

صناعة المطورين مشروع خوارزمي

هذا المشروع مدعوم بواسطة

مبادرة خطوة ومجموعة صناعة المبرمجين وآخرون

تم إنشائه بتاريخ

18 يناير 2020

الاصدرة 1.0

مقدمة

ما هي المشكلة؟

السبب الرئيسي لتطوير البرمجيات في كل مكان هو المجتمع البرمجي، على سبيل المثال، تعتمد أسواق العمل على المجتمعات، وتعتمد الاستثمارات على المجتمعات، وبالتالي يجب أن يكون المجتمع البرمجي على دراية بالقيمة الأساسية والفائدة من تطوير البرمجيات.

في السودان المشكلة تكمن في عدم إدراك المجتمع لأهمية البرمجة وهذا يسبب الكثير من المشاكل الأخرى مثلاً خفض رواتب مهندسي البرمجيات وما إلى ذلك، وهذا يسبب ندرة في عدد المبرمجين.

كيف نحل هذه المشكلة؟

من أجل حل المشكلة السابقة التي نحتاجها لبناء مجتمع جديد ، نحتاج إلى إدراك الأفراد ، لكن الأمر يتطلب الكثير من العمل والكثير من الخطط التي قد تنجح أو تفشل.

ما هو مشروع خوارزمي؟

مشروع خوارزمي هو مشروع مفتوح المصدر. ناقش العديد من المبادرات والمبرمجين هذه المشكلة وكان مخرجات ذلك النقاش هو مشروع خوارزمي، تم إختيار اسم خوارزمي نسبياً للعالم العربي العظيم [محمد بن موسى الخوارزمي](#).

حاول المطورون وضع خطة صلبة لبناء مجتمع برمجيات جيد في السودان ، أرادوا تطبيق بعض القواعد أو المنهجيات على الناس في صناعة البرمجيات ، وبعد الكثير من العمل ، توصلوا إلى هذا المشروع.

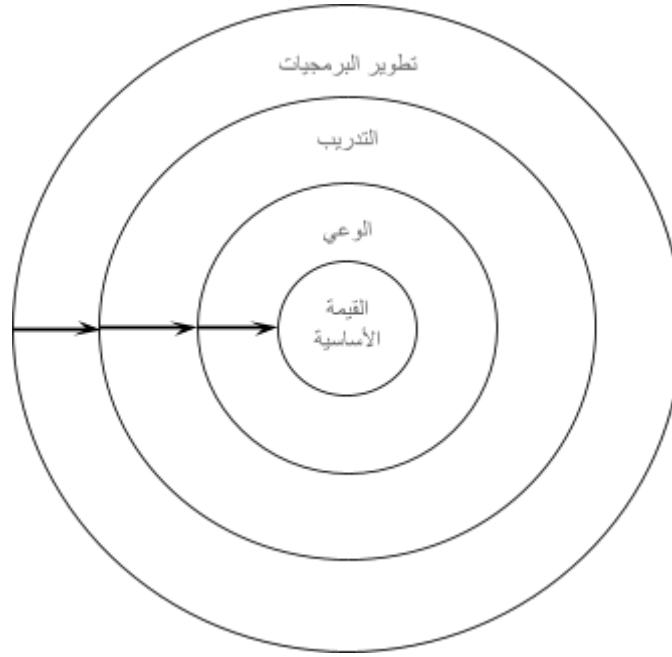
كيف عالج مشروع خوارزمي هذه المشكلة؟

من خلال إنشاء خريطة طريق للأشخاص لتعلم كيفية إنشاء البرامج بشكل مثالي ، نقسم التعلم إلى أربعة أجزاء رئيسية ، وهم:

1. القيمة الأساسية للمجال.
2. التوعية.
3. التدريب.
4. تطوير البرمجيات.

النقطة الرئيسية لهذا المشروع:

يتكون هذا المشروع من أربعة أجزاء رئيسية تغطي الكثير في مسارات تطوير البرمجيات ، من أجل الارتقاء بمجتمعنا إلى مستوى آخر. الأجزاء الأربعة كما ذكرنا من قبل ، هي القيمة الأساسية والوعي والتدريب وتطوير البرمجيات. هنا ندرجها في المخطط التالي:



ملاحظة: يشير السهم على الجانب الأيسر إلى أن كل طبقة خارجية تعتمد على الطبقة الداخلية ، على سبيل المثال: لكي تكون مطور برامج ، ستحتاج إلى تدريب ، وما إلى ذلك.

