ماجيستير خاص
الجودة في تطوير البرمجيات (Software)

tech الجامعة التيكنولوجية

الجامعة التيكنولوجية

ماجيستير خاص
الجودة في تطوير البرمجيات (Software)

- طريقة التدريس: **عبر الإنترنت**
- مدة الدراسة: **12 شهر**
- المؤهل الجامعي من: **TECH الجامعة التكنولوجية**
- مواعيد الدراسة: **وفقًا لوتيرتك الخاصّة**
- الامتحانات: **عبر الإنترنت**

رابط الدخول إلى الموقع الإلكتروني: www.techtitute.com/ae/information-technology/professional-master-degree/master-software-quality

الفهرس

01 المقدمة

صفحة 4

02 الأهداف

صفحة 8

03 الكفاءات

صفحة 14

04 هيكل الإدارة وأعضاء هيئة تدريس الدورة التدريبية

صفحة 18

05 الهيكل والمحتوى

صفحة 22

06 منهجية الدراسة

صفحة 34

07 المؤهل العلمي

صفحة 44

01

المقدمة

أدى النمو السريع للصناعة ومتطلبات السوق الحالية إلى ارتفاع مستوى الدين التقني في مشاريع البرمجيات software.
نظرًا للحاجة الملحة إلى عكس الاستجابات السريعة لمتطلبات العميل أو الشركة، دون تقييم أو تحديد تفاصيل جودة النظام نفسه. هذا يعكس الحاجة إلى مراعاة قابلية التوسع في المشروع طوال دورة حياته، الأمر الذي يتطلب مهارات تكنولوجيا المعلومات التي تركز على الجودة من top-down. يقوم هذا البرنامج بتطوير المعايير والمهام والمنهجيات المتقدمة لفهم أهمية العمل على ضرورة تنفيذ سياسات الجودة في مصانع البرمجياتSoftware Factories . ستكون دراستك عبر الإنترنت بالكامل وستستمر لمدة 12 شهراً، وفقاً للمنهجية التي تتبعها أكبر جامعة رقمية في العالم.

تخصص في الجودة في تطوير البرمجيات (Software) من منظور تقني وإداري؛ تخرج في 12 شهرًا، واصنع فرقًا في بيئتك المهنية"

يعكس مفهوم الدين الفني الذي يطبقه حالياً عدد كبير من الشركات والإدارات مع مورديها مفهوم الدين الفني الذي يعكس الطريقة الارتجالية التي تم بها تطوير المشاريع. توليد تكلفة ضمنية جديدة تتمثل في الاضطرار إلى إعادة المشروع من خلال اعتماد حل سريع وسهل بدلاً من اعتماد ما ينبغي أن يكون نهجاً قابلاً للتطوير في تطور المشروع.

منذ بضع سنوات حتى الآن، تم تطوير المشاريع بسرعة كبيرة، بهدف إبرامها مع العميل على أساس السعر والمواعيد النهائية، بدلاً من اتباع نهج الجودة. تؤثر هذه القرارات الآن على العديد من الموردين والعملاء.

سيمكن الماجستيرالخاص هذه متخصصي تكنولوجيا المعلومات من تحليل المعايير التي تقوم عليها الجودة في تطوير البرمجيات (Software) على جميع المستويات. معايير مثل توحيد قواعد البيانات، والفصل بين مكونات نظام المعلومات، والبنى القابلة للتطوير، والمقاييس، والتوثيق، سواءً الوظيفي أو التقني. بالإضافة إلى المنهجيات في إدارة المشاريع وتطويرها وغيرها من الأساليب لضمان الجودة، مثل تقنيات العمل التعاوني، بما في ذلك البرمجة الزوجية Pair Programming، التي تسمح للمعرفة بأن تكون في الشركة وليس في الأشخاص.

تركز الغالبية العظمى من شهادات الماجستير من هذا النوع على تقنية واحدة أو لغة واحدة أو أداة واحدة. هذا البرنامج فريد من نوعه من حيث الطريقة التي يجعل الممارس يدرك أهمية الجودة في تطوير البرمجيات (Software)، ويقلل من الديون الفنية للمشاريع بنهج الجودة بدلاً من النهج الاقتصادي والوقت؛ فهو يزود الدارس بالمعرفة المتخصصة بحيث يمكن تبرير وضع ميزانية المشروع.

لجعل ذلك ممكناً، جمعت TECH الجامعة التكنولوجية مجموعة من الخبراء في هذا المجال الذين سينقلون أحدث المعارف والخبرات. من خلال حرم جامعي افتراضي حديث يحتوي على محتوى نظري وعملي موزع بصيغ مختلفة. سيكون هناك 10 وحدات مقسمة إلى موضوعات ومواضيع فرعية مختلفة تتيح إمكانية التعلم في 12 شهرًا باستخدام منهجية Relearning التي تسهل الحفظ والتعلم بطريقة مرنة وفعالة.

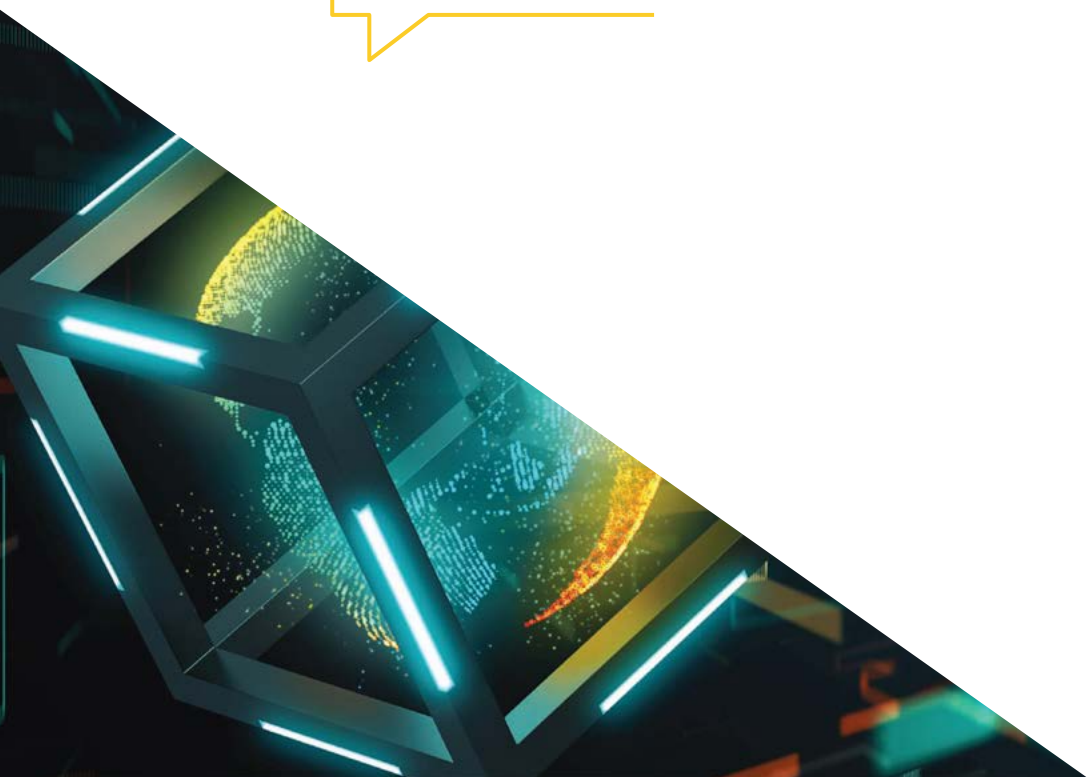
يحتوي **الماجستير الخاص في الجودة في تطوير البرمجيات (Software)** على البرنامج التعليمي الأكثر اكتمالاً وحداثة في السوق. أبرز خصائصه هي:

- تطوير دراسات الحالة المقدمة من خبراء في تطوير البرمجيات Software
- محتوياتها البيانية والتخطيطية والعملية البارزة التي يتم تصورها بها تجمع المعلومات العلمية والعملية حول تلك التخصصات الأساسية للممارسة المهنية
- التمارين العملية حيث يمكن إجراء عملية التقييم الذاتي لتحسين التعلم
- تركيزه الخاص على المنهجيات المبتكرة
- دروس نظرية وأسئلة للخبير وعمل التفكير الفردي
- توفر المحتوى من أي جهاز ثابت أو محمول متصل بالإنترنت

يحلل الماجستيرالخاص في الجودة في تطوير البرمجيات (Software) المعايير التي يقوم عليها الموضوع على جميع المستويات. وسّع من مستوى خبرتك. سجّل الآن"

برنامج يركز على زيادة الوعي بأهمية الجودة في تطوير البرمجيات (Software) والحاجة إلى تنفيذ سياسات الجودة في مصانع البرمجيات software Factories.

تعلّم بطريقة عملية ومرنة. شارك حياتك اليومية مع هذا التدريب 100% عبر الإنترنت حصرياً TECH الجامعة التكنولوجية.



"طوير المعايير والمهام والمنهجيات المتقدمة لفهم أهمية العمل الموجه نحو الجودة، وتقديم حلول فعالة لشركتك أو عميلك"

البرنامج يضم في أعضاء هيئة تدريسه محترفيهن يصبون في هذا التدريب خبرة عملهم، بالإضافة إلى متخصصين معترف بهم من الشركات الرائدة والجامعات المرموقة.

سيتيح محتوى البرنامج المتعدد الوسائط، والذي صيغ بأحدث التقنيات التعليمية، للمهني التعلم السياقي والموقعي، أي في بيئة محاكاة توفر تدريبا غامرا مبرمجا للتدريب في حالات حقيقية.

يركز تصميم هذا البرنامج على التعلّم القائم على المشكلات، والذي يجب على المهني من خلاله محاولة حل مختلف مواقف الممارسة المهنية التي تنشأ على مدار السنة الدراسة. للقيام بذلك، سيحصل على مساعدة من نظام فيديو تفاعلي مبتكر من قبل خبراء مشهورين.

02

الأهداف

يوفر الماجستيرالخاص في الجودة في تطوير البرمجيات (Software) للطلاب رؤية واضحة ومتخصصة لأهمية الجودة في عمليات تطوير البرمجياتsoftware. بالإضافة إلى الأدوات الأكثر تقدماً لتنفيذ عمليات DevOps وأنظمة ضمان الجودة.
باختصار، ستوفر لهم معرفة نظرية وعملية واسعة ومتخصصة بحيث يفهمون تطوير المشاريع من منظور حديث وفعال.

ستتمكن من الوصول بسهولة إلى جميع المحتويات
وقتما تشاء. من جهاز الكمبيوتر أو الجهاز المفضل
لديك. يمكنك أيضاً تنزيلها من أجل استشارتك القادمة"

الأهداف العامة

- تطوير معايير ومهام ومنهجيات متقدمة لفهم أهمية العمل الموجه نحو الجودة
- تحليل العوامل الرئيسية في جودة مشروع البرمجيات software
- تطوير الجوانب المعيارية ذات الصلة
- تنفيذ عمليات وأنظمة DevOps لضمان الجودة
- الحد من الديون الفنية للمشاريع مع التركيز على الجودة بدلاً من النهج القائم على الاقتصاد والأطر الزمنية القصيرة
- تزويد الطالب بالمعرفة المتخصصة ليكون قادراً على قياس وتقدير جودة مشروع برمجيsoftware
- الدفاع عن مقترحات المشاريع الاقتصادية على أساس الجودة

الأهداف المحددة

الوحدة 1. الجودة في تطوير البرمجيات (Software). مستويات مستوى التطور TRL

- تطوير العناصر التي تتألف منها الجودة في تطوير البرمجيات (Software) بطريقة واضحة وموجزة.
- تطبيق النماذج والمعايير كدالة للنظام والمنتج وعملية software.
- معرفة متعمقة بمعايير الجودة ISO المطبقة بشكل عام وفي أجزاء محددة
- تطبيق المعايير وفقًا للبيئة (المحلية والوطنية والدولية).
- فحص مستويات نضج مستوى النضج TRL وتكييفها مع الأجزاء المختلفة من مشروع software المراد معالجتها.
- اكتساب القدرة على التجريد لتطبيق معيار أو أكثر من معايير عناصر ومستويات الجودة في تطوير البرمجيات (Software).
- تمييز حالات تطبيق المعايير ومستويات النضج في مشروع محاكاة لحالة حقيقية.

