

تنها مهمترین ویژگی یک برنامه نویسی حرفه‌ای، مسئولیت پذیری شخصی می‌باشد. برنامه نویسی حرفه‌ای مسئولیت حرفه، تخمین‌های زمانی، تعهدات برنامه ریزی، اشتباهات و کیفیت کار خود را بر عهده می‌گیرد. یک برنامه نویسی حرفه‌ای مسئولیت خود را گردن دیگران نمی‌اندازد.

اگر شما یک حرفه ای هستید، بنابراین مسئول حرفه‌ای خود می‌باشید. شما مسئول مطالعه و فراگیری خود هستید. مسئول به‌روز بودن در صنعت و تکنولوژی خود هستید. بسیاری از برنامه نویسان فکر می‌کنند این وظیفه‌ی کارفرما است که جهت تعلیم آنها اقدام کند. با عرض پوزش، این کاملاً غلط است. آیا فکر می‌کنید پزشکان اینگونه رفتار می‌کنند؟ آیا فکر می‌کنید وکلا اینگونه رفتار می‌کنند؟ خیر آنها در زمان آزاد متعلق به خودشان، به تعلیم خود می‌پردازند. آنها زمان بسیاری از ساعت‌های خارج از کار خود را به مطالعه‌ی مجلات مرتبط می‌پردازند. آنها خود را به‌روز نگه می‌دارند. این کاری است که ما هم باید انجام دهیم. رابطه‌ی بین شما و کارفرما به خوبی در قرارداد مشخص شده است. به صورت خلاصه او ملزم به پرداخت حقوق شما و شما ملزم به انجام دادن کار خوب هستید.

حرفه ای‌ها مسئولیت کدی که نوشته اند را بر عهده می‌گیرند. آنها کدی را بدون اطمینان از کارکرد صحیح آن، منتشر نمی‌کنند. یک دقیقه به آن فکر کنید. چگونه خود را حرفه‌ای می‌دانید اگر راضی به انتشار کدی که از آن اطمینان ندارید می‌شوید؟ برنامه نویسان حرفه ای انتظار دارند تیم تضمین کیفیت هیچ مشکلی در کار آنها پیدا نکند چرا که آنها قبل از اطمینان از صحت کار، کد خود را منتشر نمی‌کنند. البته قطعاً تیم تضمین کیفیت ایرادهایی پیدا خواهد کرد، چرا که هیچ کس، بی نقص نیست. اما به عنوان یک حرفه‌ای، نگرش ما باید این باشد که هیچ چیز برای تیم تضمین کیفیت باقی نگذاریم. حرفه‌ای‌ها بازیکنان تیم هستند. آنها مسئولیت خروجی کل تیم را بر عهده می‌گیرند، و نه فقط کار خودشان. آنها به یکدیگر کمک می‌کنند، از یکدیگر فرا می‌گیرند، یاد می‌دهند. وقتی یک هم تیمی سقوط می‌کند، دیگری خودش را دخالت می‌دهد. حرفه‌ای‌ها لیست بزرگ باگ‌ها را تحمل نمی‌کنند. سیستمی با هزاران مشکل ثبت شده در لیست Issue-tracking آن، نتیجه‌ی یک تراژدی از بی‌توجهی می‌باشد. در واقع در اکثر پروژه‌ها سیستم issue-tracking، یک نشانه از بی‌توجهی می‌باشد. فقط برای سیستم‌های بسیار بزرگ باید لیست مشکلات آنقدری باشد تا برای مدیریت آنها نیاز به نرم افزار Issue-tracking شود.

حرفه ای‌ها کار سرهم بندی نمی‌کنند. آنها کدشان را تمیز، با ساختار خوب و با خوانایی بالا نگه می‌دارند. آنها از Best practice ها و استانداردهای توافق شده پیروی می‌کنند. آنها هرگز و هیچگاه تعجیل نمی‌کنند. تصور کنید شما در حال مشاهده‌ی یک عمل قلب باز بر روی خودتان از زاویه‌ی شخص دیگری هستید. این دکتر یک ضرب الاجل دارد (در معنای تحت اللفظی). او باید کار خود را قبل از اینکه دستگاه بای پس قلبی-ریوی به سلول‌های خونی خسارت زیادی برساند به پایان برساند. می‌خواهید او چگونه رفتار کند؟ آیا می‌خواهید او به مانند یک نوع توسعه دهنده نرم افزار با عجله یک کار سرهم بندی انجام دهد؟ آیا شما می‌خواهید او به شما بگوید "من می‌روم و بعد رفعمش می‌کنم"؟ یا اینکه می‌خواهید او با تکیه بر نظم و انضباط، زمانش را صرف کند و مطمئن باشد از اینکه روشش بهترین روش منطقی موجود می‌باشد. شما یک سرهم بندی می‌خواهید به یک کار حرفه‌ای؟

