

چند وقت پیش با یک آگهی استخدام برنامه نویسی به صورت زیر برخورد کردم:

Senior Developer
.Net Technologies
Resume : -----

آگهی بالا خلاصه ترین آگهی استخدامیه که می تونید ببینید و البته بیشتر هم در رشته کامپیوتر و مخصوصا جامعه برنامه نویسان رواج دارد.

احتمالا اکثر شما هم در آگهی های استخدام برنامه نویسی، با واژه **Senior Developer** یا برنامه نویسی ارشد برخورد داشته اید. حال به راستی به چه کسی برنامه نویسی ارشد گفته می شود. یا بهتره بگم که چه زمانی ما می تونیم خودمون را یک برنامه نویسی ارشد بدونیم؟

از آنجا که تعریف مشخصی برای این واژه ها در فرهنگ خاصی وجود ندارد و عموما هر کسی یک تعریف خاص با توجه به سلیقه خودش در این زمینه ارائه می دهد نمی توان یک شرح واحد از این واژه ها نظیر Senior Developer یا حتی Junior Developer ارائه داد. در نتیجه قصد دارم با توجه به تجربیات شخصی، این موارد رو تشریح کنم.

در خیلی موارد منظور از برنامه نویسی ارشد، کسی است که:

در حداقل یک یا دو زبان برنامه نویسی تجربه داشته باشد؛

با تکنولوژی های مهم و کارآمد ارائه شده در آن زبان آشنایی کامل داشته باشد؛

توانایی استفاده از این تکنولوژی ها را در جای مناسب پروژه دارا باشد؛

با الگوهای برنامه نویسی (Design Pattern) آشنایی کامل داشته باشد؛

بتواند به سایر اعضای تیم در حل مشکلات و باگ های برنامه کمک کند؛

یک خطایاب قهار باشد (منظور این است با اکثر استثنائات و خطاهای متداول زبان و تکنولوژی که پروژه با آن پیاده سازی می شود آشنا باشد و بداند که چه زمان این خطا به وجود می آید و چگونه می توان این موارد را برطرف کرد)؛

باید با پیاده سازی سیستم های سرویس گرا (SOA) جدای از تکنولوژی پیاده سازی آن آشنایی کامل داشته باشد؛

باید با روش های تست و خطایابی و ابزارهای توسعه و پیاده سازی آن آشنا بوده و توانایی استفاده از آن ها را در پروژه های خود داشته باشد؛

یک برنامه نویسی ارشد باید در زمینه تخمین زمان تکمیل پروژه مهارت داشته باشد. باید بتواند مدت زمان لازم برای تکمیل یک Task را تخمین بزند و در همین مدت زمان، Task مربوطه را به صورت کامل پیاده سازی نماید.

در خیلی موارد باید بتواند هماهنگی لازم را بین اعضای تیم برنامه نویسی ایجاد نماید؛

و...

*ترتیب عبارات بالا دلیلی بر اولویت موارد ذکر شده نیست.

*منظور از آشنایی در عبارات بالا، یعنی تسلط و توانایی استفاده از آن با آگاهی تمام.

چه مدت زمان برای تبدیل شدن به یک برنامه نویسی ارشد نیاز است؟

برای به دست آوردن مهارت ها و توانایی های یک برنامه نویسی ارشد لازم است که حداقل 8 تا 10 سال تجربه برنامه نویسی در یک زبان را داشته باشید. البته این در مورد همگان صادق نیست. شاید کسی بتواند این راه را در کمتر از 8 سال طی نماید. این

بستگی به تلاش و استعداد فرد دارد. اما تجربه بیشتر یعنی شرکت در پروژه‌های بیشتر و آمادگی بیشتر در حل مشکلات و مسائل مختلف در طی روند تکمیل پروژه. برای مثال در زمینه تخمین اجرای یک Task داشتن تجربه بیشتر خیلی به شما کمک خواهد کرد.