الوحدة 2. تطوير مشاريع البرمجيات. الوثائق الوظيفية والتقنية

- تحديد تأثير إدارة المشروع على الجودة
- تطوير المراحل المختلفة للمشروع
- التمييز بين مفاهيم الجودة المتأصلة في التوثيق الوظيفي والتقني.
- تحليل مرحلة أخذ المتطلبات ومرحلة التحليل وإدارة الفريق ومرحلة البناء.
- إنشاء المنهجيات المختلفة لإدارة مشاريع البرمجيات software.
- وضع معايير لتحديد المنهجية الأنسب اعتماداً على نوع المشروع.

الوحدة 3. Testing للبرمجيات Software. أتمتة الاختبارات

- تحديد الاختلافات بين جودة المنتج وجودة العملية والجودة في الاستخدام.
- الإلمام بمعايير ISO/IEC 15504
- تحديد تفاصيل CMMI
- التعرّف على مفاتيح التكامل المستمر والمستودعات وتأثيرها على فريق تطوير البرمجيات software.
- إثبات أهمية دمج المستودعات من قبل مشاريع البرمجيات software. تعلم كيفية إنشائها باستخدام TFS
- فهم أهمية قابلية توسع البرمجيات software في تصميم وتطوير نظم المعلومات.

الوحدة 4. منهجيات إدارة مشاريع البرمجيات Software. المنهجيات Waterfall مقابل المنهجيات الرشيقة

- تحديد ما تتكون منه منهجيةWaterfall
- التعمق في منهجية Scrum
- تحديد الاختلافات بين Scrumg Waterfall
- تحديد الاختلافات بين منهجيات الشلال و سكروموكيف يراها العميل.
- تصفح لوحة Kanban
- نهج الشلال ونهج سكروم للمشروع نفسه
- إعداد مشروع هجين

الوحدة 5. TDD Test Driven Developement. تصميم البرمجيات Software المدفوعة بالاختبار

- التعرف على التطبيق العملي لـ TDD وإمكانياته لاختبار مشروع برمجي software في المستقبل.
- استكمال حالات المحاكاة الحقيقية المقترحة، كتعلم مستمر لمفهوم TDD هذا.
- تحليل، في حالات المحاكاة، مدى نجاح الاختبارات أو فشلها، من وجهة نظر بناءة
- تحديد بدائل TDD، وإجراء تحليل مقارن بينها

الوحدة 6. DevOps. إدارة الجودة في تطوير البرمجيات (Software)

- تحليل أوجه القصور في العملية التقليدية
- تقييم الحلول الممكنة واختيار الأنسب منها
- فهم احتياجات العمل وتأثيرها على التنفيذ
- تقييم تكاليف التحسينات التي سيتم تنفيذها
- تطوير دورة حياة برمجيات software قابلة للتطوير، تتكيف مع الاحتياجات الحقيقية
- توقع الأخطاء المحتملة وتجنبها من عملية التصميم
- تبرير استخدام نماذج التنفيذ المختلفة

الوحدة 7. DevOps والتكامل المستمر. الحلول العملية المتقدمة في تطوير البرمجيات Software

- تحديد مراحل تطوير البرمجيات software ودورة التسليم المكيفة مع حالات معينة
- تصميم عملية تسليم البرمجيات software من خلال التكامل المستمر
- بناء وتنفيذ التكامل والنشر المستمر بناءً على تصميمك السابق
- إنشاء نقاط فحص تلقائية للجودة عند كل عملية تسليم برمجيات software
- الحفاظ على عملية تسليم برمجيات software تلقائية وقوية
- تكييف الاحتياجات المستقبلية مع عملية التكامل والنشر المستمرين
- تحليل الثغرات الأمنية وتوقعها أثناء عملية تسليم البرمجيات software وبعد التسليم

الوحدة 8. تصميم قاعدة البيانات. التوحيد والأداء القياسي. الجودة في تطوير البرمجيات (Software)

- تقييم استخدام نموذج العلاقة بين الكيانات والعلاقة بين الكيانات للتصميم الأولي لقاعدة البيانات
- تطبيق كيان أو سمة أو مفتاح، وما إلى ذلك، لتحسين تكامل البيانات
- تقييم التبعيات والنماذج وقواعد تطبيع قاعدة البيانات
- تخصص في تشغيل نظام مستودع بيانات OLAP، وتطوير واستخدام كل من جدول الحقائق وجدول الأبعاد
- تحديد النقاط الرئيسية لأداء قاعدة البيانات
- إكمال حالات المحاكاة الحقيقية المقترحة، كتعلم مستمر على تصميم قواعد البيانات وتوحيدها وأدائها
- وضع في حالات المحاكاة، الخيارات التي يجب حلها في إنشاء قاعدة البيانات من وجهة نظر بنائية

الوحدة 9. تصميم البنى القابلة للتطوير. البنية في دورة حياة البرمجيات Software

- تطوير مفهوم بنية البرمجيات software وخصائصها
- تحديد الأنواع المختلفة لقابلية التوسع في بنية البرمجيات software
- تحليل المستويات المختلفة التي يمكن أن تحدث في قابلية توسع الويب
- اكتساب معرفة متخصصة بمفهوم دورة حياة البرمجيات software ومراحلها ونماذجها
- تحديد تأثير البنية على دورة حياة البرمجيات software، مع مزاياها وقيودها والأدوات الداعمة لها
- استكمال حالات المحاكاة الحقيقية المقترحة، كتعلم مستمر لبنية البرمجيات software ودورة حياتها
- تقييم، في حالات المحاكاة، إلى أي مدى قد تجعل تصميم الهندسة المعمارية مجدياً أو غير ضروري

الوحدة 10. معايير الجودة ISO/IEC 9126 ISO/IEC 9126. مقاييس الجودة في تطوير البرمجيات (Software)

- تطوير مفهوم معايير الجودة والجوانب ذات الصلة
- مراجعة المواصفة ISO/IEC 9126، الجوانب والمؤشرات الرئيسية
- تحليل القياسات المختلفة لمشروع برمجيات software لتلبية التقييمات المتفق عليها
- دراسة السمات الداخلية والخارجية التي يجب معالجتها في جودة مشروع البرمجيات software
- التمييز بين المقاييس وفقًا لنوع البرمجة (المهيكلة، الموجهة للكائنات، الطبقات، إلخ)
- إكمال حالات المحاكاة الحقيقية كتعلم مستمر في قياس الجودة
- الرؤية في حالات المحاكاة إلى أي مدى يكون ذلك ممكنًا أو غير ضروري؛ أي من وجهة نظر المؤلفين البناءة

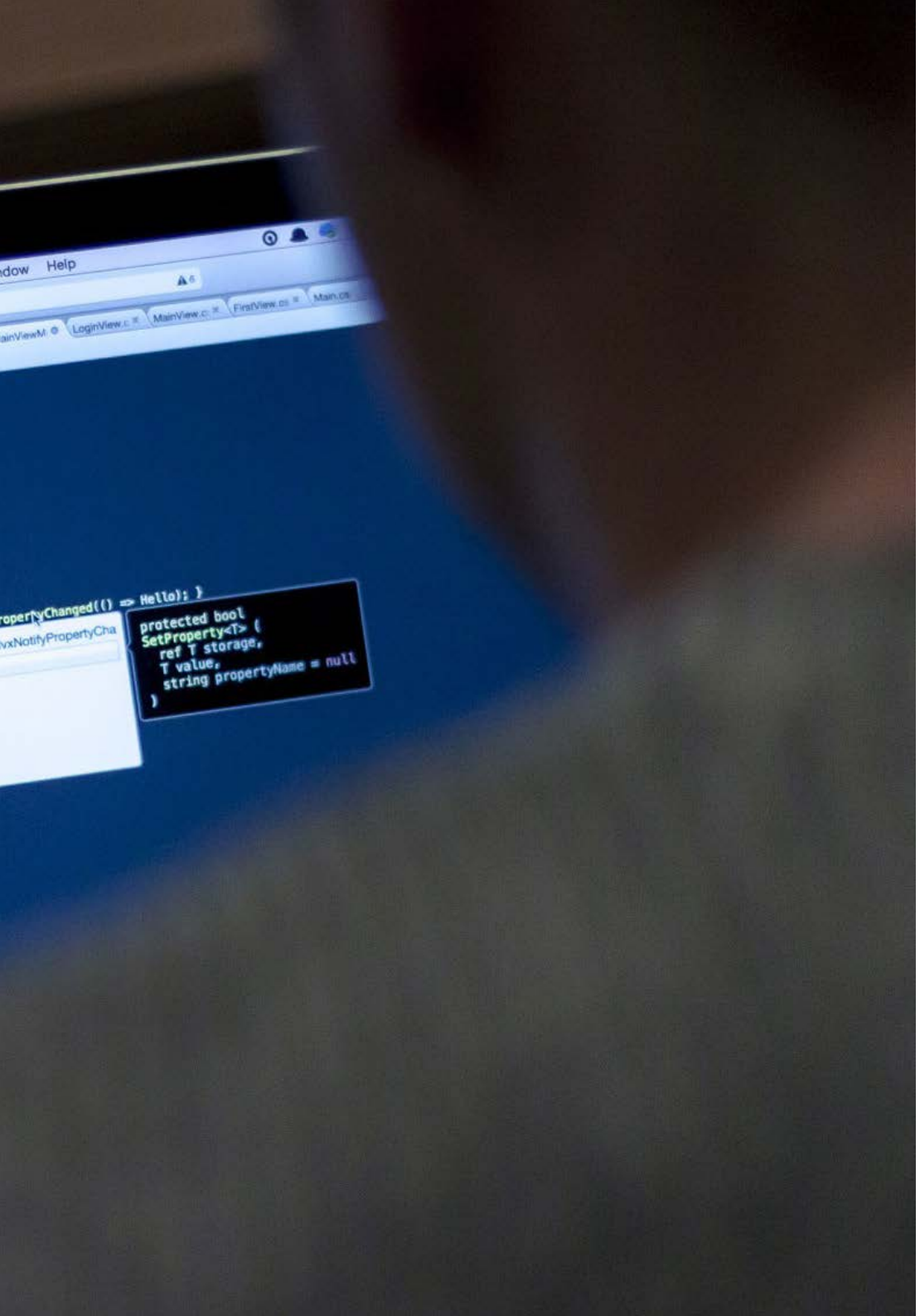
قم بإبراز سيرتك الذاتية المهنية من خلال هذا التدريب الحصري. احصل على شهادتك في 12 شهرًا وبطريقة عملية مع المنهجية التي لا يمكن أن تقدمها لك سوى TECH الجامعة التكنولوجية"

03

الكفاءات

سيكون خريجو الماجستيرالخاص في الجودة في تطوير البرمجيات (Software) قد أتقنوا هذا الموضوع من المنظور التقني والإداري. القدرة على تطوير النهج المتبع في المشروع، فضلاً عن تنفيذه ووضع بنية مستدامة وفعالة وذات جودة عالية لمشاريع البرمجيات software المقدمة إليه. لتحقيق هذه الغاية، سكب أعضاء هيئة التدريس كل خبراتهم الشخصية في وضع العديد من الحالات العملية، والتي ستكون بمثابة سياق وتطور في مواجهة ذلك "الدين التقني" الذي لن يكون موجودًا بعد الآن.

