



استفاده از متدولوژی XP در آموزش مهندسی نرم افزار

علی غفاری برانقار

دانشگاه آزاد اسلامی-واحد قزوین

al_ghafari@yahoo.com

چکیده - درک عمیق مفاهیم، با بکارگیری آنها امکان پذیر می گردد. در آموزش مهندسی نرم افزار، مجموعه ای از مفاهیم و اصول وجود دارد که صرف آموزش نظری آنها، موجب یادگیری و بکارگیری فعالانه آنها نخواهد بود. در این مقاله از متدولوژی XP به عنوان ابزار و چارچوبی برای آموزش مفاهیم و اصول مهندسی نرم افزار بهره گرفته ایم و نشان می دهیم که چگونه مشکلات دانشجویان در درک مفاهیم با قواعد و اصول XP برطرف می گردد. نتایج نشان می دهد که دانشجویان در طی انجام پروژه، مفاهیم مهندسی نرم افزار و مهارت‌های مورد نیاز در تولید نرم افزار را به سرعت و با عمق بیشتری، فراگرفته و احساس رضایت بالاتری از پیشرفت خود می نمایند.

کلمات کلیدی: آموزش مهندسی نرم افزار، متدولوژی XP

۱- مقدمه

متدولوژی XP در سال ۱۹۹۶ توسط آقای کنت بک معرفی شد [1]. XP یکی از متدولوژیهای سبک وزن است که در سالهای اخیر مورد توجه جامعه نرم افزاری و بخصوص جامعه دانشگاهی قرار گرفته است. از این رو، محققین بسیاری در این حوزه با زوایای مختلفی XP، کاربردها و کارایی آنرا مورد مطالعه قرار داده اند. بسیاری از محققین بر کارایی آن نسبت به روشهای سنتی صحنه می گذارند [1]. بعضی از محققین علت‌های روانشناسی برای کارایی XP را مورد مطالعه قرارداده اند [9,10]. در [3]

تجربیات بدست آمده از بکارگیری متدولوژی XP مورد بررسی قرار گرفته و اشاره می نماید که پشت قواعد XP نکات پنهانی وجود دارد که در صورت عدم توجه به آنها، کارایی گزارش شده برای آن حاصل نخواهد شد. در [5]

به اشتباهات رایج در بکارگیری XP اشاره می نماید و هفت مورد از این قبیل اشتباهات و راه های جلوگیری از آنها را بیان می نماید. یکی از کاربردهای جدی XP کاربرد آموزشی آن است که مورد بررسی قرار گرفته است [1,2,6,7,8]. این تحقیق از آن حیث اهمیت می یابد که یکی از چالشهای مهم در تولید نرم افزار،

دغدغه کیفیت نرم افزار است. عوامل مختلفی برای کیفیت نرم افزار مطرح میگردد [11]. اما یکی از عوامل حیاتی در صنعت نرم افزاری، عامل نیروی انسانی است. بسیاری بر این باورند که استفاده از نیروهای خبره بخصوص در طراحی، کیفیت نهایی نرم افزار را تعیین می نماید [1]. در [4] گزارش جالبی از وضعیت صنعت نرم افزاری که در آن افراد با رشته های غیر تخصصی وارد عرصه تولید نرم افزار شده اند. به عبارت دیگر یکی از کمبودهای اساسی در این عرصه به روشنی مشخص میگردد و این یکی از تبعات عدم وجود آموزشهای موثر و کارآمد در عرصه مهندسی نرم افزار می باشد [1].

در این مقاله از متدولوژی XP به عنوان یک راهکار در آموزش مهندسی نرم افزار بهره گرفته ایم و تجربیات بدست آمده و مشکلاتی که دانشجویان معمولاً در آموزش مهندسی نرم افزار با آن برخورد می نمایند را مطرح و راه حل آنها را با استفاده از XP بیان می نماییم.

۲- معرفی متدولوژی XP

XP، یکی از متدولوژیهای سبک وزن است که اخیراً مورد توجه صنعت نرم افزاری قرار گرفته است. در این میان با توجه به ویژگیهای خاص آن، مورد توجه



یکپارچه نمود. بنابراین، هر روز تعداد زیادی محصول ساخته شده خواهیم داشت.

