22.6 يمكن استخدام عقود السعر الثابت ، حيث يقدم المقاول سعراً ثابتاً لإكمال تطوير النظام ، لنقل مخاطر المشروعمن العميل إلى المتعاقد. إذا حدث خطأ ما ، يجب على المقاول أن يدفع. اقترح كيف يمكن أن يؤدي استخدامثل هذه العقود إلى زيادة احتمالية ظهور مخاطر المنتج.

22.7 اشرح لماذا يمكن لإبقاء جميع أعضاء المجموعة على اطلاع بشأن التقدم والقرارات الفنية في المشروع أن يحسنتماسك المجموعة.

22.8 ما هي صفات أعضاء المجموعة المتماسكة التي تجعل المجموعة قوية؟ اذكر الفوائد الرئيسية لإنشاءمجموعة متماسكة.

22.9 اكتب دراسة حالة بالأسلوب المستخدم هنا لتوضيح أهمية الاتصالات في فريق المشروع. افترض أن بعض أعضاءالفريق يعملون عن بعد وأنه من غير الممكن جمع الفريق بأكمله في وقت قصير.

22.10 يطلب منك مديرك تسليم البرامج وفقاً لجدول زمني تعرف أنه لا يمكن الوفاء به إلا من خلال مطالبة فريق المشروعبالعمل لوقت إضافي بدون أجر. جميع أعضاء الفريق لديهم أطفال صغار. ناقش ما إذا كان يجب عليكقبول هذا الطلب من مديرك أو ما إذا كان عليك إقناع فريقك بإعطاء وقتهم للمؤسسة بدلاً من أسرهم.ما هي العوامل التي قد تكون مهمة في قرارك؟

المراجع

باس ،بي إم ، وجي دونتمان. 1963. "السلوك في المجموعات كدالة للذات والتفاعل والتوجه نحو المهام.".*غير طبيعي.شركة علم النفس .ا*66 (4): 28-19. دوى: 10.1037 / h0042764.

".نموذج حلزوني لتطوير البرمجيات وتحسينها" .Boehm، BW 1988*كمبيوتر IEEE* 21 (5): 72-61. دوى: 10.1109 / 2.59.005.

هول ،إي 1998.*إدارة المخاطر: طرق تطوير أنظمة البرمجيات*. القراءة ، ماجستير: أديسون ويسلي.

مارشالوجيه إي ور. هيسلين. 1975. "الفتيان والفتيات معا. التركيب الجنسي وتأثير الكثافة على حجم المجموعة والتماسك".*من الشخصية وعلم النفس الاجتماعي .ا*35 (5): 961-952. دوى: 10.1037 / ساعة0076838.

ماسلو ،AA 1954.*الدافع والشخصية*. نيويورك: هاربر ورو.

ولد ،م. 1999.*إدارة جودة البرمجيات ومخاطر الأعمال*. تشيتشيستر ، المملكة المتحدة: جون وايلي وأولاده.

تخطيطoject

أهداف

الهدفمن هذا الفصل هو تقديم تخطيط المشروع والجدولة وتقدير التكلفة. عندما تقرأ الفصل ،سوف:

- فهمأساسيات حساب تكلفة البرامج والعوامل التي تؤثر على سعر نظام البرنامج الذيسيتم تطويره للعملاء الخارجيين ؛
- معرفةالأقسام التي يجب تضمينها في خطة المشروع التي يتم إنشاؤها ضمنعملية تطوير مدفوعة بالخطة ؛
- فهمما الذي يتضمنه جدولة المشروع واستخدام المخططات الشريطية لتقديم الجدولالزمني للمشروع ؛
- تمتقديمهم إلى تخطيط المشروع السريع بناءً على "لعبة التخطيط" ؛
- فهمتقنيات تقدير التكلفة وكيف يمكن استخدام نموذج COCOMO II لتقديرتكلفة البرمجيات.

محتويات

يعدتخطيط المشروع من أهم وظائف مدير مشروع البرمجيات. كمدير ، عليك تقسيم العمل إلىأجزاء وتعيينها لأعضاء فريق المشروع ، وتوقع المشاكل التي قد تنشأ ، وإعداد حلول مؤقتة لتلكالمشاكل. تسُتخدم خطة المشروع ، التي يتم إنشاؤها في بداية المشروع ويتم تحديثها مع تقدمالمشروع ، لإظهار كيفية إنجاز العمل وتقييم التقدم المحرز في المشروع.