در پایان چهار ردیف یا رده مختلف را که بین برنامه نویسان رواج دارد ذکر می‌کنم. (از پایین به بالا)

Junior : عموماً به کسی گفته می‌شود که 1 تا 3 سال تجربه برنامه نویسی دارد. معمولاً کدهای نوشته شده توسط این افراد باید بررسی شود چون احتمال اشتباه در آن زیاد است. اکثر کدهای نوشته شده توسط این افراد به صورت dirty code است. راهنمایی‌هایی که به این افراد داده می‌شود شامل راهنمایی در زمینه ساختاری و الگوریتمی نیز می‌باشد.

Mid-Level : برنامه نویسان در این رده بین 4 تا 6 سال تجربه دارند. می‌توانند به تنهایی نیز یک مشکل موجود در پروژه را حل نمایند. با مباحث مربوطه به طراحی کامپوننت آشنایی دارند و پروژه را بی نیاز از کامپوننت خواهند کرد. حتی در بعضی موارد می‌توانند به تنهایی یک پروژه در سطح کوچک با متوسط را توسعه دهند.

Senior Developer : در بالا به صورت کامل شرح داده شد.

Luminary : به صورت معمول به کسی گفته می‌شود که تمام توانایی‌های یک برنامه نویس ارشد را داراست و فقط تجربه برنامه نویسی آن قطعا از 10 سال بیشتر است.

از این واژه کمتر در رده‌های برنامه نویسی استفاده می‌شود و بیشتر به همان واژه Senior Developer بسنده می‌کنند.

نظرات خوانندگان

نویسنده: مهمان
تاریخ: ۱۳:۴۳ ۱۳۹۲/۰۵/۰۴

با سلام و سپاس

من کمتر از 3 سال تجربه برنامه نویسی (به صورت حرفه ای) دارم ولی خودم رو از سطح Junior و Mid-Level خیلی بالاتر می بینم (بدون اعتماد به نفس کاذب) از طرفی با توجه به تعاریفی که جنابعالی از Senior Developer داشتید خودم رو در این سطح هم نمی دانم... به نظر شما من کدوم اصطلاح رو برای خودم به کار ببرم؟ (البته لازمه بگم که من فاصله خیلی زیادی رو بین Mid-Level و Senior Developer می بینم)

با سپاس

نویسنده: رضا خان میرزائی
تاریخ: ۱۳:۴۹ ۱۳۹۲/۰۵/۰۴

ممنون از مطلبتون ، این عنوان ها واقعا برام سوال شده بود چون وقتی در linkedin رزومه های دیگران رو می بینم همه می نویسند Senior Developer !

نویسنده: مسعود م. پاکدل
تاریخ: ۱۴:۲۷ ۱۳۹۲/۰۵/۰۴

اصولا بهترین روش برای تشخیص اینکه شما در کدام سطح قرار دارید این است که خودتون به مسیری که طی کردید تا به جایگاه فعلی برسید یک نگاه دقیق داشته باشید. اصولا اکثر برنامه نویسانی که در شرکت های نرم افزاری مشغول به کار هستند در سطح Junior یا Mid Level هستند. در این پست تمرکز اصلی برای تشریح سطح برنامه نویس ارشد بود. متاسفانه در اکثر آگهی های استخدام شاهد این هستیم که برنامه نویس باید با انبوهی از تکنولوژی های آشنایی داشته باشد. مثلا حتما در آگهی ها دیده اید که برنامه نویس Asp.Net MVC مسلط به JQuery و HTML و CSS و JavaScript و EF و یعنی باید برنامه نویس هم در لایه های سطح پایین نظیر Business و سرویس برنامه نویسی کند و هم به عنوان یک [Front- End](#) کار.

به صورت معمول زمانی که برنامه نویسان خودشون را در جایگاه Senior Developer نمی بینند از واژه Software Developer استفاده می کنند.