مشتری در محل⁸

مشتریها، همیشه به منظور شفاف سازی و اعتبار سنجی نیازمندیها در حین فرآیند در محل تولید حضور دارند.

آزمون واحد ها

موارد آزمون جعبه سفید زیادی قبل از تولید کد نوشته می شود. این آزمونهای خودکار، به کد پایه افزوده می شود. قبل از آنکه یک برنامه نویس بتواند کد خود را درون کد پایه یکپارچه سازی نماید این کدها باید تمام موارد آزمون را پشت سر بگذارد. این عمل باعث اطمینان از اینکه کد جدید وظایف کد پایه را به مختل نمی کند و در ضمن خود وظیفه اش را به درستی انجام می دهد.

آزمون وظیفه⁹

بطور سنتی، تکنیکهای مدیریت پروژه بر اساس ارزیابی های تولید کنندگان از مقدار وظایفی است که تکمیل شده است. در حالیکه XP از موارد آزمون وظیفه برای محاسبه میزان پیشرفت پروژه استفاده می نماید. XP به این ارزیابی با اصطلاح "سرعت پروژه"¹⁰ اشاره می نماید. موارد آزمون وظیفه بر اساس سناریوهای مشتری است. زمانیکه هر مورد آزمون وظیفه، بطور موفقیت آمیزی پذیرفته می شود، به این معنی است که آن وظیفه بطور مناسبی پیاده سازی شده است. تکمیل پروژه بر اساس درصد موارد آزمون وظیفه ای پذیرفته شده، اندازه گیری می شود.

زوج برنامه نویسی¹¹

همواره دو برنامه نویس در کنار یکدیگر، پشت یک دستگاه کامپیوتر روی یک طراحی، الگوریتم، کد و یک تست با یکدیگر همکاری می نمایند.

استاندارد کدنویسی

جامعه دانشجویان نیز می باشد [1]. در اینجا فعالیتهای XP را به اختصار مرور خواهیم کرد:

استعاره¹

متدولوژی XP بر این باور است که هر نرم افزار کاربردی دارای یک استعاره است که بر اساس آن یکپارچه شده است. به عنوان مثال، سیستم حقوق و دستمزد همانند یک سیستم خط تولید است که ساعات کارکرد ورودی آن است و نهایتا فیشهای حقوقی خروجی آن خواهد بود.

مالکیت کد²

در پروژه بر اساس XP، هیچ برنامه نویسی، صرفا مالک کد نیست. به عبارت دیگر، هر کدام از اعضای تیم احساس مالکیت کد را دارد. بنابراین هر عضو تیم می تواند بدون اخذ مجوز از بقیه اعضای تیم، کد را تغییر دهد.

طراحی ساده³

متدولوژی XP، روی طراحی ساده فوق العاده تاکید می نماید. متدولوژی XP بر این باور است که برنامه نویسان نباید سعی در پیش بینی نیازمندیهای پروژه نمایند و بر اساس آن طراحی را انجام دهند و بر این معتقدند که "ساده ترین چیزی که قابل انجام است، آنرا انجام دهید".

بازسازی⁴

بازسازی به معنای فرآیند بهبود ساختار کد بدون اینکه وظیفه آنرا تغییر دهیم [1]. XP بطور مستمر به بازسازی کد توصیه می نماید.

نشرهای کوچک⁵

XP ایده مدل مارپیچی⁶ را بکار می برد و نشرهای کوچک ۳ الی ۴ هفته ای دارد. در انتهای هر نشری، مشتری محصول را بازبینی، ایرادات آنرا شناسایی و نیازمندیهای آتی را تنظیم می نماید.