يتمتخطيط المشروع على ثلاث مراحل في دورة حياة المشروع:

1. في مرحلة الاقتراح ، عندما تقدم عطاءً للحصول على عقد لتطوير أو توفير نظام برمجيات. أنتبحاجة إلى خطة في هذه المرحلة لمساعدتك على تحديد ما إذا كان لديك الموارد لإكمالالعمل ولتحديد السعر الذي يجب أن تقدمه للعميل.

2. أثناء مرحلة بدء المشروع ، عندما يتعين عليك التخطيط لمن سيعمل في المشروع ، وكيف سيتمتقسيم المشروع إلى زيادات ، وكيف سيتم تخصيص الموارد عبر شركتك ، وما إلى ذلك.هنا ، لديك معلومات أكثر مما كانت عليه في مرحلة الاقتراح ، وبالتالي يمكنك تحسينتقديرات الجهود الأولية التي أعددتها.

3. بشكل دوري خلال المشروع ، عند تحديث خطتك لتعكس معلومات جديدة حول البرنامج وتطويره.تتعلم المزيد عن النظام الذي يتم تنفيذه وقدرات فريق التطوير الخاص بك. مع تغيرمتطلبات البرامج ، يجب تعديل توزيع العمل وتمديد الجدول الزمني. تسمح لك هذه المعلوماتبعمل تقديرات أكثر دقة للمدة التي سيستغرقها العمل.

التخطيطفي مرحلة الاقتراح هو تخمين لا محالة ، حيث ليس لديك مجموعة كاملة من المتطلباتللبرنامج المطلوب تطويره. يجب عليك الرد على دعوة لتقديم مقترحات بناءً على وصفعالي المستوى لوظيفة البرنامج المطلوبة. غالباً ما تكون الخطة جزءاً مطلوباً من الاقتراح ،لذلك عليك وضع خطة موثوقة لتنفيذ العمل. إذا فزت بالعقد ، فعليك إعادة تخطيط المشروع ،مع مراعاة التغييرات منذ تقديم الاقتراح والمعلومات الجديدة حول النظام وعملية التطويروفريق التطوير.

عندماتقوم بالمزايدة على عقد ، عليك أن تحدد السعر الذي ستقترحه على العميل لتطوير البرنامج.كنقطة بداية لحساب هذا السعر ، تحتاج إلى وضع تقدير لتكاليفك لإكمال عمل المشروع.يتضمن التقدير تحديد مقدار الجهد المطلوب لإكمال كل نشاط ، ومن هذه الخطوة ، حسابالتكلفة الإجمالية للأنشطة. يجب عليك دائماً حساب تكاليف البرامج بشكل موضوعي ،بهدف التنبؤ الدقيق بتكلفة تطوير البرنامج. بمجرد أن يكون لديك تقدير معقول للتكاليفالمحتملة ، فأنت في وضع يسمح لك بحساب السعر الذي ستعرضه على العميل. كماأناقش في القسم التالي ، هناك العديد من العوامل التي تؤثر على تسعير مشروع البرمجيات- إنها ليست مجرد تكلفة زائد ربح.

التكاليفالعامة

عندماتقدر تكاليف الجهد المبذول في مشروع برمجي ، فأنت لا تضاعف ببساطة رواتب الأشخاص المعنيين بالوقت الذي يقضيه في المشروع.يجب أن تأخذ في الاعتبار جميع النفقات العامة التنظيمية (مساحة المكتب ، الإدارة ، إلخ) التي يجب أن يغطيها الدخل من المشروع.يمكنك حساب التكاليف عن طريق حساب هذه النفقات العامة وإضافة نسبة إلى تكاليف كل مهندس يعمل في مشروع.

http://software-engineering-book.com/web/overhead-costs/

يجبعليك استخدام ثلاث معلمات رئيسية عند حساب تكاليف مشروع تطوير البرمجيات:

- تكاليفالجهد (تكاليف الدفع لمهندسي ومديري البرمجيات) ؛
- تكاليفالأجهزة والبرامج ، بما في ذلك صيانة الأجهزة ودعم البرامج ؛ و
- تكاليفالسفر والتدريب.