حرفه‌ای‌ها مسئولیت پذیر هستند. آنها مسئولیت حرفه‌ی خود را بر عهده می‌گیرند. آنها مسئولیت اطمینان از اینکه کدشان صحیح کار می‌کند را بر عهده می‌گیرند. مسئولیت کیفیت کار خود را بر عهده می‌گیرند. وقتی که ضرب الاجل، از دور نمایان می‌شود، دست از اصولشان پس نمی‌کشند. در واقع، هر چه قدر فشار بیشتر می‌شود آنها به اصولی که معتقدند صحیح است بیشتر چنگ می‌زنند. -- [Robert Cecil MartinAdd](#)

مطلبی خواندم تحت عنوان 7 راه برای آنکه در سال 2014 برنامه نویسی بهتری باشیم. مطلب بسیار خوبی است و حاوی نکات تاثیر گذاری است.

یکی از بندهای آن اختصاص دارد به "ویژگی‌های برنامه نویسی حرفه ای" که آن را در مطلب جاری نقل قول کردم.

به امید حرفه ای شدن .

نظرات خوانندگان

نویسنده: مهدی پایروند
تاریخ: ۱۳۹۲/۱۰/۱۴ ۲۳:۱۰

به عنوان یک برنامه نویسی که شیفته تکنیک‌ها و کتابخانه‌های تست شده و کارا است، در مواقعی البته کم نه، نمیتوان همکاری را متقاعد کرد که روششون بروز نیست و نیاید چرخ‌ها رو از اول اختراع کنید. و یا اینکه نوع استفاده از ORM غلط و حادثه خیز است. من در دامنه و حیطه کاری خودم مطمئناً کار تمیز و کم باگ تحویل سورس کنترلر میدم ولی مطمئناً این ناهماهنگی مانع از گسترش مناسب و مفید کدهای تمیز در برنامه میشه. بنظر من باید اضلاع این مثلث با هم رشد کنه. 1- برنامه نویسی 2- قوانین 3- فناوری مورد استفاده. ممنون از مطلب خوبتون

نویسنده: علی رسول زاده
تاریخ: ۱۳۹۲/۱۰/۱۵ ۷:۴۷

یک برنامه نویسی حرفه ای باید دستمزد حرفه ای نیز بگیرد

نویسنده: محبوبه محمدی
تاریخ: ۱۳۹۲/۱۰/۱۵ ۱۳:۰۰

البته حرفه ای بودن به این آسونی‌ها هم نیست، برنامه نویسی حرفه ای بودن فضای کار حرفه ای، همکاری حرفه ای، مدیر حرفه ای می‌خواد. این رو باید به مطالب بالا اضافه کرد که برنامه نویسی حرفه ای در یک فضای غیر از این کار نمی‌کنه. اگر در همچین فضایی قرار گرفت بهتره قید حرفه ای بودن رو بزنه یا ترک کن اون کار رو چرا که از قدیم گفته اند: نزدیک شتر نخواب تا خواب آشفته نبینی!