یکپارچه سازی مستمر⁷

کد نویسی به وظایف کوچکی تقسیم می شود و احتمالاً انجام آنها نیز کمتر از یک روز طول می کشد. بعد از تکمیل هر وظیفه باید آنرا با کد پایه

باشد و نه اینکه کیفیت نرم افزار به حضور افرادی با توانایی های بالای طراحی وابسته باشد. در صورت وجود افرادی خاص با توانایی مطلوب و شاید هم در حد نابعه در پروژه، موفقیت حاصل، به افراد نسبت داده خواهد شد و سازمان تولید کننده نرم افزار در سطح بلوغ اولیه خواهد بود. طبیعی است که در صورت ازدست دادن این نیروها، موفقیت های قبلی به احتمال زیاد تکرار نخواهد شد. لذا دیدگاه اول، متوجه استفاده از نیروی انسانی موجود در صنعت است و این کاملاً "طبیعی است که بکارگیری افراد کارآمد، عاملی بسیار حیاتی در افزایش کیفیت نرم افزار خواهد بود. همانطور که در [4] گزارش شده است، در بسیاری از پروژه های نرم افزاری، افرادی که ایفای نقش می نمایند، از دانش آموختگان رشته های غیر رشته مهندسی کامپیوتر می باشند و این اثر نامطلوبی بر فرآیند تولید نرم افزار و کیفیت آن خواهد داشت. بنابراین دیدگاه دوم به حل اساسی مساله توجه دارد. یکی از بستره های مهم آموزش مهندسی نرم افزار، آموزش دانشجویان رشته کامپیوتر به خصوص گرایش مهندسی نرم افزار می باشد. به علاوه باید به این نکته نیز اذعان داشت که با توجه به رشد روز افزون نیاز تقاضای صنعت به تولیدات نرم افزاری، در آینده نیاز به نیروهای آموزش دیده به مراتب بیشتر از تقاضای فعلی خواهد بود.

۴- مدل های آموزش مهندسی نرم افزار

در [1] به مدل های مختلفی برای آموزش مهندسی نرم افزار اشاره شده است. هر کدام از آنها دارای نقاط ضعفی می باشد که برای مقایسه آنها می توانیم انواع دانشی که در آنها بدست می آید را ملاک مقایسه قرار دهیم. در [1] به سه مدل آموزش مهندسی نرم افزار اشاره می شود :

۱- مدل مبتنی بر تئوری

۲- مدل مبتنی بر تئوری + عملی

۳- مدل فرآیند محور

در مدل اول، صرف آموزش مفاهیم و اصول موجود در مهندسی نرم افزار ملاک است. یکی از انتقادهای وارد

به منظور درک راحت و بهتر کدها توسط اعضا تیم، از یکسری استانداردها پیروی می گردد. با توجه به زوج برنامه نویسی و مالکیت کد توسط افراد تیم، استاندارد سازی کد اهمیت بیشتری پیدا می کند.

۱۲ فضای کاری آزاد

روابط افراد تیم با یکدیگر و با مشتری در XP از اهمیت فوق العاده ای برخوردار است.

هفته های ۴۰ ساعتی

XP معتقد است که نباید برنامه نویسان با اضافه کاری خوشان را خسته نمایند. تجربه نشان داده است که اضافه کاری و خستگی حاصل از آن باعث افت کیفیت محصولات تولیدی می شود.

۱۳ بازی برنامه ریزی

تکنیکهای جمع آوری نیازمندیها در XP برگرفته از روشهای سنتی است. در ابتدا مشتری نیازمندیها را به زبان خود کاملاً "غیر رسمی و داستان گونه می نویسد. تیم تولید، زمان تخمینی را تعیین می نماید و مشتری نیز اولویت ها را مشخص می کند. حال بازی برنامه ریزی شروع می شود، مشتری کارتهایی را انتخاب می نماید که بیشترین کارها در کمترین زمان قابل تحویل باشد.

۳- کیفیت و آموزش مهندسی نرم افزار

با توجه به گزارشات ارائه شده در [1,4] جامعه نرم افزاری دو رویکرد کلی در افزایش کیفیت نرم افزار دارد :

(۱) استفاده از نیروی انسانی متخصص

(۲) آموزش موثر نیروی انسانی

بسیاری بر این عقیده اند که تولید نرم افزار با کیفیت عالی، مدیون حضور طراحان عالی در تیم پروژه است [1]. البته این مطلب درست است ولی با توجه به فضای رقابتی موجود در صنعت نرم افزار و از سوی دیگر، محدودیت تعداد این قبیل افراد باعث برانگیختن انتقاد نسبت به این دیدگاه می شود. در [11] آنچه از مدل بلوغ فرآیند CMM بدست می آید، توجه و تمرکز بر تولید فرآیندهای مهندسی نرم افزار است که در آنها دانش افراد متخصص در دل فرآیند نهفته شده