يُعد محترف تكنولوجيا المعلومات الذي يركز على الجودة من الأصول الصاعدة في مجال استشارات البرمجيات software والشركات الكبيرة. سجل الآن في الماجستيرالخاص في الجودة في تطوير البرمجيات (Software)"



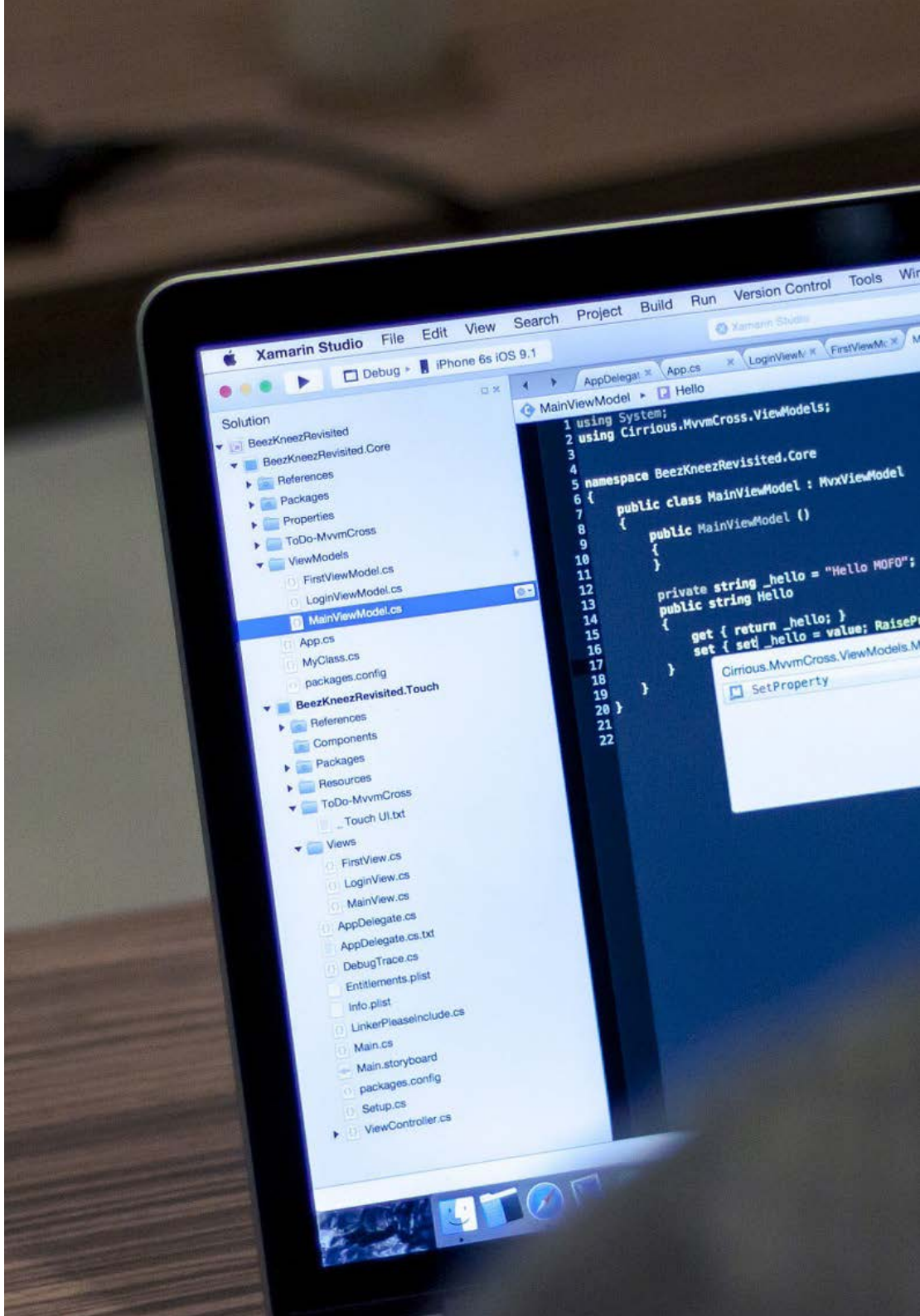
الكفاءات العامة

- الحد من الديون الفنية للمشاريع مع التركيز على الجودة بدلاً من النهج القائم على الاقتصاد والأطر الزمنية القصيرة
- قياس وقياس جودة مشروع البرمجيات software وتحديدها كمياً
- تنفيذ TDD بشكل صحيح، بحيث يتم رفع معايير الجودة في تطوير البرمجيات (Software)
- تبرير وضع ميزانية المشاريع الموجهة نحو الجودة في الميزانية
- تطوير قواعد ونماذج ومعايير الجودة
- فحص تقييمات النضج التكنولوجي المختلفة
- الحد من المخاطر وضمان صيانة الإصدارات اللاحقة والتحكم فيها
- إتقان المراحل التي يتم تقسيم المشروع إليها

"قم بتعزيز مهاراتك واكتشف إمكانيات لا حصر لها للنمو المهني التي تنفتح مع هذه التجربة الجديدة"

الكفاءات المحددة

- تقييم نظام البرمجيات software من حيث درجة التقدم في عملية المشروع
- معالجة قضايا الموثوقية والمقاييس والضمان هذه في مشاريع البرمجيات software بشكل صحيح واستراتيجي
- تناول عملية اتخاذ القرار بشأن المنهجية التي سيتم استخدامها في المشروع
- إتقان الجوانب التنظيمية الضرورية لإنشاء البرمجياتsoftware
- التطوير التلقائيTesting
- إقامة تواصل مناسب مع العميل، وفهم كيف ينظر العميل إلى المشروع وفقاً للمنهجية المطبقة
- وضع قائمة بمتطلبات الاختبار
- إجراء التجريد والتقسيم إلى اختبارات أكثر وحدوية وإزالة ما لا ينطبق على الأداء الجيد لاختبارات مشروع البرمجيات software المراد تنفيذها
- تحديث قائمة متطلبات الاختبار بطريقة مدروسة وصحيحة
- تكييف ثقافة DevOps مع احتياجات العمل
- تطوير أحدث الممارسات والأدوات في التكامل والنشر المستمر
- إعادة الهيكلة والتعامل مع إدارة البيانات وتنسيقها



04

هيكل الإدارة وأعضاء هيئة تدريس الدورة التدريبية

يقوم المدرسون الخبراء الذين لديهم منهج دراسي مكثف في مجال حلول تكنولوجيا المعلومات وتطوير البرمجيات software والبحوث، بتوجيه الماجستيرالخاص هذا لتوفير الأدوات والمعرفة اللازمة للخريج المستقبلي الذي يركز على الجودة في عمليات تطوير البرمجيات software والأدوات الأكثر تقدماً لتنفيذ عمليات وأنظمة DevOps لضمان الجودة. سيقوم فريق من المتخصصين بتوجيه الطالب في جميع الأوقات، من أجل تحقيق الأهداف عن بُعد، حيث أنه برنامج إلكتروني بحت ويتبع المنهجية الأكثر تطوراً التي تنفذها جامعة TECH.

يلتزم المعلمون المتخصصون بتزويدك بأفضل محتوى وجعل عملية التعلم تجربة مرنة وديناميكية. أحصل على إجابات لشكوكك ومرافقتك على طول الطريق"

هيكل الإدارة

أ. Molina Molina, Jerónimo

- رئيس قسم الذكاء الاصطناعي في Helphone
- مهندس الذكاء الاصطناعي ومهندس البرمجيات في ناسات، إنترنت الأقمار الصناعية المتنقلة
- استشاري أول في شركة Hexa Ingeniero
- مُقدِّم الذكاء الاصطناعي (التعلم الآلي والسيرة الذاتية)
- خبير في الحلول القائمة على الذكاء الاصطناعي في مجالات الرؤية الحاسوبية وتعلم الآلة/تعلم الآلة ML/DLوالبرمجة اللغوية العصبية NLP
- شهادة الخبرة الجامعية في إنشاء وتطوير الأعمال التجارية في Bancaixa و Fundeun
- مهندس كمبيوتر من جامعة Alicante
- ماجستير في الذكاء الاصطناعي من الجامعة الكاثوليكية في Ávila
- مدير تنفيذي في ماجستير إدارة الأعمال في المنتدى الأوروبي لرجال الأعمال

الأساتذة

أ. Tenrero Morán, Marcos

- مهندس DevOps في شركة Allot Communications
- مدير إدارة دورة حياة التطبيقات في Cegid Meta4
- مهندس أتمتة ضمان الجودة في Cegid Meta4
- الماجستيرالخاص في تطوير التطبيقات الاحترافية للأندرويد من جامعة Galileo. غواتيمالا
- ماجستير في تطوير الخدمات السحابية، Node.Js، JavaScript، HTML5 من جامعة بوليتكنيك بمدريد
- تطوير الويب باستخدام (Angular-CLI 4) و Ionic و Node.Js، Meta4 من جامعة Rey Juan Carlos
- خريج هندسة الحاسب الآلي من جامعة Rey Juan Carlos

أ. Martínez Cerrato, Yésica

- خبيرة في تحليلات الأعمال وإدارة نظم المعلومات
- Product Manager في الأمن الإلكتروني في Securitas Direct
- مديرة مشروع في مجال إدماج الحسابات الكبيرة في البريد
- محللة ذكاء الأعمال في Ricopia Technologies
- أستاذة في الدراسات الجامعية و بعد الجامعية
- بكالوريوس في هندسة الاتصالات السلكية واللاسلكية من جامعة ألكالا

أ. Pi Morell, Oriol

- محلل وظيفي فيFihoca
- Product Owner de Hosting والبريد الإلكتروني في CDmon
- محلل وظيفي Software Engineerg في Capgeminig Atmira
- محاضر و Atmira ،Capgemini ،Capgemini Forms
- شهادة في الهندسة التقنية في إدارة الكمبيوتر من جامعة برشلونة المستقلة
- ماجستير في الذكاء الاصطناعي من الجامعة الكاثوليكية في Ávila
- MBA في إدارة الأعمال والإدارة من مؤسسة IMF Smart Education
- ماجستير في إدارة نظم المعلومات من مؤسسة IMF Smart Education
- الدراسات العليا في أنماط التصميم من Universitat Oberta de Catalunya

د. Peralta Martín-Palomino، Arturo

- الرئيس التنفيذي CEO ومدير قسم التكنولوجيا CTO في Prometeus Global Solutions
- مدير قسم التكنولوجيا في Korporate Technologies
- مدير قسم التكنولوجيا في AI Shepherds GmbH
- مرشد ومستشار الأعمال الاستراتيجية في Alliance Medical
- مدير التصميم والتطوير في DocPath
- دكتور في هندسة الحاسوب من جامعة Castilla-La Mancha
- دكتور في الاقتصاد والأعمال والماليات من جامعة Camilo José Cela
- دكتور في علم النفس من جامعة Castilla-La Mancha
- الماجستير التنفيذي MBA من جامعة Isabel I
- ماجستير في الإدارة التجارية والتسويق من جامعة Isabel I
- ماجستير خبير في البيانات الضخمة Big Data من تدريب Hadoop
- ماجستير في تقنيات الكمبيوتر المتقدمة من جامعة Castilla-La Mancha
- عضو في: مجموعة البحوث SMILE

أ. Soto Jiménez, Manuel

- Lynx Financial Crime Tech في مجموعة Santander
- شهادة في هندسة الحاسب الآلي من جامعة مدريد المستقلة
- بكالوريوس في الرياضيات من جامعة مدريد المستقلة
- دورة الكم101 Quantum الشهادة المهنية في Quantum Computing & Quantum Internet من جامعة Delft التقنية
- دورة في التعلم العميق Deep Learning باستخدام TensorFlow من IBM
- لغات البرمجة: Python و R و C و SQL و MongoDB و Matlab و Sage و Cypher و VHDL و Prolog و Javascript و CSS. لغات الترميز: تخفيض السعر، HTML، لاتكس

> سيزودك فريق التدريس لدينا بكل معارفه حتى تكون على اطلاع بأحدث المعلومات حول هذا الموضوع"

05

الهيكل والمحتوى

لقد تم تطوير هيكل ومحتويات الماجستيرالخاص هذه لتغطية أهم الموضوعات لتطوير برمجيات Software مع الجودة. يتألف البرنامج من 10 وحدات تعليمية، بدءاً من تطوير المشاريع البرمجية software، والتوثيق الوظيفي والتقني، و test Driven Developement والمنهجيات المختلفة، وصولاً إلى تنفيذ الحلول العملية المتقدمة مع DevOps والتكامل المستمر، وكلها تعتمد على تحقيق الجودة في تطوير البرمجيات (Software). سيكون المحتوى الشامل للوسائط المتعددة، الذي تم اختياره بدقة من قبل محاضرين خبراء، داعمًا كبيرًا لتخفيف العبء التدريسي وسيكون بمثابة مادة مرجعية للرجوع إليها في المستقبل.

