания курс и име на лектор на този курс. Следва проверка в базата данни на университета дали подадените данни са валидни и регистриран ли е студентът към съответния курс. В последствие, чрез известите по електронната поща, той получава достъп със съответно потребителско име и парола.

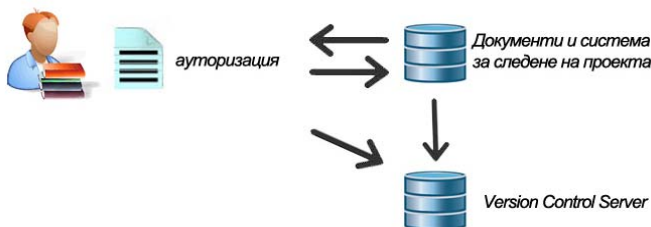
Този тип на работа в чуждестранните университети е свързана с практиката при задаване на практическа задача да се оценява както нейното изпълнение, така и работа на студентите в екип. С този подход всеки член на екипа има достъп до проекта по всяко време, възможно е да види какви промени са направени, каква част е изпълнена, както и да обменя с колегите си информация, свързана с оставащите задачи от заданието. Така преподавателят може да следи по всяко време развитието на проекта.

6. Нашият подход към внедряването на автоматизирани системи в процеса на проверка и оценка

Предложението, направено в статията е да се осигури сървър, на който да се инсталира система за контрол на код, в която всеки студент ще има лично хранилище. Към всяко хранилище достъп могат да имат двама души - студентът и преподавателят, който е и с права на администратор. Студентите не трябва да имат администраторски права към системата за контрол, за да не могат произволно да дават достъп на трети лица към техния код.

По време на упражненията през семестъра, студентът ще бъде задължен да обновява кода в хранилището си, след всяка реализирана точка (или група от точки) от изискванията към заданието. Предвид факта, че средната бройка точки за изпълнение по даден проект рядко е повече от десет, бихме могли да приемем, че пет обновявания на кода ще бъдат достатъчни да се проследи развитието на проекта във времето. Освен коректността на изпълнените точки, преподавателя ще може да проследи и темповете на изпълнение на отделните задачи. Целта е това да намали драстично практиката, работата по проектите да се отлага до последния възможен момент. Хранилища, в които е направено само едно обновяване и като резултат в тях се е появил изцяло готов проект ще дават ясен знак на преподавателя, че той е бил заимстван и студента умишлено е избегнал да отрази информацията за извършената работа през семестъра.

Всяка направена промяна в хранилището на проекта, трябва да бъде добре документирана.



Фиг. 2

За да се улесни процеса по преминаване към предложеня начин на работа, от полза ще бъде създаването на уеб приложение към сайта на Факултета по Математика и Информатика към Пловдивски Университет „Паисий Хилендарски“. Неговата цел ще е автоматизация на процеса на създаване на хранилище. Данните нужни за това са представени в следната форма:

Факултетен номер:	<input type="text" value="0801261030"/>
Име на хранилището	<input type="text" value="Разпределени приложения"/>
Описание на хранилището	<input type="text" value="В това хранилище ще разработвам заданието, дадено ми по дисциплината Разпределени приложения."/>
Потребителско име	<input type="text" value="student30"/>
Период на ползване (в месеци)	<input type="text" value="6"/>
E-mail	<input type="text" value="pavlov_k@uni-plovdiv.bg"/>
<input type="button" value="Изпрати"/>	

Фиг. 3

За идентификация в системата ще се изисква личният факултетен номер, той се сравнява с централизираната база данни на студентите във факултета.

Трябва да се посочи име на хранилището, което ще бъде свързано с името на изучаваната дисциплина.

В „Описание на хранилището“, ще се посочва допълнителна информация за проекта, който ще разработва, име на дисциплина, преподавател, име на проекта и т.н.

За улеснение е представена и опцията „Потребителско име“ – чрез него ще може да достъпва даденото хранилище, като алтернатива на факултетния номер.

Всяко хранилище ще има и период на ползване – тук студента ще трябва да посочи за какъв период от време желае да ползва хранилището, което ще му бъде създадено. След изтичане на този период, достъпът му до него ще бъде преустановен. Ако до месец след изтичане на посочения срок не бъде подадена заявка за удължаване срока на активността на хранилището, то ще бъде изтрито.