برای تولید نرم افزار پیشنهاد شده است و امروزه بسیاری از ابزارهای CASE از این مدل پشتیبانی می نمایند که یکی از تجربیات موفق در مهندسی نرم افزار محسوب می گردد [13]. در [12] مدل آبشاری مورد انتقاد قرار گرفته و یکی از پیش فرضهای خوشبینانه و غیر عملی آن، داشتن درک درست و کامل از نیازمندیهای نرم افزار، در ابتدای پروژه (فرآیند) است. به عبارت دیگر هیچ ریسکی در کل اجرای فرآیند وجود ندارد و تغییر در نیازمندیها وجود ندارد و یا نامحسوس است.

در مدل اول آموزش مهندسی نرم افزار که متمرکز بر تئوری است، دانشجویان اصلاً "تفکر خوشبینانه (آبشاری) و واقعی (ریسک محور) از فرآیند را درک نمی کنند.

در مدل دوم آموزش مهندسی نرم افزار، به فعالیتهای مهندسی توجه می شود و بر طبق یک زمانبندی از قبل تعیین شده که ترویج همان تفکر و نگرش آبشاری به فرآیند تولید است باید محصولات فازهای مختلف را تحویل دهند. بنابراین، در دیدگاه آنها مدل تولید نرم افزار همان مدل آبشاری است. این قبیل دانشجویان در انجام پروژه های پایانی خود همین تفکر را اجرا می نمایند.

در مدل دوم آموزش مهندسی نرم افزار، دانشجو چون بدون درک مفهوم ریسک و تغییر در نیازمندیها بر خروجی فعالیت متمرکز است، بصورت پیش فرض، مدل آبشاری را پیگیری و درک می نماید.

در مدل سوم آموزش مهندسی نرم افزار، با استفاده از متدولوژی XP که از مدل مارپیچی بوهم^{۱۴} بهره میگیرد، دانشجویان یکسری نیازمندیهای کلی و مبهم که توسط مشتری (در اینجا مدرس درس) بیان می شود، فرآیند را آغاز می نمایند و باید در عرض یک تا دو هفته یک نشر از نرم افزار را تهیه نمایند. بنابراین با پیروی از XP دانشجویان مجبور به تغییر طرز نگرش آبشاری به فرآیند تولید نرم افزار خواهند بود.

مشکل ۲: دام تحلیل

بر این مدل، دانش دانستن چپستی است و نه دانش دانستن چگونگی. در حالیکه دانشجویان با این مدل قادر به انجام پروژه های خود نمی باشند.

در مدل دوم، تمرکز بر انجام فعالیتهای عملی است. به عبارت دیگر سعی در برطرف نمودن نقطه ضعف مدل اول بر آمده است ولی تجربیات نشان می دهد در این مدل که دانش چگونگی انجام کار بدست آمده، صرفاً در محدوده خروجی فعالیت های مهندسی نرم افزار است. دانشجویان با استفاده از این مدل، ضرورتاً توانایی های لازم در پروژه های عملی را پیدا نمی کنند. چراکه بسیاری از مهارت های لازم در عمل اتفاق می افتد و نه صرفاً در اجرای یک فعالیت مهندسی بدون توجه به زمینه کاری واقعی مورد نیاز. در مدل سوم، برای برطرف نمودن ایرادات موجود در مدل های اول و دوم، به فرآیند تولید نرم افزار توجه می شود. به عبارت دیگر سعی می شود، محیط واقعی تولید نرم افزار که در آن بسیاری از مهارت های مخفی وجود دارد، شبیه سازی گردد تا دانشجویان ضمن آموزش و بهبود مهارتهای مورد نیاز خود، به دانش دیگری دست یابند که در دو مدل قبلی بوجود نمی آید. این دانش، دانش فرا ادراکی است. به این معنی که دانشجویان دلیل انجام فعالیت های خود را می یابند. این دانش از نوع دانشی است که افراد خبره در زمینه های تحلیل و طراحی در اختیار دارند.