ستعمل الحالات العملية، المستندة إلى الواقع، على تعزيز جميع النظريات التي تم تعلمها خلال البرنامج ووضعها في سياقها"

الوحدة 1. الجودة في تطوير البرمجيات (Software). مستويات مستوى التطور TRL

1.1. العناصر المؤثرة في جودة البرمجيات (1) الدين الفني
1.1.1. الدين الفني الأسباب والعواقب
2.1.1. الجودة في تطوير البرمجيات (Software). مبادئ عامة
3.1.1. برامج Software الجودة غير المبدئية والمبدئية
1.3.1.1. العواقب
2.3.1.1. الحاجة إلى تطبيق مبادئ الجودة في تطوير البرمجيات (Software)
4.1.1. الجودة في تطوير البرمجيات (Software). الأنماط
5.1.1. Softwareعالية الجودة. ميزات محددة
2.1. العناصر المؤثرة في الجودة في تطوير البرمجيات (Software) (2). التكاليف المرتبطة
1.2.1. الجودة في تطوير البرمجيات (Software). العناصر المؤثرة
2.2.1. الجودة في تطوير البرمجيات (Software). المفاهيم الخاطئة
3.2.1. الجودة في تطوير البرمجيات (Software). التكاليف المرتبطة
3.1. نماذج الجودة في تطوير البرمجيات (Software) (1). إدارة المعرفة
1.3.1. نماذج الجودة العامة
1.1.3.1. إدارة الجودة الشاملة
2.1.3.1. نموذج التميز في الأعمال الأوروبي (EFQM)
3.1.3.1. نموذج الستة سيجما
2.3.1. نماذج إدارة المعرفة
1.2.3.1. موديلو Dyba
2.2.3.1. نموذج SEKS
3.3.1. مصنع الخبرة ونموذج مشاريع الأثر السريع QIP
4.3.1. نماذج الجودة في الاستخدام (25001)
4.1. نماذج الجودة في تطوير البرمجيات (Software) (3). الجودة في البيانات والعمليات والنماذج SEI
1.4.1. نموذج جودة البيانات
2.4.1. نمذجة عملية software
3.4.1. SPEM Software & Systems Process Engineering Metamodel Specification
4.4.1. نماذج SEI
1.4.4.1. CMMI
2.4.4.1. SCAMPI
3.4.4.1. IDEAL

5.1. معايير الجودة في تطوير البرمجيات Software ISO (1). تحليل المعايير
1.5.1. معايير ISO 9000
1.1.5.1. معايير ISO 9000
2.1.5.1. عائلة معايير الجودة ISO (9000)
2.5.1. معايير ISO الأخرى المتعلقة بالجودة
3.5.1. معايير نمذجة الجودة (ISO 2501)
4.5.1. معايير قياس الجودة (n2502 ISO)
6.1. معايير الجودة في تطوير البرمجيات Software ISO (2). المتطلبات والتقييم
1.6.1. المعايير الخاصة بمتطلبات الجودة (n 2503)
2.6.1. معايير تقييم الجودة (n2504)
3.6.1. ISO/ISO/EC 2007:24744.
7.1. مستويات تطوير TRL (1). المستويات من 1 إلى 4
1.7.1. مستويات TRL
2.7.1. المستوى 1: المبادئ الأساسية
3.7.1. المستوى 2: المفهوم و/أو التطبيق
4.7.1. المستوى 3: الوظيفة التحليلية الحرجة
5.7.1. المستوى 4: التحقق من صحة المكونات في بيئة معملية
8.1. مستويات تطوير TRL (2). المستويات من 5 إلى 9
1.8.1. المستوى 5: التحقق من صحة المكونات في البيئة ذات الصلة
2.8.1. المستوى 6: نموذج النظام/النظام الفرعي
3.8.1. المستوى 7: عرض توضيحي في بيئة حقيقية
4.8.1. المستوى 8: نظام كامل ومعتمد
5.8.1. المستوى 9: النجاح في العالم الحقيقي
9.1. مستويات تطوير TRL. الاستخدامات
1.9.1. مثال على شركة ذات بيئة معملية
2.9.1. مثال على شركة للبحث والتطوير والابتكار
3.9.1. مثال على شركة صناعية للبحث والتطوير والابتكار
4.9.1. مثال على شركة هندسية مختبرية هندسية مشتركة

10.1. الجودة في تطوير البرمجيات (Software). التفاصيل الرئيسية
1.10.1. التفاصيل المنهجية
2.10.1. التفاصيل الفنية
3.10.1. التفاصيل في إدارة مشاريع software
1.3.10.1. جودة أنظمة تكنولوجيا المعلومات
2.3.10.1. جودة المنتج software
3.3.10.1. جودة عملية software

الوحدة 2. تطوير مشاريع البرمجيات. التوثيق الوظيفي والتقني

1.2. إدارة المشاريع
1.1.2. إدارة المشروع في الجودة في تطوير البرمجيات (Software)
2.1.2. إدارة مشاريع. المزايا
3.1.2. إدارة مشاريع. الأنماط
2.2. المنهجية في إدارة المشاريع
1.2.2. المنهجية في إدارة المشاريع
2.2.2. منهجيات المشروع. الأنماط
3.2.2. المنهجيات في إدارة المشاريع. التطبيق
3.2. مرحلة تحديد المتطلبات
1.3.2. تحديد متطلبات المشروع
2.3.2. إدارة اجتماعات المشروع
3.3.2. الوثائق الواجب تقديمها
4.2. النموذج
1.4.2. المرحلة الأولى
2.4.2. مرحلة التحليل
3.4.2. مرحلة البناء
4.4.2. مرحلة الإختبار
5.4.2. تسليم
5.2. نموذج البيانات الذي سيتم استخدامه
1.5.2. تحديد نموذج البيانات الجديد
2.5.2. تحديد خطة ترحيل البيانات
3.5.2. مجموعة البيانات

6.2. التأثير على المشاريع الأخرى
1.6.2. تأثير المشروع. الأمثلة
2.6.2. المخاطر في المشروع
3.6.2. إدارة المخاطر
7.2. المشروع "Must"
1.7.2. المشروع Must
2.7.2. تحديد المشروع Must
3.7.2. تحديد نقاط التنفيذ الخاصة بتسليم المشروع
8.2. فريق إنشاء المشروع
1.8.2. أدوار التدخل حسب المشروع
2.8.2. اتصل بالموارد البشرية للتوظيف
3.8.2. نواتج المشروع والجدول الزمني
9.2. الجوانب التقنية لمشروع البرمجيات software
1.9.2. مهندس المشروع. الجوانب الفنية
2.9.2. القادة الفنيون
3.9.2. بناء مشروع البرنامج software
4.9.2. تقييم جودة الكود، السونار الصوتي
10.2. نواتج المشروع
1.10.2. التحليل الوظيفي
2.10.2. نموذج البيانات
3.10.2. مخطط الحالة
4.10.2. الوثائق الفنية

الوحدة 3. Testing للبرمجيات Software. أتمتة الاختبارات

1.3. نماذج جودة software
1.1.3. جودة المنتج
2.1.3. جودة العملية
3.1.3. جودة الاستخدام
2.3. جودة العملية
1.2.3. جودة العملية
2.2.3. نماذج النضج
3.2.3. معيار ISO 15504
1.3.2.3. الغرض
2.3.2.3. السياق
3.3.2.3. المراحل
3.3. المعيارية ISO/IEC 15504
1.3.3. فئات العمليات
2.3.3. عملية التطوير. مثال
3.3.3. جزء الملف الشخصي
4.3.3. المراحل
4.3. CMMI (دمج نموذج نضج القدرات المتكاملة)
1.4.3. CMMI. دمج نماذج نضج القدرات المتكاملة
2.4.3. النماذج والمناطق. الأنماط
3.4.3. مجالات العملية
4.4.3. مستويات السعة
5.4.3. إدارة العمليات
6.4.3. إدارة المشاريع
5.3. إدارة التغيير والمستودعات
1.5.3. إدارة تغيير البرمجيات
1.1.5.3. عنصر التكوين. التكامل المستمر
2.1.5.3. الخطوط
3.1.5.3. مخططات انسيابية
4.1.5.3. الفروع
2.5.3. المستودع
1.2.5.3. التحكم في الإصدار
2.2.5.3. فريق العمل واستخدام المستودع
3.2.5.3. التكامل المستمر في المستودع
6.3. Team Foundation Server TFS
1.6.3. التثبيت والتكوين
2.6.3. إنشاء مشروع جماعي
3.6.3. دمج المحتوى في التحكم في التعليمات البرمجية المصدرية
4.6.3. on CloudTFS
7.3. Testing
1.7.3. الدافع للاختبار
2.7.3. اختبارات التحقق
3.7.3. الاختبار التجريبي
4.7.3. التنفيذ والصيانة
8.3. اختبار الحمولة
1.8.3. Load testing
2.8.3. الاختبار باستخدام LoadView
3.8.3. الاختبار6K باستخدام Cloud
4.8.3. الاختبار باستخدام Loader
9.3. اختبارات الوحدة والإجهاد والتحمل
1.9.3. الدافع لاختبار الوحدة
2.9.3. أدوات Unit Testing
3.9.3. دوافع اختبارات الإجهاد
4.9.3. الاختبار باستخدام StressTesting
5.9.3. الدافع لاختبارات التحمل
6.9.3. الاختبار باستخدام LoadRunner
10.3. قابلية التوسع. تصميم برمجيات software قابلة للتطوير
1.10.3. قابلية التوسع وبنية البرمجيات software
2.10.3. الاستقلالية بين الطبقات
3.10.3. الاقتران بين الطبقات. الأنماط المعمارية

الوحدة 4. منهجيات إدارة مشاريع البرمجيات Software. المنهجياتWaterfall مقابل المنهجيات الرشيقة