1. القيمة الأساسية

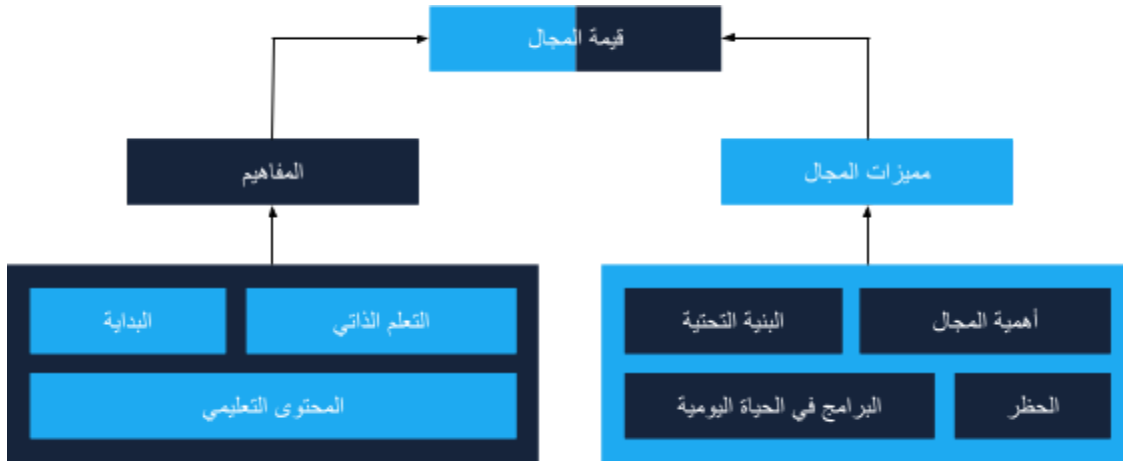
القيمة الأساسية لتطوير البرمجيات بشكل عام مهمة للغاية لأن تطوير البرمجيات يحدث اليوم ، كل شيء يمكن الآن تحويله أو ترجمته إلى برنامج ، لذلك يستحق استثمار الكثير من الوقت والمال في تصميم وتطوير المشاريع البرمجية.

2. الوعي

هناك الكثير من الأشخاص الذين بدأوا في تعلم تطوير البرمجيات ، ولكن الكثير منهم محبطون ، ومن المحتمل أن يستسلموا ، ولأنهم بدأوا بطريقة خاطئة أو ربما اختاروا مسارا خاطئا ، أو أن الأداة التي يستخدمونها ليست مناسبة أو غير مريحة معهم. ولكن إذا اتبعنا بعض الإستراتيجية لعدم ترك الاشخاص يستسلمون ، فإننا نطبق **خوارزمي** بالضبط. أولاً قبل أن نبدأ في تعليم الاشخاص الأدوات / اللغات ، يجب أن تكون النقاط التالية منطقية بالنسبة لهم.

1. القيمة الحقيقية للمجال.
2. مميزات المجال.
3. البرمجيات في حياتنا اليومية.
4. الحاجة للبرمجيات.
5. البداية.
6. محتوى التعلم.
7. التعلم الذاتي.
8. المفاهيم.
9. البنية التحتية.
10. الحظر في السودان.

حاولنا ترتيب النقاط العشر أعلاه حسب أولوياتها في التدفق التالي:



3. التدريب

التدريب هو الجزء الثالث من هذا المشروع ، وسيكون من المهم أن يدرك الناس القيمة الكاملة لكونك مطور برامج. لتعليم الناس بناء و / أو تطوير البرامج ، يجب علينا تطبيق بعض القواعد الفعالة على الأنظمة الأساسية (تطبيقات سطح المكتب وتطبيقات الويب وتطبيقات الجوال) والقواعد هي:

1. تصميم واجهات المستخدم.

واجهات المستخدم هي الجزء الذي يستخدمه العملاء ، واجهة المستخدم هي عملية التفاعل الفعلية التي تحدث بين العملاء والبرامج. كلما كانت واجهات المستخدم سهلة وجميلة، كلما تمت مشاركة البرنامج بشكل أسرع.

لذا في الجزء التدريبي ، يجب أن يحصل الأشخاص على الفائدة الكاملة لتصميم واجهات المستخدم الغنية والتفاعلية بشكل كامل.

2. المنطق داخل البرنامج.

الشيء الذي يعرف سلوك البرنامج هو المنطق الذي يحدث في البرامج وراء الكواليس ، مثل عمليات التحقق من الصحة عندما تريد التحقق مما إذا كانت البيانات الموجودة في حقل إدخال البيانات صالحة أم لا ، وإذا لم يحدث ذلك ، فعلى سبيل المثال ، يمكنك إظهار خطأ للمستخدم وسيعرف ما هي المشكلة. كل برنامج له سلوكه الخاص ويعتمد المنطق على البرنامج الذي تم إعداده له.