۵- مشکلات دانشجویان در مدل های آموزش مهندسی نرم افزار و راه حل آنها

در این بخش می خواهیم، مشکلاتی که دانشجویان با مدل های مختلف آموزش مهندسی نرم افزار دارند و راه حل برطرف نمودن و یا کاهش آنها با استفاده از فعالیتهای موجود در XP را بررسی نماییم.

مشکل ۱: دانشجویان آبشاری فکر می کنند مگر اینکه فکر آنها را عوض نماییم

فرآیند، دیدگاه و چارچوب متدولوژی تولید نرم افزار است و مدل های فرآیند مختلفی تاکنون پیشنهاد شده است که هر کدام از آنها دارای نقاط قوت و ضعف خاص خود می باشد [11]. در [12] مدل مارپیچی

طراحی و پیاده سازی می نمایند. لذا به طور مستمر، با درک و پیاده سازی مساله های کوچک، مهارت کافی در این فعالیت ها را در خود بهبود می بخشند.

مشکل ۳: دام نمادها

مدلسازی، یکی از هسته های اصلی فعالیتهای مهندسی است. مدلسازی بر اساس یکسری نمادها انجام می پذیرد. یکی از مشکلاتی که به تجربه در میان دانشجویان در آموزش مهندسی نرم افزار دیده ایم، عدم درک نحوه و دلیل استفاده از نمادهای مدلسازی است. اکثر دانشجویان بجای تمرکز بر درک فعالیت مدلسازی، نگران نمادهای مدلسازی می گردند به گونه ای که این توجه، تمرکز آنها را از حل مساله و رسیدن به یک نتیجه مطلوب باز می دارد.

در متدولوژی XP، یکی از ارزشهای مهم، تمرکز بر ایجاد ارزش افزوده از دید مشتری است. از دید مشتری، نرم افزاری که کار کند، حاصل نهایی و موفق پروژه است.

بنابراین، دانشجویان بطور مستمر و بصورت زوج برنامه نویسی باید بعد از درک تحلیل و طراحی وارد پیاده سازی وظایف نرم افزار شوند و در فواصل زمانی مشخص نشرهای نرم افزار در اختیار مشتری قرار دهند. طبیعت تکرار XP باعث افزایش مهارت دانشجویان در دوره های زمانی کوتاهی خواهد شد. لذا دانشجویان احساس می کنند نتیجه مطلوب را از آموزش خود می یابند.

در XP، مدلسازی زمانی انجام می شود که کمک به درک مساله و ایجاد ارزش افزوده نماید و با توجه به اینکه در هر دور، بخشی از نیازمندیهای نرم افزار برای پیاده سازی انتخاب می شود، نیاز چندانی به مدلسازی با استفاده از نمادهای استاندارد وجود ندارد.

مشکل ۴: تغییر فرهنگی و کار تیمی

استفاده از متدولوژی، مستلزم تغییر فرهنگی است [5]. زوج برنامه نویسی، یکی از فعالیتهای مهم XP است که برای اجرای آن باید تغییر فرهنگی در افراد تیم بوجود بیاید. در حالیکه دانشجویان عادت کرده اند بصورت انفرادی پشت یک دستگاه کامپیوتر، برنامه

ریشه تحلیل^{۱۵} به مشکلی اطلاق می گردد که در آن اصرار شدید بر فهم کامل مساله (نیازمندیها) در یک دوره خاص (در ابتدای پروژه) داشته باشیم. بسیاری بر این عقیده هستند که فهم کامل از یک مساله امکان پذیر نیست. در دنیای تولید نرم افزار که تغییرات در ابعاد مختلف به عنوان یک اصل و واقعیت پذیرفته شده است، تاییدی بر درستی این عقیده است.

بسیاری از دانشجویان، زمان بسیاری از پروژه را صرف تحلیل مساله می نمایند و با توجه به اینکه تجربه و مهارت کافی در بکارگیری تکنیک ها و روشهای تحلیل ندارند، این کار زمانگیرتر نیز خواهد شد. این زمان زیاد به این منظور صرف می شود که درک درست و کاملی از مساله بیابند تا در مراحل بعدی پروژه دچار تغییرات کمتری شوند و شاید هم بقیه پروژه را با سرعت بیشتری طی نمایند.