بالنسبةلمعظم المشاريع ، تكون التكلفة الأكبر هي تكلفة الجهد. يجب عليك تقدير الجهد الإجمالي(في شهر واحد) الذي من المحتمل أن يكون مطلوباً لإكمال عمل المشروع. من الواضحأن لديك معلومات محدودة لإجراء مثل هذا التقدير. لذلك ، فأنت تقوم بأفضل تقدير ممكنثم تضيف احتياطياً (وقتاً وجهداً إضافيين) في حال كان تقديرك الأولي متفائلاً.

بالنسبةللأنظمة التجارية ، عادةً ما تستخدم أجهزة سلعية ، وهي رخيصة نسبياً. ومع ذلك ، يمكنأن تكون تكاليف البرامج كبيرة إذا كان عليك ترخيص البرامج الوسيطة وبرامج النظام الأساسي.قد تكون هناك حاجة إلى سفر مكثف عند تطوير مشروع في مواقع مختلفة. في حين أنتكاليف السفر نفسها عادة ما تكون جزءاً صغيراً من تكاليف الجهد ، فإن الوقت الذي يقضيهالسفر غالباً ما يضيع ويضيف بشكل كبير إلى تكاليف جهود المشروع. يمكنك استخدام أنظمةالاجتماعات الإلكترونية والبرامج التعاونية الأخرى لتقليل السفر وبالتالي إتاحة المزيد من الوقتللعمل المنتج.

بمجردمنح عقد تطوير نظام ، يجب تنقيح مخطط المشروع التفصيلي للمشروع لإنشاء خطةبدء تشغيل المشروع. في هذه المرحلة ، يجب أن تعرف المزيد عن متطلبات هذا النظام. يجبأن يكون هدفك هو إنشاء خطة مشروع بتفاصيل كافية للمساعدة في اتخاذ قرارات بشأنالتوظيف والميزانية للمشروع. يمكنك استخدام هذه الخطة كأساس لتخصيص الموارد للمشروعمن داخل المنظمة وللمساعدة في تحديد ما إذا كنت بحاجة إلى تعيين موظفين جدد.

يجبأن تحدد الخطة أيضاً آليات مراقبة المشروع. يجب عليك تتبع التقدم المحرز في المشروعومقارنة التقدم والتكاليف الفعلية والمخطط لها. على الرغم من أن معظم الشركات لديهاإجراءات رسمية للمراقبة ، يجب أن يكون المدير الجيد قادراً على تكوين صورة واضحة لمايجري من خلال المناقشات غير الرسمية مع موظفي المشروع. يمكن للرصد غير الرسمي التنبؤبمشاكل المشروع المحتملة من خلال الكشف عن الصعوبات عند حدوثها. على سبيل المثال ،المناقشات اليومية مع المشروع

قديكشف فريق العمل أن الفريق يواجه مشاكل في خلل برمجي في أنظمة الاتصالات. يمكن لمديرالمشروع بعد ذلك تعيين خبير اتصالات للمشكلة على الفور للمساعدة في العثور على المشكلةوحلها.

تتطورخطة المشروع دائماً أثناء عملية التطوير بسبب تغييرات المتطلبات ، وقضايا التكنولوجيا ،ومشاكل التنمية. يهدف تخطيط التطوير إلى ضمان أن تظل خطة المشروع وثيقة مفيدةللموظفين لفهم ما يجب تحقيقه ومتى يتم تسليمه. لذلك ، يجب مراجعة الجدول الزمنيوتقدير التكلفة والمخاطر مع تطوير البرنامج.

إذاتم استخدام طريقة Agile ، فلا تزال هناك حاجة إلى خطة بدء المشروع لأنه بغض النظر عنالنهج المستخدم ، لا تزال الشركة بحاجة إلى التخطيط لكيفية تخصيص الموارد للمشروع. ومعذلك ، فهذه ليست خطة مفصلة ، وتحتاج فقط إلى تضمين المعلومات الأساسية حول تفاصيلالعمل والجدول الزمني للمشروع. أثناء التطوير ، يتم وضع خطة غير رسمية للمشروع وتقديراتللجهود لكل إصدار من البرنامج ، مع مشاركة الفريق بأكمله في عملية التخطيط. تمتبالفعل تغطية بعض جوانب التخطيط السريع في الفصل 3 ، وأنا أناقش المناهج الأخرى في القسم 23.4.