نویسنده: آرمان فرقانی
تاریخ: ۱۳۹۲/۱۰/۱۵ ۱۵:۲۹

ضمن احترام به نظر همه دوستان باید عرض کنم حرفه ای بودن مفهومی نسبی است و تعریف مطلق آن در چند پاراگراف و یا دادن دستورالعملی منطقی و کوتاه برای برنامه نویسی حرفه ای شدن یا بودن میسر نیست. اما از دیدگاه من قطعاً برنامه نویسی که حرفه ای‌تر است در هر محیطی می‌تواند متمایز شود و قطعاً بخشی از حرفه ای بودن یک برنامه نویسی به تأثیری که در محیط می‌گذارد بستگی دارد. بحث از شخص برنامه نویسی است و چندان حرفه ای بودن یک برنامه نویسی تابعی از محیط نیست. در مورد خود فرآیند تولید نرم افزار محیط حرفه ای‌تر به کیفیت بالاتر محصول می‌انجامد. البته بدیهی است در محیط مناسب امکان بروز خلاقیت و ظهور استعداد و مهارت و ... برای برنامه نویسی حرفه ای بیشتر است. اما برنامه نویسی حرفه ای‌تر در هر محیطی حرفه ای‌تر است و اگر بنا بود به بهانه عدم وجود محیط حرفه ای اهمیتی به حرفه ای بودن برنامه نویسی به عنوان عضوی از تیم تولید داده نشود هم اکنون همین سایت نیز وجود نمی‌داشت. در فرآیند تولید همه نقش دارند و برای تحقق محیط حرفه ای باید همه نقش خود را هر روز بهتر ایفا نمایند. بسیاری از مشخصه‌های حرفه ای بودن یک برنامه نویسی هیچ ارتباطی به محیط ندارد و صرفاً به وظیفه شخص برنامه نویسی باز می‌گردد.

نویسنده: محسن خان
تاریخ: ۱۳۹۲/۱۰/۱۵ ۱۵:۳۱

عنوان صحیح این مطلب احتمالاً این بوده «wish list یک صاحبکار رند»

نویسنده: شاهین کیاست
تاریخ: ۱۳۹۲/۱۰/۱۵ ۱۸:۲۹

مطلبی که در پست جاری توسط آقای Robert Cecil Martin بیان شده سعی کرده برنامه نویسان را به خودآموز بود و مسئول بودن ترغیب کنه. به طور مثال خودآموز بودن و به روز بودن خیلی ربطی به محیط کار ندارد، هر چند محیط کار هم می‌تواند خیلی

موثر باشد. کسی که از شرایط کاریش گلایه می‌کند اگر سعی بر بروز بودن و پیشرفت و "حرفه ای" شدن کند می‌تواند شغل بهتری هم پیدا کند.

این مطلب نفی کننده‌ی شرایط کاری نیست .. من کمتر از دو سال پیش مطلبی نوشتم تحت عنوان "حقوق انسانی نیروی نرم افزار" ؛ آن وبلاگ دیگر نگه داری نمی‌شود بنابراین مطلب را در همین بازخورد ذکر می‌کنم .

حقوق انسانی یک برنامه نویسی :

در وبلاگ رادمان مطلبی نوشته شده تحت عنوان "[فراخوان برای نیروهای آماده به کار](#)" ، در این مطلب خواسته شده که :
پس از بحث‌های مفصلی که در نظرات نوشته چالش برانگیز آقای آواژ در زمینه "[تغییر هنجارهای کاری جامعه](#)" مطرح شد (ا [ینجا](#))، آقای کمالیان از نیروهای آماده به کار خواسته اند انتظاراتشان را از محیط کار بگویند تا پاسخی همه جانبه پیدا کنند برای اینکه "مشکل کجاست که شرکتها دنبال نیروی کارند و جویندگان کار، بی کار؟" ([اینجا](#))
لطفا اگر فرصت دارید در این زمینه مشارکت کنید. با توجه به تخصص جناب کمالیان در زمینه منابع انسانی، نظرات شما می‌تواند بسیار به ایشان و بالطبع به شرکتهایی نظیر ما که از مشاوره ایشان در حوزه منابع انسانی استفاده می‌کنند کمک نماید.
در تجاربی که در شرکت‌های قبلی داشتم مواردی بوده که من و دیگر برنامه نویسی را از کار دلسرد و زده می‌کرده که به صورت تیتروار به آنها اشاره می‌کنم : **آینده‌ی برنامه نویسی:**