اما در عمل با توجه به عوامل زیر این اتفاق نمی افتد:

- عدم تجربه کافی در بکارگیری تکنیک ها و روشها
- بخشی از تغییرات تحت کنترل تیم پروژه نیست
- درک کامل تیم باید با بازخورد مثبت مشتری/کاربر توأم باشد

با توجه به وابستگی بین درک تیم از مساله و تایید کاربر است که می توان به اعتبار تحلیل صحنه گذاشت، لذا تحلیل یک امر و تعامل بین تیم تحلیل و مشتری/کاربر است و نه یک فعالیت خاص تحلیلگر.

بنابراین، اهمیت فعالیت تحلیل، نباید باعث از بین رفتن زمان و هزینه گردد. این مطلب زمانی شدت می یابد که دانشجویان مهارت کافی در شناخت نیازمندیها و مدلسازی آنها ندارند. به عبارت دیگر به قدری غرق تحلیل می شوند که زمانی برای پیاده سازی باقی نمی ماند.

در مدل سوم آموزش مهندسی نرم افزار با استفاده از متدولوژی XP ضرورتی در درک کامل نیازمندیها در ابتدای پروژه وجود ندارد. از طرف دیگر، با توجه به اینکه دانشجویان تجربه کافی در تحلیل، طراحی و برنامه نویسی ندارند، بخش کوچکی از مساله را

نویسی نمایند. این رفتار تک برنامه نویسی، در مدل‌های اول و دوم آموزش مهندسی نرم افزار، به نوعی تایید می گردد. چراکه در این دو مدل، روی همکاری تیمی، در چارچوب اجرای فرآیند تولید نرم افزار تاکید می وجود ندارد. بسیاری از مهارت‌های مورد نیاز دانشجویان، در فضای همکاری تیمی در چارچوب اجرای یک متدولوژی شکل می گیرد. چراکه در متدولوژی‌های مختلف برای افراد تیم، نقش‌های مختلفی وجود دارد و باید دانشجویان سعی در ایفای آن نقش داشته باشند. همکاری تیمی، صرفاً به معنای همکاری چند نفر با یکدیگر و تقسیم کار نیست بلکه باید در جهت اجرای یک متدولوژی انجام پذیرد. استفاده از XP بعنوان چارچوب آموزش مهندسی نرم افزار در مدل سوم، دانشجویان را مجبور به همکاری تیمی می نماید. بنابراین، بصورت تدریجی و ضمنی، تغییر فرهنگی مورد نیاز در افراد تیم ایجاد می گردد.

مشکل ۵: عدم مهارت کافی در بکارگیری ابزارها

در صورت استفاده از متدولوژی‌هایی مانند RUP که سنگین وزن هستند [13] در آموزش مفاهیم مهندسی نرم افزار، باید دانشجویان قادر به پیشگویی و تخمین باشند که با شرایط بی تجربه گی دانشجویان سنخیتی ندارد. در حالیکه متدولوژی XP از جمله متدولوژی‌های تطبیقی هستند و حالت تجربه در آنها بصورت تکراری در خود تکرارهای فرآیند بوجود می آید و مسبوق به تجربیات قبلی نیست.

همانطور که در [3] گزارش شده است، زوج برنامه نویسی باعث می شود اعضای تیم، دانش و تجربه و ایده های خود را با یکدیگر تسهیم نمایند و این خود باعث افزایش سرعت انجام پروژه و افزایش کیفیت کد خروجی خواهد شد. از این منظر، افرادی که در تیم پروژه مهارت کافی در بکارگیری ابزارها مورد نیاز در پروژه را ندارند به سرعت آموزش های ضمن کار را خواهند دید و می توانند در انجام فعالیت های پروژه نقش مفیدی ایفا نمایند. تجربه ما در محیط کلاس پروژه نشان می دهد که این قبیل افراد با توجه به اینکه در کنار افراد دیگر تیم، به سرعت آموزش می

بینند، انگیزه قوی برای ادامه فعالیتها از خود نشان داده و در واقع از انجام فعالیت‌های پروژه لذت می‌برند.