23.1 تسعير البرمجيات

منحيث المبدأ ، فإن سعر نظام برمجي تم تطويره للعميل هو ببساطة تكلفة التطوير بالإضافة إلىربح المطور. ومع ذلك ، من الناحية العملية ، فإن العلاقة بين تكلفة المشروع والسعر المعروضللعميل ليست بهذه البساطة في العادة. عند حساب السعر ، تأخذ في الاعتبار اعتباراتتنظيمية واقتصادية وسياسية وتجارية أوسع (الشكل 23.1). تحتاج إلى التفكير في المخاوفالتنظيمية ، والمخاطر المرتبطة بالمشروع ، ونوع العقد الذي سيتم استخدامه. قد تتسببهذه المشكلات في تعديل السعر لأعلى أو لأسفل.

لتوضيحبعض مشكلات تسعير المشروع ، ضع في اعتبارك السيناريو التالي:

توظفشركة برمجيات صغيرة ، PharmaSoft ، 10 مهندسي برمجيات. لقد أنهت للتومشروعاً كبيراً ولكن ليس لديها سوى عقود سارية تتطلب خمسة من موظفي التطوير.ومع ذلك ، فهي تقدم عطاءً لعقد كبير جداً مع شركة أدوية كبرى يتطلب جهداً لمدة30 عاماً على مدار عامين. لن يبدأ المشروع لمدة 12 شهراً على الأقل ، ولكن في حالةمنحه ، سيغير الوضع المالي للشركة.

تحصلPharmaSoft على فرصة لتقديم عطاءات على مشروع يتطلب ستة أشخاص ويجبأن يكتمل في غضون 10 أشهر. تقدر التكاليف (بما في ذلك النفقات العامة لهذا المشروع) بمبلغ 1.2 مليون دولار. ومع ذلك ، من أجل تحسين مركزها التنافسي ، قررتPharmaSoft تقديم عرض سعر للعميل قدره 0.8 مليون دولار. وهذا يعني أنه على الرغم من خسارة الأموال في هذا العقد ، إلا أنه يمكنها الاحتفاظ بالموظفين المتخصصينللمشاريع المستقبلية الأكثر ربحية والتي من المحتمل أن تبدأ العمل في غضون عام.

عامل	وصف
الشروطالتعاقدية	قديكون العميل على استعداد للسماح للمطور بالاحتفاظ بملكية كود المصدر وإعادة استخدامهفي مشاريع أخرى. قد يتم تخفيض السعر الذي يتم تحصيله بعد ذلك ليعكس قيمةكود المصدر للمطور.
عدماليقين في تقدير التكلفة	إذاكانت المنظمة غير متأكدة من تقدير التكلفة الخاص بها ، فقد تزيد سعرها باحتياطي يزيدعن ربحها العادي.
القدرةالمادية	الشركاتالتي تعاني من مشاكل مالية قد تخفض أسعارها للحصول على عقد. من الأفضلتحقيق ربح أو كسر أقل من المعتاد بدلاً من الخروج من العمل. التدفق النقدي أكثرأهمية من الربح في الأوقات الاقتصادية الصعبة.
فرصةالسوق	قدتقتبس منظمة التطوير سعراً منخفضاً لأنها ترغب في الانتقال إلى قطاع جديد من سوق البرمجيات.قد يؤدي قبول ربح منخفض في مشروع واحد إلى منح المنظمة الفرصة لتحقيقربح أكبر لاحقاً. قد تساعد الخبرة المكتسبة أيضاً في تطوير منتجات جديدة.
تقلبالمتطلبات	إذاكان من المحتمل أن تتغير المتطلبات ، فقد تخفض المنظمة سعرها للفوز بعقد. بعد منحالعقد ، يمكن فرض أسعار عالية للتغييرات في المتطلبات.

الشكل23.1عوامل تؤثرعلى البرمجيات التسعير

هذامثال على نهج لتسعير البرامج يسمى "التسعير للفوز". التسعير للفوز يعني أن الشركة لديهابعض*فكرة*من السعر الذي يتوقع العميل دفعه ويقدم عطاءً للعقد بناءً على السعر المتوقع للعميل.قد يبدو هذا غير أخلاقي وغير تجاري ، لكن له مزايا لكل من العميل ومزود النظام.