3. تصميم وبناء قواعد البيانات Back-End.

قاعدة البيانات هي جوهر أي برنامج ، وهناك الكثير من الطرق لتنفيذ قاعدة البيانات ، ويعتمد ذلك على حالة استخدام البرنامج نفسه. تصميم قاعدة البيانات مهم للغاية ، لذلك يجب أن يعرف الناس بالتأكيد المفهوم الرئيسي لقاعدة البيانات والطريقة الصحيحة لتصميم قواعد بيانات وواجهات برمجة جيدة التنظيم.

4. أنماط التصميم Design Patterns

في هذه المرحلة ، يجب أن يتعلم المطورون فوائد البرامج جيدة التصميم وكذلك عدة طرق لتصميم الكود. هناك الكثير من الأشياء التي تحدث هنا مثل تصميم أنظمة منفصلة للغاية وهو أمر جيد.

5. اختبار الكود.

الاختبار هو المرحلة الأكثر أهمية في دورة حياة تطوير البرمجيات. لا يقوم معظم المطورين في الواقع بتنفيذ الاختبارات في برامجهم ، وهذا ليس جيدًا على الإطلاق. في الجزء التدريبي من هذا

المشروع، سنعلم المطورين كيفية كتابة كود يختبر أجزاء البرنامج (testing) ، في البرنامج الذي يقومون بإنشائها.

6. إطلاق النسخة النهائية للبرنامج.

نريد أخيرًا إطلاق البرنامج للمستخدم بعد الانتهاء من تطويره. هنا سيعرف المطورون كل شيء عن تثبيت التطبيق ، ويستخدمون العديد من الأدوات المختلفة في هذه المرحلة.

7. صيانة البرنامج.

بعد عملية نشر البرنامج ، قد نضطر إلى قضاء بعض الوقت في صيانة البرنامج الذي أنشأناه وإصلاح بعض الخلل إذا كان هناك خلل ، لذلك علينا التأكد من أننا طورناه بشكل احترافي.

8. تطوير البرنامج.

يجب تقييم أي مشروع بما في ذلك المشاريع البرمجية ، لذا فإن إعطاء المشاريع البرمجية للمحترفين للحصول على تقييمهم أو حتى الأشخاص الذين يستخدمون برنامجك يستحق كل هذا العناء.

التدريب في شكل مشاريع.

في الجزء التدريبي ، يجب تعليم الأفراد في المشاريع ، وسنحسن هذه البنية من مهارات المطورين وتتيح لهم الاعتماد على بيئة تطوير البرمجيات كفريق أو فرد.

كيف نقوم باختيار أدوات التدريب ولغات البرمجة؟

يجب أن يعتمد اختيار أدوات التدريب على السوق والمجتمع لإجراء عملية الدعم عبر الإنترنت / دون اتصال بالإنترنت ، إذا كانت هناك مشكلة.

معايير اختيار الأدوات / لغات البرمجة.

1. سوق العمل.
2. الحظر.
3. مفتوحة المصدر / مغلقة المصدر.
4. مجانية / مدفوعة.
5. المجتمع الداعم للأدوات.

ملاحظة.

كيف تفكر كمطورًا حقيقيًا وأن تعرف استراتيجيات وطرق حل المشكلات. كيفية إنشاء فرق واختيار الأشخاص المناسبين ليكونوا جزءًا من فريقك ، والأهداف الأكثر شيوعًا التي يمكنك تحقيقها كفريق واحد.

المشاكل التي تواجه مرحلة التدريب:

1. مشكلة المواصلات.

الطول المقترحة لهذه المشكلة هي: إمتلاك سيارة ، سيارات الأجرة ، مشاركة تكلفة النقل ، التدريب عبر الإنترنت.

2. تأهيل المدرب قبل التدريب.

3. عملية إدارة الوقت.

4. خبرات المدرب.

تتضمن هذه النقطة ثلاث نقاط رئيسية وهي: المشاريع السابقة ، المحتويات والموجهون.

4. تطوير البرمجيات

في صناعة البرمجيات ، الشيء الأكثر أهمية الذي يحتاجه المطور هو الأخلاقيات. قد تكون قادرًا على إنشاء أكثر البرامج تعقيدًا وفائدة ، ولكن بدون أخلاقيات ، لن يتعامل معك أحد أبدًا ما لم تكن شخصًا جيدًا يتمتع بأخلاق عالية. قد تعتمد هذه المشكلة على الوعي والتدريب بدلاً من أن يتم تدريسها في مرحلة التطوير.

هناك بعض النقاط الرئيسية التي يجب أن نهتم بها

1. Soft-skills

2. Experience

3. Volunteer

4. Business