مشکل ۶: باور اصول، مفاهیم و تکنیک ها

به تجربه دریافته ایم بسیاری از دانشجویان، ضرورت بکارگیری متدولوژی (فرآیند) را درک نمی کنند مگر در عمل آنرا باور نمایند. در بکارگیری متدولوژی XP باید علت اساسی اصول را درک نمود. نه اینکه صرفاً اصول را بصورت سطحی اجرا نمود. تجربه نشان داده است که باور اصول متدولوژی به مراتب نتایج بهتری در پروژه نرم افزاری بهره می آورد. XP دارای سه سطح بلوغ است [3] متدولوژی XP دارای قواعدی است و می توان نحوه نگرش و برخورد با این قواعد را به سه سطح تقسیم بندی نماییم:

سطح اول: هر کاری را همانطور که نوشته شده، انجام دهید

سطح دوم: بعد از آنکه آنرا انجام دادید، گونه های دیگری در قواعد را نیز تجربه نمایید.

سطح سوم: نهایتاً نگران نباشید که XP کار می کنید یا غیر XP.

ما به تجربه دریافته ایم که صرف آموزش اصول و مفاهیم باعث یادگیری و بکارگیری این مفاهیم و اصول در دانشجویان نخواهد شد [1].

از دیگر اصول که در صورت درک سطحی نتایج خوبی در پروژه بدست نمی آیند، عدم رعایت اصل زوج برنامه نویسی است. در کلاس پروژه مشاهده کردیم، افرادی که در تیم پروژه، تعویض نقش گرداننده^۶ (کد نویس) و راهبر را انجام نداده اند و به عبارت دیگر به دلیل و ضرورت این فعالیت و تجربه پی نبرده اند، در انجام پروژه دارای سرعت کمتری بوده اند.

افرادی که به آنها اجازه داده شده است که در پروژه، خارج از اتاق پروژه کارکنند، دارای بهره وری پایینتری بوده اند و کمتر پیش آمده که از افراد تیم پروژه درخواست کمک نمایند. بطور تمام وقت در کنار یکدیگر بودن در اتاق پروژه این مشکل را حل نموده است [5].



کیفی انجام پروژه های دانشجویان و نیز دانش کسب شده در جدول ۱ نقاط ضعف و قوت سه مدل آموزش مهندسی نرم افزار مقایسه شده است.

جدول ۱: مقایسه سه مدل آموزش مهندسی نرم افزار

مدل	مبتنی بر تئوری	مبتنی بر تئوری + مبتنی بر عملی	مدل فرآیند محور
نقاط ضعف	عدم توانایی در انجام پروژه های عملی	عدم توانایی در انجام پروژه های عملی	زمانگیر بودن آموزش
نقاط قوت	آموزش مفاهیم و اصول تحلیل و طراحی	دانش چگونگی انجام فعالیت های مهندسی (تحلیل و طراحی) به صورت مجزا	دانش فرا ادراکی

۷- نتیجه گیری

در آموزش مهندسی نرم افزار، مجموعه ای از مفاهیم و اصول وجود دارد که صرف آموزش نظری آنها، موجب یادگیری و بکارگیری فعالانه آنها نخواهد بود. در این مقاله سه مدل آموزش مهندسی نرم افزار را مطرح نمودیم که دو مدل اول دارای بازدهی کمی هستند. مدل سوم که فرآیند محور است با استفاده از متدولوژی XP به عنوان ابزار و چارچوبی برای آموزش مفاهیم و اصول مهندسی نرم افزار، مشکلات دانشجویان در درک مفاهیم، قواعد و اصول را برطرف مینماید. نتایج نشان می دهد که دانشجویان در طی انجام پروژه، مفاهیم مهندسی نرم افزار و مهارت های مورد نیاز در تولید نرم افزار را به سرعت و با عمق بیشتری، فراگرفته و احساس رضایت بالاتری از پیشرفت خود می نمایند.

بسیاری از مشکلات مانند زمان بر بودن برنامه نویسی و ابهامات با این تعاملات در بین افراد تیم به راحتی حل می شود [5].