1.4. منهجية Waterfall
1.1.4. منهجية Waterfall
2.1.4. منهجية Waterfall. التأثير على الجودة في تطوير البرمجيات (Software)
3.1.4. منهجية Waterfall. الأمثلة
2.4. منهجية بسيطة
1.2.4. منهجية بسيطة
2.2.4. المنهجية المرنة. التأثير على الجودة في تطوير البرمجيات (Software)
3.2.4. المنهجية المرنة. الأمثلة
3.4. منهجية Scrum
1.3.4. منهجية Scrum
2.3.4. بيان Scrum
3.3.4. تنفيذ Scrum
4.4. لوحة Kanban
1.4.4. طريقة Kanban
2.4.4. لوحة Kanban
3.4.4. لوحة Kanban. مثال على التطبيق
5.4. إدارة المشاريع فيWaterfall
1.5.4. مراحل المشروع
2.5.4. الرؤية في مشروعWaterfall
3.5.4. المنجزات التي يجب مراعاتها
6.4. إدارة المشروع في Scrum
1.6.4. المراحل في مشروع Scrum
2.6.4. الرؤية في مشروع Scrum
3.6.4. المنجزات التي يجب مراعاتها
7.4. Waterfall مقابل مقارنة Scrum
1.7.4. نهج المشروع التجريبي
2.7.4. مشروع تنفيذWaterfall. مثال
3.7.4. مشروع تطبيق Scrum. مثال
8.4. رؤية العميل
1.8.4. المستندات في Waterfall
2.8.4. المستندات في Scrum
3.8.4. مقارنة
9.4. هيكل Kanban
1.9.4. قصص المستخدمين
2.9.4. Backlog
3.9.4. تحليل Kanban
10.4. المشاريع الهجينة
1.10.4. إنشاء المشروع
2.10.4. إدارة المشاريع
3.10.4. المنجزات التي يجب مراعاتها

الوحدة 5. TDD Test Driven Developement. تصميم Software المدفوعة بالاختبار

1.5. TDD. Test Driven Development
1.1.5. TDD. Test Driven Development
2.1.5. TDD. تأثير TDD على الجودة
3.1.5. التصميم والتطوير القائم على الأدلة. الأمثلة
2.5. دورة TDD
1.2.5. اختيار المتطلبات
2.2.5. الاختبار. الأنماط
1.2.2.5. اختبار الوحدة
2.2.2.5. اختبارات التكامل
3.2.2.5. اختبارات End To End
3.2.5. التحقق من الاختبار. الإخفاقات
4.2.5. إنشاء التنفيذ
5.2.5. تنفيذ الاختبارات الآلية
6.2.5. القضاء على الازدواجية
7.2.5. تحديث قائمة المتطلبات
8.2.5. كرر دورة TDD
9.2.5. دورة TDD. مثال نظري وعملي

3.5. استراتيجيات تنفيذ TDD
1.3.5. التنفيذ الخاطئ
2.3.5. التنفيذ الثلاثي
3.3.5. التنفيذ الواضح
4.5. TDD. الاستخدام المميزات والعيوب
1.4.5. مزايا الاستخدام
2.4.5. حدود الاستخدام
3.4.5. توازن الجودة في التنفيذ
5.5. TDD. الممارسات الجيدة
1.5.5. قواعد TDD
2.5.5. القاعدة 1: قم بإجراء اختبار سابق يفشل قبل الترميز في الإنتاج
3.5.5. القاعدة 2: لا تكتب أكثر من اختبار وحدة واحد فقط
4.5.5. القاعدة 3: لا تكتب كودًا برمجيًا أكثر من اللازم
5.5.5. الأخطاء والأنماط المضادة التي يجب تجنبها في تطوير TDD
6.5. محاكاة مشروع حقيقي لاستخدام TDD (1)
1.6.5. وصف عام للمشروع (الشركة أ)
2.6.5. تنفيذ TDD
3.6.5. التمارين المقترحة
4.6.5. التمارين. Feedback
7.5. محاكاة مشروع حقيقي لاستخدام TDD (2)
1.7.5. وصف عام للمشروع (الشركة ب)
2.7.5. تنفيذ TDD
3.7.5. التمارين المقترحة
4.7.5. التمارين. Feedback
8.5. محاكاة مشروع حقيقي لاستخدام TDD (3)
1.8.5. وصف عام للمشروع (الشركة ج)
2.8.5. تنفيذ TDD
3.8.5. التمارين المقترحة
4.8.5. التمارين. Feedback

9.5. بدائل TDD. Test Driven Development
1.9.5. TCR (Test Commit Revertl)
2.9.5. BDD Behavior Driven Development
3.9.5. ATDD Acceptance Test Driven Development
4.9.5. TDD. المقارنة النظرية
10.5. TDD TCR و BDD و ATDD. مقارنة عملية
1.10.5. تعريف المشكلة
2.10.5. الدقة مع TCR
3.10.5. الدقة مع BDD
4.10.5. الدقة مع ATDD

الوحدة 6. DevOps. إدارة الجودة في تطوير البرمجيات (Software)

1.6. DevOps. إدارة الجودة في تطوير البرمجيات (Software)
1.1.6. DevOps
2.1.6. DevOps والجودة في تطوير البرمجيات (Software)
3.1.6. DevOps. فوائد ثقافة DevOps
2.6. DevOps. العلاقة مع Agile
1.2.6. التسليم السريع
2.2.6. الجودة
3.2.6. تقليل التكاليف
3.6. تطبيق DevOps
1.3.6. تحديد المشاكل
2.3.6. التنفيذ في الشركة
3.3.6. مقاييس التنفيذ
4.6. دورة تسليم software
1.4.6. طرق التصميم
2.4.6. الاتفاقيات
3.4.6. خريطة الطريق
5.6. تطوير كود خالي من الأخطاء البرمجية
1.5.6. كود قابل للصيانة
2.5.6. أنماط التنمية
3.5.6. Testing الكود
4.5.6. تطوير software على مستوى التعليمات البرمجية. الممارسات الجيدة
6.6. التشغيل التلقائي
1.6.6. الأتمتة. أنواع الاختبارات
2.6.6. تكلفة الأتمتة والصيانة
3.6.6. الأتمتة. تخفيف الأخطاء
7.6. عمليات النشر
1.7.6. تقييم الأهداف
2.7.6. تصميم عملية تلقائية ومكيفة
3.7.6. الملاحظات والاستجابة
8.6. إدارة الحوادث
1.8.6. التأهب للحوادث
2.8.6. تحليل الحوادث وحلها
3.8.6. كيفية تجنب الأخطاء المستقبلية
9.6. أتمتة النشر
1.9.6. التحضير لعمليات النشر التلقائي
2.9.6. تقييم صحة العملية التلقائية
3.9.6. المقاييس والقدرة على التحول
10.6. الممارسة الجيدة. تطور DevOps
1.10.6. دليل أفضل ممارسات DevOps
2.10.6. DevOps. منهجية الفريق
3.10.6. تجنب المنافذ

الوحدة 7. DevOps والتكامل المستمر. الحلول العملية المتقدمة في تطوير Software

1.7. تدفق تسليم software
1.1.7. تحديد الجهات الفاعلة والمصنوعات اليدوية
2.1.7. تصميم تدفق تسليم البرامج
3.1.7. تدفق تسليم software. المتطلبات بين المراحل
2.7. أتمتة العمليات
1.2.7. التكامل المستمر
2.2.7. النشر المستمر
3.2.7. إعداد البيئات وإدارة الأسرار

3.7. خطوط الأنابيب التوضيحية
1.3.7. الاختلافات بين خطوط الأنابيب التقليدية الشبيهة بالرموز وخطوط الأنابيب التوضيحية
2.3.7. خطوط الأنابيب التوضيحية
3.3.7. خطوط الأنابيب التوضيحية في Jenkins
4.3.7. مقارنة بين مزودي خدمات التكامل المستمر
4.7. بوابات الجودة والتغذية الراجعة المثرية
1.4.7. أبواب عالية الجودة
2.4.7. معايير الجودة مع أبواب ذات جودة عالية. الصيانة
3.4.7. متطلبات العمل في طلبات التكامل
5.7. إدارة المصنوعات اليدوية
1.5.7. المصنوعات اليدوية ودورة الحياة
2.5.7. أنظمة تخزين القطع الأثرية وإدارتها
3.5.7. الأمن في إدارة القطع الأثرية
6.7. النشر المستمر
1.6.7. النشر المستمر في شكل حاويات
2.6.7. النشر المستمر مع المنصة كخدمة (PaaS)
3.6.7. النشر المستمر لتطبيقات الهاتف المحمول
7.7. تحسين وقت تشغيل خط الأنابيب: التحليل الثابت وخطافات Git Hooks
1.7.7. تحليل ثابت
2.7.7. قواعد نمط الكود
3.7.7. Git Hooks و اختبارات الوحدات
4.7.7. تأثير البنية التحتية
8.7. نقاط ضعف الحاويات
1.8.7. نقاط ضعف الحاويات
2.8.7. المسح الضوئي للصور
3.8.7. التقارير والتنبيهات الدورية

الوحدة 8. تصميم قاعدة البيانات. التوحيد والأداء القياسي. الجودة في تطوير البرمجيات (Software)

1.8. تصميم قاعدة البيانات
1.1.8. قواعد بيانات. الأنماط
2.1.8. قواعد البيانات المستخدمة حالياً
1.2.1.8. علاقية
2.2.1.8. قيمة المفتاح
3.2.1.8. قائم على الرسم البياني
3.1.8. جودة البيانات
2.8. تصميم نموذج العلاقة بين الكيان والعلاقة بين الكيانات (1)
1.2.8. نموذج العلاقة بين الكيان والعلاقة بين الكيانات. الجودة والتوثيق
2.2.8. المؤسسات
1.2.2.8. كيان قوي
2.2.2.8. كيان ضعيف
3.2.8. الخصائص
4.2.8. مجموعة من العلاقات
1.4.2.8. 1 إلى1
2.4.2.8. 1 إلى الكثير
3.4.2.8. الكثير إلى 1
4.4.2.8. الكثيرإلى الكثير
5.2.8. مفاتيح
1.5.2.8. المفتاح الأساسي
2.5.2.8. مفتاح أجنبي
3.5.2.8. المفتاح الأساسي للكيان الضعيف
6.2.8. القيود
7.2.8. الكاردينالية
8.2.8. الوراثة
9.2.8. التجميع