من (من نوعی) هیچ آینده ای برای خودم در شرکت متصور نیستم ، هیچ روندی را طی نمی‌کنم به عنوان برنامه نویسی وارد شرکت می‌شوم و به در همان سمت می‌مانم و نهایتا بعد از مدتی لقب مدیر پروژه را یدک خواهم کشید. در واقع رتبه و درجه ای نیست که من بخواهم برای آن تلاش کنم و به آن برسم تا در شغلم پیشرفت کنم و به مزایای بیشتری دست پیدا کنم. در واقع بین برنامه نویسی کارمند و برنامه نویسی علاقه مند هیچ تفاوتی در عمل وجود ندارد. ([تفاوت‌ها یک برنامه نویسی کارمند و یک برنامه نویسی علاقه مند \(حقوق:](#)

حقوق کار در شرکت به خصوص در شهرهایی به جز تهران خیلی کم هست . خیلی بیشتر از خیلی ! خیلی از افراد فقط برای اینکه در خانه نباشند و در محیط دیگری باشند کار می‌کنند . **محیط شرکت:**

محیط شرکت خیلی مهم هست ، هر چه قدر هم صمیمیت بین همکارها برقرار باشد هر کسی باید از نظر فضا استقلال خودش را داشته باشد من دوست ندارم میز همکارم چسبیده به به میز من باشد دوست ندارم 10 دقیقه ای یک بار مدیر پروژه یا مدیر شرکت سرش را در مانیتور من بکند و بخواد ببینه من مشغول چه کاری هستم ؟ (این مورد به شعور افراد هم بر می‌گردد) ، من به عنوان برنامه نویسی دوست ندارم ده دقیقه یک بار صدای تلفن شرکت را تحمل کنم .. **همکاران:**

شخصیت فنی و فردی همکارها در شرکت مهم هست ، شرکتی که افراد نه دنبال یادگیری هستند نه دانش خود را به اشتراک می‌گذارند شما را افسرده خواهد کرد ! **شخصیت شرکت:**

شرکتی کار تولید نرم افزارهای مبتنی بر وب و ویندوز می‌کند و یک سایت درست حسابی ندارد خیلی تو ذوق می‌زند. معمولا بهانه‌ی مدیران هم نداشتن وقت و ... هست . **قدردانی :**

من به عنوان یک برنامه نویسی نیاز به قدردانی و تشویق دارم ! اگر کار من باعث پیشرفت شرکت ، باعث سود آوری بیشتر می‌شود باید به گونه ای قدردانی شود که انگیزه‌ی حفظ این روند در من حفظ شود . (گفتن آفرین خسته نباشی کفایت نمی‌کند).

استراتژی فنی شرکت:

: شرکتی که شما را به استفاده از الگوهای روز تشویق می‌کند تا شرکتی که به بهانه‌های خنده دار (Productive بودن) شما را مجبور به انجام کار بزن و در رو می‌کند خیلی تفاوت دارد. شرکتی که از ASP.NET MVC استفاده می‌کند تا شرکتی که از مهاجرت ASP.NET Web Form به ASP.NET MVC می‌ترسد خیلی تفاوت دارد.

وضعیت فنی مدیر پروژه و مسئول فنی :

این که یک نفر با هر سمتی به بعضی از تکنولوژی‌ها اشراف نداشته باشه یا اشتباه فنی کنه مسئله ای طبیعی هست ، اما این که فردی مدام اصرار به اشتباه کند و اشتباه خود را نپذیرد مسئله ای هست که پس از مدتی حسابی ایجاد مشکل می‌کند. **علامه‌های دهر ... :**

همیشه افرادی پیدا می‌شوند که خود را بدون هیچ فروتنی علامه‌ی دهر می‌دانند ، این افراد همه جا روی روان هستند اما اگر قرار باشند مسئول فنی شرکت همچین ویژگی داشته باشد که اشکالات خودش را قبول نکند ، ایرادات فنی را نپذیرد روزگار کابوس واری با وی خواهید داشت... به طور مثال مدیر فنی شرکت 4 ماه زمان گذاشته و ابزار یا کنترلی توسعه داده اما با دلایل کاملا منطقی به وی اثبات می‌کنید که دوست عزیز استفاده از این ابزار این معایب را به همراه خواهد داشت اما از آنجا که ایشان 4 ماه وقت و انرژی از شرکت صرف کرده و اعتبار خودش به خطر می‌افتد شما را مجبور به استفاده می‌کند. گاهی دیده شده مدیر پروژه اشتباهات خودش را گردن برنامه نویسی می‌اندازد ! (دیدم که می‌گم!) **مسیر آموزشی :**