يتمالاتفاق على تكلفة المشروع على أساس اقتراح مخطط تفصيلي. ثم تجري المفاوضات بينالعميل والعميل لتحديد المواصفات التفصيلية للمشروع. هذه المواصفات مقيدة بالتكلفةالمتفق عليها. يجب أن يتفق البائع والمشتري على وظيفة النظام المقبولة. العامل الثابتفي العديد من المشاريع ليس متطلبات المشروع ولكن التكلفة. يمكن تغيير المتطلبات بحيثتظل تكاليف المشروع في حدود الميزانية.

علىسبيل المثال ، لنفترض أن شركة (OilSoft) تقدم عطاءات للحصول على عقد لتطوير نظامتوصيل الوقود لشركة نفط تقوم بجدولة شحنات الوقود إلى محطات الخدمة الخاصة بها. لاتوجد وثيقة متطلبات تفصيلية لهذا النظام ، لذلك تقدر OilSoft أن سعراً قدره 900 ألف دولارمن المرجح أن يكون تنافسياً وضمن ميزانية شركة النفط. بعد منح العقد ، تقوم OilSoft بعدذلك بالتفاوض بشأن المتطلبات التفصيلية للنظام بحيث يتم تسليم الوظائف الأساسية. ثمتقوم بتقدير التكاليف الإضافية للمتطلبات الأخرى.

هذاالنهج له مزايا لكل من مطور البرامج والعميل. يتم التفاوض على المتطلبات لتجنب المتطلباتالتي يصعب تنفيذها والتي من المحتمل أن تكون باهظة الثمن. تسهل المتطلبات المرنةإعادة استخدام البرامج. منحت شركة النفط العقد إلى شركة معروفة يمكنها الوثوق بها. علاوةعلى ذلك ، قد يكون من الممكن توزيع تكلفة

المشروععلى عدة إصدارات من النظام. قد يقلل هذا من تكاليف نشر النظام ويسمح للعميل بوضعميزانية لتكلفة المشروع على مدى عدة سنوات مالية.

23.2 تنمية المدفوعة بالخطة

التطويرالقائم على الخطة أو المستند إلى الخطة هو نهج لهندسة البرمجيات حيث يتم التخطيطلعملية التطوير بالتفصيل. يتم إنشاء خطة مشروع تسجل العمل الذي يتعين القيام به ،ومن سيقوم بذلك ، والجدول الزمني للتطوير ، ومنتجات العمل. يستخدم المديرون الخطة لدعماتخاذ قرارات المشروع وكوسيلة لقياس التقدم. يعتمد التطوير المدفوع بالخطة على تقنياتإدارة المشاريع الهندسية ويمكن اعتباره الطريقة "التقليدية" لإدارة مشاريع تطوير البرامجالكبيرة. يتضمن التطوير السريع عملية تخطيط مختلفة ، تمت مناقشتها في القسم 23.4 ،حيث يتم تأخير القرارات.

تكمنمشكلة التطوير المبني على الخطة في أنه يجب مراجعة القرارات المبكرة بسبب التغييراتفي البيئات التي يتم فيها تطوير البرنامج واستخدامه. يؤدي تأخير قرارات التخطيط إلىتجنب إعادة العمل غير الضروري. ومع ذلك ، فإن الحجج المؤيدة للنهج القائم على الخطة هيأن التخطيط المبكر يسمح بأخذ القضايا التنظيمية (توافر الموظفين ، والمشاريع الأخرى ، وماإلى ذلك) في الاعتبار. يتم اكتشاف المشكلات والتبعيات المحتملة قبل بدء المشروع ، وليسبمجرد بدء المشروع.

فيرأيي ، فإن أفضل نهج لتخطيط المشروع ينطوي على مزيج معقول من التنمية المستندةإلى الخطة والمرونة. يعتمد التوازن على نوع المشروع ومهارات الأشخاص المتاحين. فيأحد الأطراف ، تتطلب أنظمة الأمان والحالات الحرجة الكبيرة تحليلاً أولياً شاملاً وقد يتعين اعتمادهاقبل استخدامها. يجب أن تكون هذه الأنظمة في الغالب مدفوعة بالخطة. على الجانبالآخر ، يجب أن تكون أنظمة المعلومات الصغيرة إلى المتوسطة الحجم ، لاستخدامها فيبيئة تنافسية سريعة التغير ، مرنة في الغالب. عندما تشارك عدة شركات في مشروع تطوير ،عادة ما يتم استخدام نهج مبني على الخطة لتنسيق العمل عبر كل موقع تطوير.