3.8. نموذج العلاقة بين الكيان والعلاقة بين الكيانات (2). الأدوات
1.3.8. نموذج العلاقة بين الكيان والعلاقة بين الكيانات. الأدوات
2.3.8. نموذج العلاقة بين الكيان والعلاقة بين الكيانات. مثال عملي
3.3.8. نموذج العلاقة بين الكيان والكيان القابل للتطبيق
1.3.3.8. العرض المرئي
2.3.3.8. عينة في التمثيل الجدولي
4.8. توحيد قاعدة البيانات (DB) (1). اعتبارات الجودة في تطوير البرمجيات (Software)
1.4.8. توحيد قاعدة البيانات والجودة
2.4.8. التبعية
1.2.4.8. الاعتماد الوظيفي
2.2.4.8. خصائص الاعتماد الوظيفي
3.2.4.8. الممتلكات المخصومة
3.4.8. مفاتيح
5.8. توحيد قاعدة البيانات (2). النماذج العادية وقواعد Codd
1.5.8. الأشكال العادية
1.1.5.8. الصيغة العادية الأولى
2.1.5.8. الصيغة العادية الثانية
3.1.5.8. الصيغة العادية الثالثة
4.1.5.8. الشكل الطبيعيBoyce-Codd
5.1.5.8. الشكل العادي الرابع
6.1.5.8. الصيغة العادية الخامسة
2.5.8. قواعد Codd
1.2.5.8. القاعدة 1: المعلومات
2.2.5.8. القاعدة 2: الوصول المضمون
3.2.5.8. القاعدة 3: المعالجة المنهجية للقيم الصفرية
4.2.5.8. القاعدة 4: وصف قاعدة البيانات
5.2.5.8. القاعدة 5: اللغة الفرعية المتكاملة
6.2.5.8. القاعدة 6: تحديث المشاهدات
7.2.5.8. القاعدة 7: الإدراج والتحديث
8.2.5.8. القاعدة 8: الاستقلالية الجسدية
9.2.5.8. القاعدة 9: الاستقلال المنطقي
10.2.5.8. القاعدة 10: استقلالية النزاهة
1.10.2.5.8. قواعد التكامل
11.2.5.8. القاعدة 11: التوزيع
21.2.5.8. القاعدة 12: عدم التخريب
3.5.8. مثال عملي
6.8. مستودع البيانات / نظام OLAP
1.6.8. مستودع البيانات
2.6.8. جدول الحقائق
3.6.8. جدول الأبعاد
4.6.8. إنشاء نظام OLAP. الأدوات
7.8. أداء قاعدة البيانات
1.7.8. تحسين الفهرس
2.7.8. تحسين الاستعلامات
3.7.8. تقسيم الجداول
8.8. محاكاة المشروع الحقيقي لتصميم قاعدة البيانات (1)
1.8.8. وصف عام للمشروع (الشركة أ)
2.8.8. تنفيذ تصميم قاعدة البيانات
3.8.8. التمارين المقترحة
4.8.8. التمارين المقترحة. Feedback
9.8. محاكاة المشروع الحقيقي لتصميم قاعدة البيانات (2)
1.9.8. وصف عام للمشروع (الشركة ب)
2.9.8. تنفيذ تصميم قاعدة البيانات
3.9.8. التمارين المقترحة
4.9.8. التمارين المقترحة. Feedback
10.8. أهمية تحسين قاعدة البيانات في الجودة في تطوير البرمجيات (Software)
1.10.8. تحسين التصميم
2.10.8. تحسين رمز الاستعلام
3.10.8. تحسين كود الإجراء المخزن
4.10.8. تأثير Triggers على الجودة في تطوير البرمجيات (Software). توصيات للاستخدام

الوحدة 9. تصميم البنى القابلة للتطوير. البنية في دورة حياة البرمجيات Software

1.9. تصميم البنى القابلة للتطوير (1)
1.1.9. البنى القابلة للتطوير
2.1.9. مبادئ البنية القابلة للتطوير
1.2.1.9. موثوقة
2.2.1.9. قابل للتطوير
3.2.1.9. قابلة للصيانة
3.1.9. أنواع قابلية التوسع
1.3.1.9. العمودي
2.3.1.9. الأفقي
3.3.1.9. مشترك
2.9. الهندسة المعمارية DDD Domain-Driven Design
1.2.9. نموذج DDD. توجيه المجال
2.2.9. الطبقات ومشاركة المسؤولية وأنماط التصميم
3.2.9. الفصل كأساس للجودة
3.9. تصميم البنى القابلة للتطوير (2). الفوائد والقيود واستراتيجيات التصميم
1.3.9. بنية قابلة للتطوير. الفوائد
2.3.9. بنية قابلة للتطوير. القيود
3.3.9. استراتيجيات تطوير البنى القابلة للتطوير (جدول وصفي)
4.9. دورة حياة software (1). المراحل
1.4.9. دورة حياة software
1.1.4.9. مرحلة التخطيط
2.1.4.9. مرحلة التحليل
3.1.4.9. مرحلة التصميم
4.1.4.9. مرحلة التنفيذ
5.1.4.9. مرحلة الاختبار
6.1.4.9. مرحلة التثبيت/النشر
7.1.4.9. مرحلة الاستخدام والصيانة
5.9. نماذج دورة حياة software
1.5.9. النموذج التعاقبي
2.5.9. النمط المتكرر
3.5.9. النموذج الحلزوني
4.5.9. نموذج Big Bang
6.9. دورة حياة software (2). التشغيل التلقائي
1.6.9. دورات حياة تطويرsoftware. الحلول
1.1.6.9. التكامل والتطوير المستمر (CI/CD)
2.1.6.9. المنهجيات الرشيقة
3.1.6.9. التطوير/الإنتاج DevOps
2.6.9. الاتجاهات المستقبلية
3.6.9. أمثلة عملية
7.9. بنية software في دورة حياة software
1.7.9. الفوائد
2.7.9. القيود
3.7.9. الأدوات
8.9. محاكاة المشروع الحقيقي لتصميم بنية software (1)
1.8.9. وصف عام للمشروع (الشركة أ)
2.8.9. تطبيق تصميم هندسة software
3.8.9. التمارين المقترحة
4.8.9. التمارين المقترحة. Feedback
9.9. محاكاة المشروع الحقيقي لتصميم بنية البرمجيات software (2)
1.9.9. وصف عام للمشروع (الشركة ب)
2.9.9. تطبيق تصميم هندسة software
3.9.9. التمارين المقترحة
4.9.9. التمارين المقترحة. Feedback
10.9. محاكاة المشروع الحقيقي لتصميم بنية البرمجيات software (3)
1.10.9. وصف عام للمشروع (الشركة ج)
2.10.9. تطبيق تصميم هندسة software
3.10.9. التمارين المقترحة
4.10.9. التمارين المقترحة. Feedback

الوحدة 10. معايير الجودة ISO، IEC 6219. مقاييس الجودة في تطوير البرمجيات (Software)

1.10. معايير الجودة. معيار ISO، IEC 9126
1.1.10. معايير الجودة
2.1.10. الجودة في تطوير البرمجيات (Software). المبرر. معيار ISO، IEC 9126
3.1.10. قياس الجودة في تطوير البرمجيات (Software) كمؤشر رئيسي لقياس جودة software
2.10. معايير الجودة في تطوير البرمجيات (Software). الخصائص
1.2.10. المصداقية
2.2.10. الوظائف
3.2.10. كفاءة
4.2.10. قابلية الاستخدام
5.2.10. قابلية الصيانة
6.2.10. قابليه
7.2.10. الأمان
3.10. المواصفة القياسية ISO، IEC 9126 (1). المقدمة
1.3.10. وصف المواصفة القياسية ISO، IEC 9126
2.3.10. الوظائف
3.3.10. المصداقية
4.3.10. قابلية الاستخدام
5.3.10. قابلية الصيانة
6.3.10. قابليه
7.3.10. الجودة في الاستخدام
8.3.10. مقاييس الجودة في تطوير البرمجيات (Software)
9.3.10. مقاييس الجودة في المواصفة القياسية ISO 9126
4.10. المواصفة القياسية ISO، IEC 9126 (2). نموذجا Boehmg McCall
1.4.10. نموذج ماكول: عوامل الجودة
2.4.10. نموذج Boehm
3.4.10. المستوى المتوسط. الخصائص
5.10. مقاييس الجودة في تطوير البرمجيات (Software) (1). العوامل
1.5.10. المقياس
2.5.10. المقاييس
3.5.10. المؤشر
1.3.5.10. أنواع المؤشرات
4.5.10. الأحجام والنماذج
5.5.10. نطاق مقاييس البرامج software
6.5.10. تصنيف مقاييس البرمجيات software
6.10. قياس الجودة في تطوير البرمجيات (Software) (2). ممارسة القياس
1.6.10. جمع البيانات المترية
2.6.10. قياس سمات المنتج الداخلية
3.6.10. قياس سمات المنتج الخارجية
4.6.10. قياس الموارد
5.6.10. مقاييس الأنظمة الموجهة للكائنات
7.10. تصميم مؤشر واحد للجودة في تطوير البرمجيات (Software)
1.7.10. مؤشر واحد كمؤشر واحد كمؤهل عام
2.7.10. وضع المؤشرات وتبريرها وتنفيذها
3.7.10. مثال على التطبيق. بحاجة إلى معرفة التفاصيل
8.10. محاكاة مشروع حقيقي لقياس الجودة (1)
1.8.10. وصف عام للمشروع (الشركة أ)
2.8.10. تطبيق قياس الجودة
3.8.10. التمارين المقترحة
4.8.10. التمارين المقترحة. Feedback
9.10. محاكاة مشروع حقيقي لقياس الجودة (2)
1.9.10. وصف عام للمشروع (الشركة ب)
2.9.10. تطبيق قياس الجودة
3.9.10. التمارين المقترحة
4.9.10. التمارين المقترحة. Feedback
10.10. محاكاة مشروع حقيقي لقياس الجودة (3)
1.10.10. وصف عام للمشروع (الشركة ج)
2.10.10. تطبيق قياس الجودة
3.10.10. التمارين المقترحة
4.10.10. التمارين المقترحة. Feedback

06

منهجية الدراسة

TECH هي أول جامعة في العالم تجمع بين منهجيةدراسات الحالة مع التعلم المتجدد، وهو نظام تعلم 100% عبر الإنترنت قائم، قائم على التكرار الموجهتم تصميم هذه الاستراتيجية التربوية المبتكرة لتوفير الفرصة للمهنيين لتحديث معارفهم وتطوير مهاراتهم بطريقة مكثفة ودقيقة. نموذج تعلُّم يضع الطالب في مركز العملية الأكاديمية ويمنحه كل الأهمية، متكيّفًا مع احتياجاته ومتخليًا عن المناهج الأكثر تقليدية

TECH تُعدُّك لمواجهة تحديات جديدة في بيئات غير مؤكدة
وتحقيق النجاح في مسيرتك المهنية"

الطالب: الأولوية في جميع برامج TECH

في منهجية الدراسة في TECH، يعتبر الطالب البطل المطلق.

تم اختيار الأدوات التربوية لكل برنامج مع مراعاة متطلبات الوقت والتوافر والدقة الأكاديمية التي، في الوقت الحاضر، لا يطلبها الطلاب فحسب، بل أيضًا أكثر المناصب تنافسية في السوق

مع نموذج TECH التعليمي غير المتزامن، يكون الطالب هو من يختار الوقت الذي يخصصه للدراسة، وكيف يقرر تنظيم روتينه، و كل ذلك من الجهاز الإلكتروني المفضّل لديه. لن يحتاج الطالب إلى حضور دروس مباشرة، والتي غالبًا ما لا يستطيع حضورها. سيقوم بأنشطة التعلم عندما يناسبه ذلك سيستطيع دائمًا تحديد متى وأين يدرس

> في TECH لن تكون لديك دروس مباشرة (والتي لا يمكنك حضورها أبدًا لاحقًا)"

المناهج الدراسية الأكثر شمولاً على مستوى العالم

تتميز TECH بتقديم أكثر المسارات الأكاديمية اكتمالاً في المحيط الجامعي. يتم تحقيق هذه الشمولية من خلال إنشاء مناهج لا تغطي فقط المعارف الأساسية، بل تشمل أيضًا أحدث الابتكارات في كل مجال.

من خلال التحديث المستمر، تتيح هذه البرامج للطلاب البقاء على اطلاع دائم على تغييرات السوق واكتساب المهارات الأكثر قيمة لدى أصحاب العمل. وبهذه الطريقة، يحصل الذين ينهون دراساتهم في TECH الجامعة التكنولوجية على إعداد شامل يمنحهم ميزة تنافسية ملحوظة للتقدم في مساراتهم المهنية.

وبالإضافة إلى ذلك، سيتمكنون من القيام بذلك من أي جهاز، سواء كان حاسوبًا شخصيًا، أو جهازًا لوحيًا، أو هاتفًا ذكيًا.