За комуникация по проекта, ще бъде задължително да се въведе валиден E-mail. На него ще бъде изпратено писмо, когато хранилището бъде създадено или преподавателя има бележки по текущата версия на проекта.

След автоматична проверка за валидност на данните, които студентът е въвел, ще бъде създадено негово собствено хранилище. Той ще получи електронно писмо на въведеният от него адрес, с което ще бъде уведомен, че хранилището е създадено успешно. В него ще бъде посочена паролата, която е генерирана автоматично към потребителското име, което е въвел. По време на първото използване на хранилището, системата ще изисква от студента да смени първоначално генерираната парола с нова. Целта е по-голяма сигурност и удобство. В писмото ще бъде посочена и датата, до която потребителското име ще бъде активно, спрямо въведеният от него период на ползване във формата.

Обръщаме внимание на времето на валидност на хранилището, тъй като неговата цел е улесняване на процеса на оценяване на конкретно задание и подтикването към по-активна комуникация между студенти и преподаватели в рамките на текущо провежданата дисциплина.

Поддържането на лични хранилища с неограничен срок на валидност, в които всеки студент ще може да запазва важни за него материали по време на цялото си обучение, повдига допълнителни въпроси, които са в страни от конкретната цел на статията.

Въпреки това се надяваме, че предложения подход ще доведе до улеснения, не само в работата по конкретна дисциплина, но и за в бъдеще, ще бъде предоставено място, където да се развиват и други проекти, свързани с учебната дейност.

Това въведение би било подходящо при работа върху по-големи съвместни проекти между преподаватели и студенти, чрез които те ще имат възможност да разработват в реално време общ проект. Пример за такъв вид дългосрочни проекти са дипломните работи.

Това ще е в помощ на студентите, защото ще ги насърчи да се научат да работят в екип върху отделни модули, каквато е и практиката във всички големи фирми.

Ако трябва да посочим подходяща, според нас, система за контрол на кода, това е Subversion (SVN) [1], която би могла да бъде приложена във Факултета по Математика и Информатика към Пловдивски Университет

„Паисий Хилендарски“. Първото и предимство е, че тя е програма с отворен код и съответно е безплатна. Наложила се е като водеща система за контрол на кода при разработката на софтуер от големи компании, напълно е тествана и с течение на времето са премахнати основните недостатъци и дефекти. Също така са налични множество разширения и уеб базирани приложения към нея за подобряване процеса на работа, като някои от тях са за управление на SVN сървъра, което улеснява значително неговата поддръжка. Не на последно място ще посочим, че са разработени SVN клиенти за най-използваните операционни системи.

С изложеното до тук и извлечения практически опит бихме могли да посочим тази система и като най-лесна за поддръжане. Затова тя най-лесно и бързо би могла да бъде приложена в учебната практика и използвана от студентите и преподавателите.

7. Заключение

В статията са разгледани текущите предизвикателства и проблемите, които възникват в процеса на обучение в семинарните занятия по информатичните дисциплини във Факултета по Математика и Информатика към Пловдивски университет „Паисий Хилендарски“. Обсъдени са начини за автоматизиране и улесняване на процеса по оценяване на извършената от студентите работа по време на занятията и извън аудиторната им заетост. Разгледани са съществуващи системи за контрол на програмен код. Обсъдени са начините за тяхното прилагане в процеса на обучение и ползите, произтичащи от това. В заключение смятаме, че висшето образование ще извлече големи ползи от системите за контрол на кода и управление на проекти, както за подобряването на процеса на оценяване, така и в приближаване на учебната среда с още една стъпка към изискванията на бизнеса.

Литература

- [1] Apache Subversion Documentation, <http://subversion.apache.org/docs/>
- [2] CVS Documentation, <http://savannah.nongnu.org/cookbook/?group=cvs>
- [3] GIT Reference Manual, <http://www.kernel.org/pub/software/scm/git/docs/>
- [4] Mercurial Guide, <http://mercurial.selenic.com/guide/>
- [5] The Trac Project, <http://trac.edgewall.org/wiki/TracGuide>
- [6] Fogbugz Bug Tracking, <http://www.fogcreek.com/fogbugz/>
- [7] University of Bath, <http://www.bath.ac.uk/>
- [8] University of Monash, <http://www.monash.edu.au/>