بهتر است شرکت‌ها برنامه ای برای آموزش نیروها داشته باشند. البته این مورد به خود شخص برنامه نویسی هم بر می‌گردد که

تمایلی برای پیشرفت داشته باشد. **آچار فرانسه :**

از پرسنل متخصص به عنوان آچار فرانسه استفاده نکنید ، از وی در تخصصش استفاده کنید . **ساعت کاری :** ساعت کاری صبح تا عصر برای کار برنامه نویسی بی معنی است. یکی بازدهی خوبی در ساعات آخر شب دارد (شاید خیلی از برنامه نویس‌ها را دیده باشید که به این موضوع اعتراف دارند و علتش هم سکوت آن است!). یکی ساعات دیگر. خیلی‌ها در کار تمام وقت وسط روز چرت می‌زنند. چون ساعت کارشان نیست. (به نقل از وحید نصیری) **امکانات سخت افزاری :** monitor خوب و میز و صندلی خوب .

به سبب **بازی وبلاگی** که افشار محبی مدت‌های پیش راه اندازی کرده مطالب شبیه این مطلب قبلا نوشته شده ، امیدوارم خواننده‌ی این مطالب فقط برنامه نویسان نباشند و مسئولین شرکت‌ها هم این مطالب را بخوانند. مواردی که ذکر کردم تجاربی که خودم کسب کردم و موارد و نارضایتی‌های دیگر برنامه نویسانی که از نزدیک برخورد داشتم هست : هر چند خود برنامه نویس‌ها باید تلاش کنند جویای حق خود باشند.

نویسنده: مهدی پایروند
تاریخ: ۱۸:۳۳ ۱۳۹۲/۱۰/۱۵

ولی بنظر من همین سایت که شما ازش بعنوان مثال یاد کردید حاصل جدا شدن یک توسعه دهنده از گروه بدلائل مختلف هست. ممکنه کسی این توانایی رو داشته باشه بعنوان یک آزادکار شرایط جدید رو مناسب کنه و کس دیگری خیر. شاید برای گروه مفیدتر باشه که شما هم هم‌رنگ با جماعت کد کثیف بزنیید تا شیرازه کار از هم نپاشه. منظور منو اشتباه برداشت نکنید، مثال: یک نفر در یک تیم برنامه نویسی کدهای NHibernate رو میگیره و فکر میکنه چون اون زمان نسخه آخرش 3.0.0 بوده بگه دیگه خودم بروزش میکنم و اتفاقا برای حل یک مسئله که شاید بیرون از لایه NHibernate هم میشده حل بشه دست به خلاقیت میزنه و نه تنها تیم رو بلکه نرم افزار رو دچار چالش میکنه اونوقت یک پروژه بزرگ که بقای یک شرکت به اون بسته است با این خشت اول دیوار به ثریا میبره. حالا یک حرفه ای میخواد بیاد این لایه رو دست بزنه باید قبل از اون باید ببینه چطور کسی رو که به این 100 خط کد تعصب داره رو یک میلیمتر جابجا کنه و یا اصلا این کار بنفع نرم افزار هست یا نه. بله بنظرم محیط هم سهم بسزایی در این هماهنگی ایفا میکنه.

نویسنده: مهدی پایروند
تاریخ: ۱۸:۴۷ ۱۳۹۲/۱۰/۱۵

در مورد **علامه دهر** تجربه ای دارم که باید بگم: یک برنامه نویسی بجای رجوع به منابع TFS و درک و حل مسئله با استفاده از سورس کنترلر مورد استفاده برنامه ای تحت وب برای مدیریت تسک‌ها رو بعد از 3 ماه با کنار هم گذاشتن کنترلرهای یک کامپوننت و استفاده از EF Designer و در نهایت ساخت یک کاردستی تموم میکنه که با این نظر که کار اضافه ای کردی و بهتر از تو قبلا یک گروه این برنامه‌ها رو نوشتن و یا چرا از خود TFS استفاده نکردی، بهانه میاره که همشونو من تست کردم هر کدومشون مشکلی دارند.