> نموذج TECH الجامعة التكنولوجية غير متزامن، مما يسمح لك بالدراسة باستخدام حاسوبك الشخصي، أو جهازك اللوحي، أو هاتفك الذكي أينما شئت، ومتى شئت، وللمدة التي تريدها"

Case studies أو دراسات الحالة

كانت طريقة الحالة هي نظام التعلم الأكثر استخداماً من قبل أفضل الكليات في العالم. قد كان منهج الحالة النظام التعليمي الأكثر استخدامًا من قبل أفضل كليات الأعمال في العالم. تم تطويره في عام 1912 لكي لا يتعلم طلاب القانون القوانين فقط على أساس المحتوى النظري، بل كان دوره أيضًا تقديم مواقف حقيقية معقدة لهم. وهكذا، يمكنهم اتخاذ قرارات وإصدار أحكام قيمة مبنية على أسس حول كيفية حلها. في عام 1924 تم تحديد هذه المنهجية كمنهج قياسي للتدريس في جامعة Harvard.

مع هذا النموذج التعليمي، يكون الطالب نفسه هو الذي يبني كفاءته المهنية من خلال استراتيجيات مثل **التعلم بالممارسة** أو التفكير التصميمي، والتي تستخدمها مؤسسات مرموقة أخرى مثل جامعة ييل أو ستانفورد. سيتم تطبيق هذه الطريقة، الموجهة نحو العمل، طوال المسار الأكاديمي الذي سيخوضه الطالب مع TECH الجامعة التكنولوجية.

سيتم تطبيق هذه الطريقة الموجهة نحو العمل على طول المسار الأكاديمي الكامل الذي سيخوضه الطالب مع TECH. وبهذه الطريقة سيواجه مواقف حقيقية متعددة، وعليه دمج المعارف والبحث والمجادلة والدفاع عن أفكاره وقراراته. كل ذلك مع فرضية الإجابة على التساؤل حول كيفية تصرفه عند مواجهته لأحداث معقدة محددة في عمله اليومي.

طريقة Relearning

في TECH، يتم تعزيز دراسات الحالة بأفضل طريقة تدريس عبر الإنترنت بنسبة 100%: **إعادة التعلم**.

هذه الطريقة تكسر الأساليب التقليدية للتدريس لوضع الطالب في مركز المعادلة، وتزويده بأفضل المحتويات في صيغ مختلفة. بهذه الطريقة، يتمكن من مراجعة وتكرار المفاهيم الأساسية لكل مادة وتعلم كيفية تطبيقها في بيئة حقيقية.

وفي هذا السياق، وبناءً على العديد من الأبحاث العلمية، يعتبر التكرار أفضل وسيلة للتعلم. لهذا السبب، تقدم TECH بين 8 و16 تكرارًا لكل مفهوم أساسي داخل نفس الدرس، مقدمة بطرق مختلفة، بهدف ضمان ترسيخ المعرفة تمامًا خلال عملية الدراسة.

ستتيح لك منهجية إعادة التعلم والمعروفة باسم Relearning، التعلم بجهد أقل ومزيد من الأداء، وإشراكك بشكل أكبر في تخصصك، وتنمية الروح النقدية لديك، وكذلك قدرتك على الدفاع عن الحجج والآراء المتباينة: إنها معادلة واضحة للنجاح.

حرم جامعي افتراضي 100% عبر الإنترنت مع أفضل الموارد التعليمية.

من أجل تطبيق منهجيته بفعالية، يركز برنامج TECH على تزويد الخريجين بمواد تعليمية بأشكال مختلفة: نصوص، وفيديوهات تفاعلية، ورسوم توضيحية وخرائط معرفية وغيرها.
تم تصميمها جميعًا من قبل مدرسين مؤهلين يركزون في عملهم على الجمع بين الحالات الحقيقية وحل المواقف المعقدة من خلال المحاكاة، ودراسة السياقات المطبقة على كل مهنة مهنية والتعلم القائم على التّكرار من خلال الصوتيات والعروض التقديمية والرسوم المتحركة والصور وغيرها.

تشير أحدث الأدلة العلمية في مجال علم الأعصاب إلى أهمية مراعاة المكان والسياق الذي يتم فيه الوصول إلى المحتوى قبل البدء في عملية تعلم جديدة.
إن القدرة على ضبط هذه المتغيرات بطريقة مخصصة تساعد الأشخاص على تذكر المعرفة وتخزينها في الحُصين من أجل الاحتفاظ بها على المدى الطويل.
هذا هو نموذج يسمى التعلم الإلكتروني المعتمد على السياق العصبي المعرفي العصبي، والذي يتم تطبيقه بوعي في هذه الدرجة الجامعية.

من ناحية أخرى، ومن أجل تفضيل الاتصال بين المرشد والمتدرب قدر الإمكان، يتم توفير مجموعة واسعة من إمكانيات الاتصال، سواء في الوقت الحقيقي أو المؤجل (الرسائل الداخلية، ومنتديات المناقشة، وخدمة الهاتف، والاتصال عبر البريد الإلكتروني مع مكتب السكرتير الفني، والدردشة ومؤتمرات الفيديو)

وبالمثل، سيسمح هذا الحرم الجامعي الافتراضي المتكامل للغاية لطلاب TECH بتنظيم جداولهم الدراسية وفقًا لتوافرهم الشخصي أو التزامات العمل.
وبهذه الطريقة، سيتمكنون من التحكم الشامل في المحتويات الأكاديمية وأدواتهم التعليمية، وفقًا لتحديثهم المهني المتسارع.

> ستسمح لك طريقة الدراسة عبر الإنترنت لهذا البرنامج بتنظيم وقتك ووتيرة تعلمك، وتكييفها مع جدولك الزمني"

تُبرر فعالية المنهج بأربعة إنجازات أساسية:

1. الطلاب الذين يتبعون هذا المنهج لا يحققون فقط استيعاب المفاهيم، ولكن أيضاً تنمية قدراتهم العقلية من خلال التمارين التي تقيم المواقف الحقيقية وتقوم بتطبيق المعرفة المكتسبة.

2. يركزمنهج التعلم بقوة على المهارات العملية التي تسمح للطالب بالاندماج بشكل أفضل في العالم الحقيقي.

3. يتم تحقيق استيعاب أبسط وأكثر كفاءة للأفكار والمفاهيم، وذلك بفضل منهج المواقف التي نشأت من الواقع.

4. يصبح الشعور بكفاءة الجهد المستثمر حافزًا مهمًا للغاية للطلاب، مما يترجم إلى اهتمام أكبر بالتعلم وزيادة في الوقت المخصص للعمل في المحاضرة الجامعية.

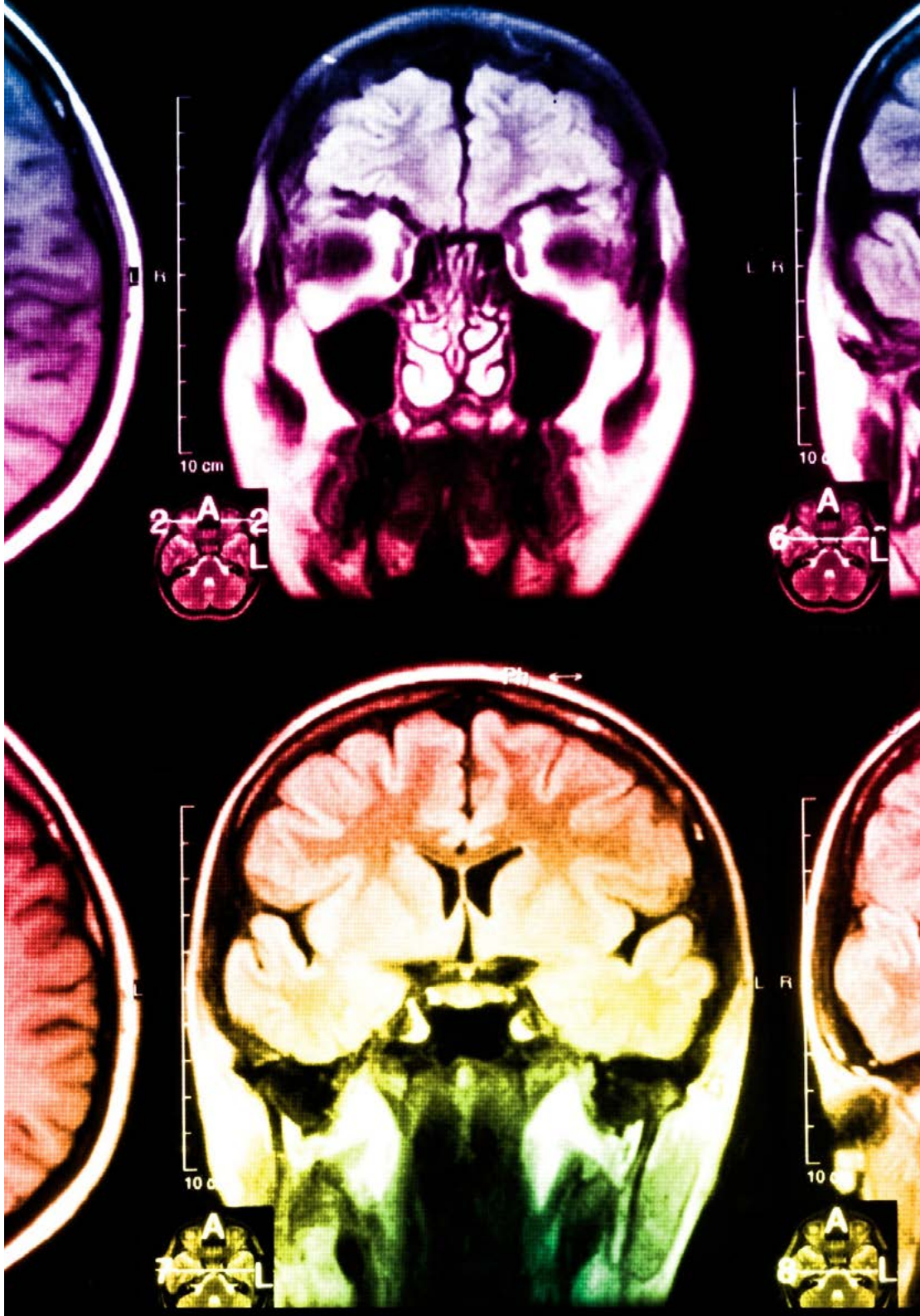
المنهجية الجامعية الأفضل تصنيفاً من قبل طلابها

نتائج هذا النموذج الأكاديمي المبتكريمكن ملاحضته في مستويات الرضا العام لخريجي TECH.