نویسنده: مسعود م. پاکدل
تاریخ: ۱۴:۳۱ ۱۳۹۲/۰۵/۰۴

اکثر این موارد به خاطر داشتن اعتماد به نفس بیش از اندازه است. یا توجه به رزومه شخص، به راحتی می توان حدس زد که فرد آیا واقعا در جایگاه Senior Developer است یا نه؟ دونستن این مطلب که جایگاه شما در بین برنامه نویسان کجاست از این نظر حائز اهمیت است که تفاوت بین حقوق دریافتی Software Developer و Senior Developer تقریبا زیاد است و قابل چشم پوشی نیست.

نویسنده: محسن خان
تاریخ: ۱۸:۴۲ ۱۳۹۲/۰۵/۰۴

«... با توجه به رزومه شخص ...» البته اگر صادقانه تهیه شده باشد. همون linkedin ایی که مثال زدند یک خاصیت جالب داره. اشخاص می تونند تأیید کنند آیا شخصی توانایی رو که ذکر کرده داره یا نه. هر چه رای بیشتری آورد یعنی صادقانه تر بوده و به حقیقت نزدیک تر.

نویسنده: مسعود م. پاکدل
تاریخ: ۲۲:۵۲ ۱۳۹۲/۰۵/۰۴

زمانی که شخص رزومه خود را برای استخدام در یک شرکت ارسال می کند اگر هم صادقانه نباشد به طور قطع در مراحل بعدی استخدام نظیر مصاحبه حضوری مشخص خواهد شد که فرد دارای چه رتبه ای است

البته این مطلب را عنوان کنم که فقط حدود 20 تا 30 درصد مصاحبه های حضوری برای استخدام در شرکت ها دارای استاندارد لازم هستند و در خیلی مواقع فرد مصاحبه کننده صلاحیت لازم برای این امر را ندارد. خود من حدود یک سال پیش برای استخدام به عنوان برنامه نویس ارشد در یک شرکت حضور پیدا کردم که در نهایت با 60 سوال کتبی (معادل 6 صفحه پشت و رو) مواجه شدم. فقط سه سوال اول رو مطالعه کردم که مربوط به تعاریف Interface و Property و Method بود که به نوعی تداعی خاطره آزمون دروس برنامه نویسی در دانشگاه بوده است. در ضمن هنگام تهیه و مطالعه رزومه باید به عناوین **تسلط و آشنایی و آگاهی** دقت لازم را داشته باشیم. چون معنی و مفهوم هر سه کلمه کاملا با هم متفاوت است.

نویسنده: سید ایوب کوکبی
تاریخ: ۰:۲۹ ۱۳۹۲/۰۵/۰۶

توضیح خوبی بود، ولی از بین تمام این مطالب این قسمت از نظر شما به دلم نشست:
در ضمن هنگام تهیه و مطالعه رزومه باید به عناوین **تسلط و آشنایی و آگاهی** دقت لازم را داشته باشیم.

خیلی از افراد کفر آدم رو در می آرن، ردیف کرده کلی تخصص حتی بدون تفکیک و شاخه بندی. جالب اینه که در همه زمینه ها هم اصطلاح تسلط رو به کار برده. من خودم به شخصه تسلط بر یک مبحث رو خیلی اصطلاح سنگین و دشواری میبینم. چون واقعا برای شایسته این اصطلاح بودن باید در سطح آگاهی و آشنایی نباشیم، باید در سطحی باشیم که چندین و چند بار از این اطلاعات و آگاهی ها در عمل استفاده کرده باشیم. به نظر من کلاس های آموزشی و مطالعه کتاب و بسیاری از موارد دیگر تنها ما رو به سطح آشنایی می رسانند ولی بدون استفاده عملی و کاربردی از اطلاعات در همین سطح باقی میمونیم و به تسلط بر آن موضوع دست پیدا نمی کنیم.