مشکل ۷: غلبه بر اسطوره های نرم افزاری

یکی از اسطوره های مهندسی نرم افزار، جهت گیری خاص نسبت به فرآیند است. اصولا در مهندسی نرم افزار بطور متوسط هر پنج سال یکبار نقطه تمرکز در جامعه نرم افزاری تغییر می نماید [11].

در [5] یکی از اشتباهاتی که در اجرای فرآیند XP وجود دارد، امتناع از انجام زوج برنامه نویسی است. چراکه این تغییرات فرهنگی در هر جامعه ای به سختی شکل می گیرد و با مقاومت همراه است. از جمله مشکلاتی که در اجرای XP وجود دارد، فرهنگ سابق و غالب در جامعه نرم افزاری است که هر برنامه نویسی پشت دستگاه خود کار برنامه نویسی را با تمرکز انجام می داده است و در واقع از دید XP این یک اسطوره نرم افزاری^{۱۷} تلقی می گردد. حداقل در پروژه های کوچک و متوسط ما را دچار بحران نموده است. در واقع بسیاری از برنامه نویسان این سبک برنامه نویسی را تجربه نکرده اند و بسیاری نیز باور نمی کنند که این روش برنامه نویسی نتایج مفید و قابل ملاحظه ای دربر داشته باشد.

۶- نتایج تجربی

برای بررسی عملی موثر بودن استفاده از متدولوژی XP در کلاسهای درس مهندسی نرم افزار ۲ و همچنین در ارائه پروژه پایانی دانشجویان کارشناسی، فاکتورهای مختلفی برای ارزیابی و مقایسه می توان در نظر گرفت که در این مقاله بر اساس میزان توانایی

منابع

- [1] Williams, L., & Upchurch, R. "Extreme programming for software engineering education?" 31st ASEE/IEEE Frontiers in Education Conference, 2001.
- [2] Reichlmayr, T., "The agile approach in undergraduate software engineering course project", 33rd ASEE/IEEE Frontiers in Education Conference, 2003.
- [3] Jackson, A., Shiu, L., T. & Gray, A. & Driver, C. & Clarke, S., "Behind the Rules: XP Experiences", Computer Society, Proceeding of the Agile Development Conference, 2004.
- [4] Long, L., N., "The Critical Need for Software Engineering Education", The Journal of Defense Software Engineering, 2008.



- [5] Lawrence, R., "XP and Junior Developers: 7 Mistakes (and how to avoid them)", Computer Society, IEEE/AGILE, 2007.
- [6] Hazzan, O., & Dubinsky, Y., "Teaching Software Development Methodology: The Case of Extreme Programming", Computer Society, Proceeding of the 16th Conference on Software Engineering Education and Training, 2003.
- [7] Wellington, C., A., "Managing a Project Course Using Extreme Programming", 35th ASEE/IEEE Frontiers in Education Conference, 2005.
- [8] Wellington, C., A. & Briggs, T. & Girard, D., "Comparison of Student Experiences with Plan-Driven and Agile Methodologies", 35th ASEE/IEEE Frontiers in Education Conference, 2005.
- [9] Bryant, S., "XP: Taking the psychology of programming to the eXtreme", 16th workshop of the Psychology of Programming Interest Group. Carlow, Ireland, April 2004.
- [10] Chong, J. & Plummer, R. & Leifer, L. & Klemmer, S., R. & Eris, O. & Toye, G., "Pair Programming: When and Why it Works", 2003.
- [11] Pressman, R., S., SOFTWARE ENGINEERING A PRACTITIONER'S APPROACH, McGraw-hill, 2001.
- [12] Boehm, B., "A Spiral Model for Software Development and Enhancement", Computer, vol. 21, no. 5, May 1998, pp. 61-72.
- [13] Kruchten, P., "The Rational Unified Process: An Introduction ", Second Edition, Boston: Addison-Wesley, 2000.

Metaphor¹
 Collective Code Ownership²
 Simple Design³
 Refactoring⁴
 Small Releases⁵
 Spiral⁶
 Continuous Integration⁷
 On-site Customer⁸
 Functional Test⁹
 Project Velocity¹⁰
 Pair Programming¹¹
 Open Workspace¹²
 Planning Game¹³
 Boehm¹⁴
 Analysis Paralysis¹⁵
 Driver¹⁶
 Software Myth¹⁷