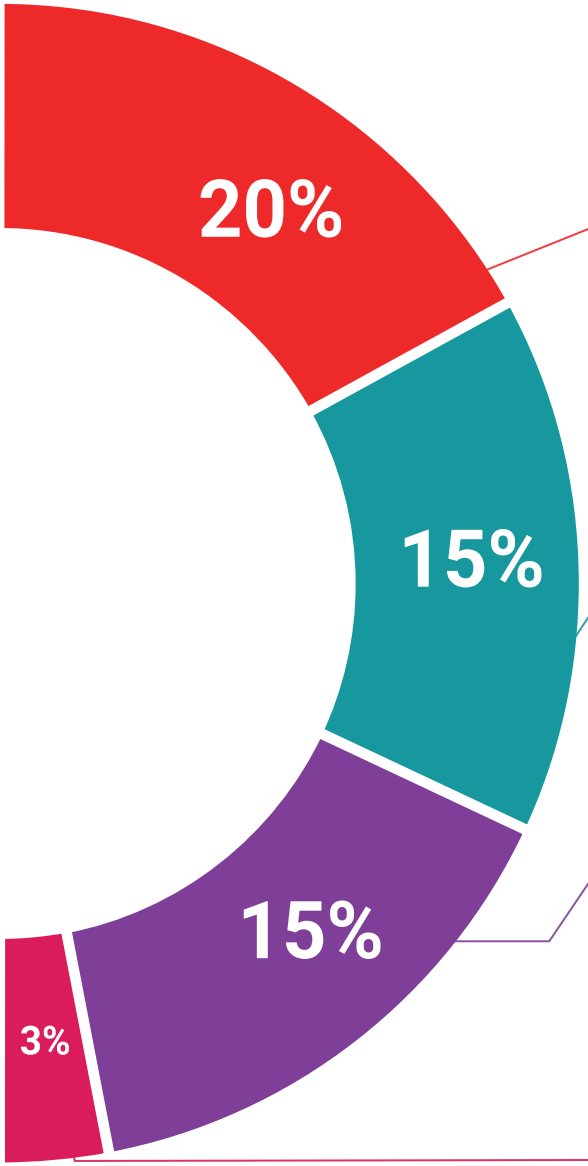
تقييم الطلاب لجودة التدريس، وجودة المواد، وهيكل الدورة وأهدافها ممتاز. ليس من المستغرب أن تصبح الجامعة الأعلى تقييماً من قِبل طلابها على منصة المراجعات Trustpilot، حيث حصلت على 4.9 من 5.

يمكنك الوصول إلى محتويات الدراسة من أي جهاز متصل بالإنترنت (كمبيوتر، جهاز لوحي، هاتف ذكي) بفضل كون TECH على اطلاع بأحدث التطورات التكنولوجية والتربوية.

"التعلم من خبير"ستتمكن من التعلم مع مزايا الوصول إلى بيئات تعليمية محاكاة ونهج التعلم بالملاحظة، أي "التعلم من خبير".



وهكذا، ستكون أفضل المواد التعليمية، المُعدّة بعناية فائقة، متاحة في هذا البرنامج:



المواد الدراسية

يتم خلق جميع محتويات التدريس من قبل المتخصصين الذين سيقومون بتدريس البرنامج الجامعي، وتحديداً من أجله، بحيث يكون التطوير التعليمي محددًا وملموسًا حقًا.

يتم بعد ذلك تطبيق هذه المحتويات على التنسيق السمعي البصري الذي سيخلق طريقتنا في العمل عبر الإنترنت، مع التقنيات الأكثر ابتكارًا التي تتيح لنا أن نقدم لك جودة عالية، في كل قطعة سنضعها في خدمتك.

التدريب العملي على المهارات والكفاءات

ستنفذ أنشطة لتطوير كفاءات ومهارات محددة في كل مجال من مجالات المواد الدراسية. التدريب العملي والديناميكيات لاكتساب وتطوير المهارات والقدرات التي يحتاجها المتخصص لنموه في إطار العولمة التي نعيشها.

ملخصات تفاعلية

نقدم المحتويات بطريقة جذابة وديناميكية في أقراص الوسائط المتعددة التي تشمل الملفات الصوتية والفيديوهات والصور والرسوم البيانية والخرائط المفاهيمية من أجل تعزيز المعرفة..

اعترفت شركة مايكروسوف بهذا النظام التعليمي الفريد من نوعه لتقديم محتوى الوسائط المتعددة على أنه"قصة نجاح أوروبية".

قراءات تكميلية

المقالات الحديثة والوثائق التوافقية والمبادئ التوجيهية الدولية... في مكتبة TECH الافتراضية، سيكون لديك وصول إلى كل ما تحتاجه لإكمال تدريبك.

دراسات الحالة (Case studies)

ستكمل مجموعة مختارة من أفضل دراسات الحالة في المادة التي يتم توظيفها.
حالات تم عرضها وتحليلها وتدريسها من قبل أفضل المتخصصين على الساحة الدولية.

الاختبار وإعادة الاختبار

نقوم بتقييم وإعادة تقييم معرفتك بشكل دوري طوال فترة البرنامج. نقوم بذلك على 3 من 4 مستويات من هرم ميلر.

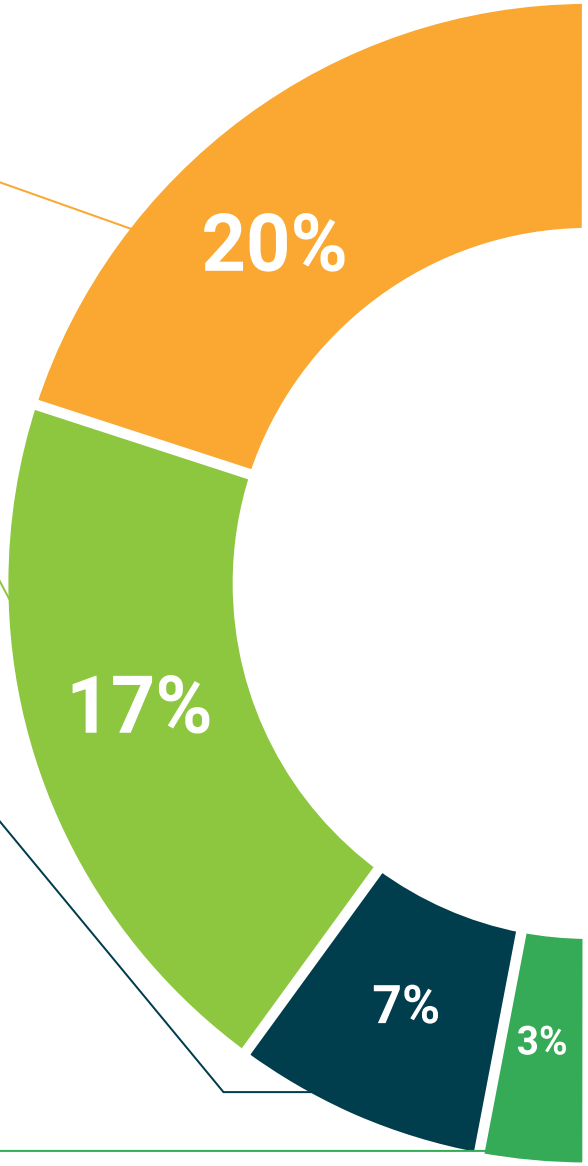
المحاضرات الرئيسية

هناك أدلة علمية على فائدة المراقبة بواسطة الخبراء كطرف ثالث في عملية التعلم.

إن ما يسمى **التعلم من خبير** يقوي المعرفة والذاكرة ، ويولد الأمان في قراراتنا الصعبة في المستقبل.

إرشادات توجيهية سريعة للعمل

تقدم TECH المحتويات الأكثر صلة بالدورة التدريبية في شكل أوراق عمل أو إرشادات توجيهية سريعة للعمل. إنها طريقة موجزة وعملية وفعالة لمساعدة الطلاب على التقدم في تعلمهم.



07

المؤهل العلمي

يضمن الماجستير الخاص في الجودة في تطوير البرمجيات (Software) بالإضافة إلى التدريب الأكثر دقة وحداثة، الحصول على مؤهل الماجستير الخاص الصادر عن TECH الجامعة التكنولوجية.

اجتز هذا البرنامج بنجاح واحصل على مؤهلك العلمي الجامعى
دون الحاجة إلى السفر أو القيام بأية إجراءات مرهقة"

تحتوي درجة **الماجستير الخاص في الجودة في تطوير البرمجيات (Software)** على البرنامج العلمي الأكثر اكتمالا وحداثة في السوق.

بعد اجتياز التقييم، سيحصل الطالب عن طريق البريد العادي* مصحوب بعلم وصول مؤهل **الماجستير الخاص** الصادرعن **TECH الجامعة التكنولوجية.**

إن المؤهل الصادرعن **TECH الجامعة التكنولوجية** سوف يشير إلى التقدير الذي تم الحصول عليه في برنامج الماجستير الخاص وسوف يفي بالمتطلبات التي عادة ما تُطلب من قبل مكاتب التوظيف ومسابقات التعيين ولجان التقييم الوظيفي والمهني.

المؤهل العلمي: **ماجيستير خاص في الجودة في تطوير البرمجيات (Software)**

طريقة الدراسة: **عبر الإنترنت**

مدة الدراسة: **12 شهر**

tech الجامعة التكنولوجية

شهادة تخرج

هذه الشهادة ممنوحة إلى

لـ

المواطن/المواطنة ..... مع وثيقة تحقيق شخصية رقم .....

لاجتيازه/لاجتيازها بنجاح والحصول على برنامج

ماجستير خاص

في

الجودة في تطوير البرمجيات (Software)

وهي شهادة خاصة من هذه الجامعة موافقة لـ 1500
ساعة، مع تاريخ بدء يوم/شهر/ سنة وتاريخ انتهاء يوم/شهر/سنة

تيك مؤسسة خاصة للتعليم العالي معتمدة من وزارة التعليم العام منذ 28 يونيو 2018

في تاريخ 17 يونيو 2020

أ. د./ Tere Guevara Navarro
رئيس الجامعة

الكود الفريد الخاص بجامعة TECH: AFWOR23S techtitute.com/certificates

ماجستير خاص في الجودة في تطوير البرمجيات (Software)

التوزيع العام للخطة الدراسية

نوع المادة	عدد الساعات
(OB) إجباري	1500
(OP) إختياري	0
(PR) الممارسات الخارجية	0
(TFM) مشروع تخرج الماجيستير	0
الإجمالي	1500

التوزيع العام للخطة الدراسية

الدورة	المادة	الطريقة	عدد الساعات
1°	الجودة في تطوير البرمجيات (Software). مستويات مستوى التطور TRL	إجباري	150
1°	تطوير مشاريع البرمجيات. الوثائق الوظيفية والتقنية	إجباري	150
1°	Testing لبرمجيات Software. أتمتة الاختبارات	إجباري	150
1°	منهجيات إدارة مشاريع البرمجيات Software. المنهجيات Waterfall مقابل المنهجيات الرشيقة	إجباري	150
1°	TDD Test Driven Development. تصميم البرمجيات Software المدفوعة بالاختبار	إجباري	150
1°	DevOps. إدارة الجودة في تطوير البرمجيات (Software)	إجباري	150
1°	DevOps والتكامل المستمر. الحلول العملية المتقدمة في تطوير البرمجيات Software	إجباري	150
1°	تصميم قاعدة البيانات. التوحيد والأداء. المقاييس الجودة في تطوير البرمجيات (Software)	إجباري	150
1°	تصميم البنى القابلة للتطوير البنية في دورة حياة البرمجيات Software	إجباري	150
1°	Software. ISO/IEC 9126 ISO/IEC 9126. مقاييس الجودة في تطوير البرمجيات	إجباري	150

tech الجامعة التكنولوجية

أ. د./ Tere Guevara Navarro
رئيس الجامعة

*تصديق لاهاي أبوستيل. في حالة قيام الطالب بالتقدم للحصول على درجته العلمية الورقية وبتصديق لاهاي أبوستيل، ستتخذ مؤسسة TECH EDUCATION الإجراءات المناسبة لكي يحصل عليها وذلك بتكلفة إضافية.

الجامعة التيكنولوجية

ماجيستير خاص
الجودة في تطوير البرمجيات (Software)

- طريقة التدريس: **عبر الإنترنت**
- مدة الدراسة: **12 شهر**
- المؤهل الجامعي من: **TECH الجامعة التكنولوجية**
- مواعيد الدراسة: **وفقًا لوتيرتك الخاصّة**
- الامتحانات: **عبر الإنترنت**

ماجيستير خاص
الجودة في تطوير البرمجيات (Software)

tech global university