با این تفاسیر متاسفانه کارفرمایان بین این واژه ها تفاوت چندانی قائل نمیشوند و بیشتر به اصطلاحات دهن پر کن توجه میکنند که البته در طولانی مدت هم با مشکل مواجه خواهند شد.
بگذارید با یک مثال ملموس این قضیه رو روشن کنم:

یادتون بیارید زمانی که تازه با کامپیوتر آشنا شدید، نصب نرم افزارهای کاربردی کوچک همیشه برای شما سورپرایزی داشته اند یعنی ممکنه برای نصب بعضی از نرم افزارها با مشکل مواجه شده باشید، شما در اینترنت و یا کتاب به دنبال راه حلی برای نصب آن نرم افزار بودید، تا اینجا در مرحله آگاهی قرار دارید، یعنی می دانید که نصب نرم افزار فرایندی بدون خطا نیست، شاید به جاهایی با مشکل مواجه بشید که به راحتی نتوانید حل کنید.

بعد از مدتی که این روند تکرار میشه آگاهی های شما تعمیم پیدا میکنه و به سطح آشنایی میرسه، دیگه نصب نرم افزارها برای شما استرس و مشکلات پیش بینی نشده ای نداره. شما اکنون در مرحله آشنایی هستید.

الان در چه مرحله ای هستید؟ یقینا در سطح تسلط هستید، امکان نداره نصب مشکلترین نرم افزارها الان برای شما در بدترین حالت بیشتر از 1 ساعت طول بکشه. دیگه حتی به حدی رسیدید که دکمه Next و پذیرفتن لایسنس و محل نصب نرم افزار و ... جز روتین کار شده و مکثی روی این موارد ندارید.

شما الان می تونید بگید که من به نصب نرم افزارها تسلط کامل دارم.

آیا در زمینه Design pattern ها هم به همین صورت هستید؟ آیا در زمینه آزمایش واحد هم به همین صورت هستید؟ آیا استفاده از فلان الگوی طراحی به اندازه استفاده از حلقه For و کاربردش در ذهنتون قرار گرفته؟ آیا به اندازه با الگوی های طراحی کار کردید که در موقعیت های خاص ناخودآگاه به یاد استفاده از فلان الگوی طراحی بیفتید؟

اگر پاسخ این سولات مثبته شما واقعا به الگوی های طراحی مسلط هستید؟

اگر نیاز به کمی مکث و آزمون و خطا دارید، شما به الگوی های طراحی آشنا هستید؟

اگر فقط می دونید که برای فلان مسئله الگوی طراحی کاربرد داره و بتونید فلان کد رو که از الگوی طراحی استفاده کرده تا حدودی تحلیل کنید، شما به الگوی های طراحی آگاهی دارید.

پس به قول آقای پاکدل بین این سه مفهوم لطفا تفاوت قائل بشید.

نویسنده: محسن خان
تاریخ: ۱۵:۲۳ ۱۳۹۲/۰۵/۰۶

یک کار بهتر بجای این سؤال و جوابها یا مشاهده رزومه می تونه مراجعه به پروژههای سورس باز یک شخص باشه و بررسی کیفیت کدهای نوشته شده او. اصلا یکی از اهداف کارهای سورس باز نمایش توانمندیهای فنی افراد است.

نویسنده: مسعود م. پاکدل
تاریخ: ۹:۱۰ ۱۳۹۲/۰۵/۰۷

این هم می تونه یک راه حل باشه البته با در نظر داشتن این مورد که؛ نباید انتظار داشته باشیم همه برنامه نویسانی که دارای مهارت و توانمندی هستند، در وب سایتها فعالیت داشته باشند یا پروژههای سورس باز ارائه دهند. هستند افرادی که هیچ گونه نامی و نشانی از آنها در دنیای وب نیست ولی دانش بسیار زیادی در زمینه برنامه نویسی و توسعه نرم افزار دارند.