23.2.1 خطط المشروع

فيمشروع تطوير مبني على الخطة ، تحدد خطة المشروع الموارد المتاحة للمشروع وتفصيل العملوالجدول الزمني لتنفيذ العمل. يجب أن تحدد الخطة النهج الذي يتم اتباعه لإدارة المخاطروكذلك المخاطر التي يتعرض لها المشروع والبرمجيات قيد التطوير. تختلف تفاصيل خططالمشروع اعتماداً على نوع المشروع والتنظيم ، لكن الخطط تتضمن عادةً الأقسام التالية:

1. *مقدمة* يصف بإيجاز أهداف المشروع ويحدد القيود (مثل الميزانية والوقت) التي تؤثر على إدارةالمشروع.
2. *منظمة المشروع* يصف الطريقة التي يتم بها تنظيم فريق التطوير والأشخاص المشاركين وأدوارهمفي الفريق.

يخطط	وصف
خطةإدارة التكوين	يصفإجراءات إدارة التكوين والهياكل التي سيتم استخدامها.
خطةالنشر	يصفكيفية نشر البرامج والأجهزة المرتبطة بها (إذا لزم الأمر) في بيئة العميل. يجبأن يتضمن ذلك خطة لترحيل البيانات من الأنظمة الحالية.
خطةالصيانة	يتنبأبمتطلبات الصيانة والتكاليف والجهد.
خطةالجودة	يصفإجراءات ومعايير الجودة التي سيتم استخدامها في المشروع.
خطةالتحقق	يصفالنهج والموارد والجدول الزمني المستخدم للتحقق من صحة النظام.

الشكل23.2مشروع مكملاتالخطة

3. *تحليل المخاطر*يصف مخاطر المشروع المحتملة ، واحتمالية ظهور هذه المخاطر ، واستراتيجياتالحد من المخاطر (التي تمت مناقشتها في الفصل 22) المقترحة.

4. *متطلبات موارد الأجهزة والبرامج*تحدد الأجهزة وبرامج الدعم المطلوبة لتنفيذ التطوير. إذا كانلابد من شراء الأجهزة ، فقد يتم تضمين تقديرات الأسعار وجدول التسليم.

5. *انهيار العمل*يحدد تقسيم المشروع إلى أنشطة ويحدد المدخلات والمخرجات من كل نشاط منأنشطة المشروع.

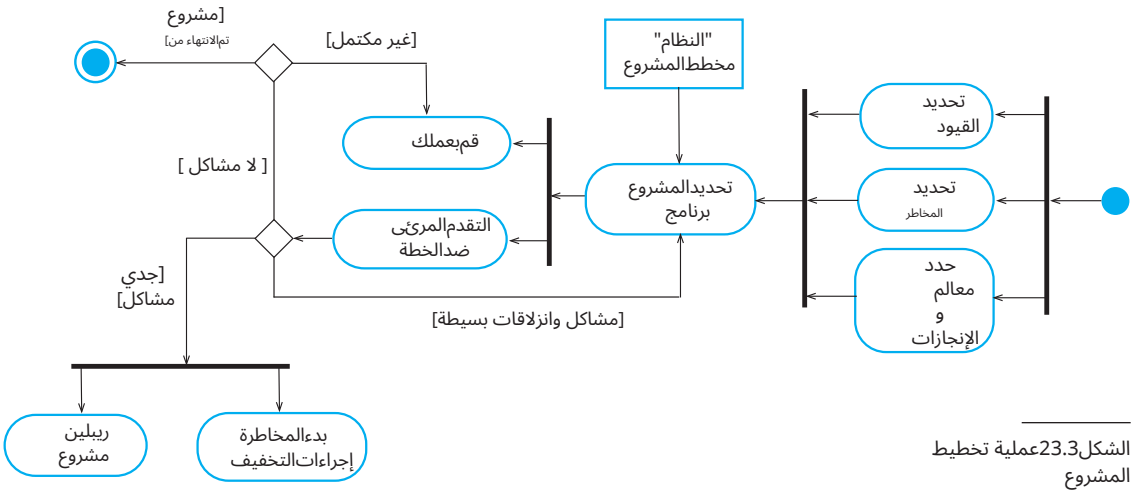
6. *الجدول الزمنى للمشروع*إظهار التبعيات بين الأنشطة والوقت المقدر المطلوب للوصول إلىكل حدث رئيسي وتخصيص الأشخاص للأنشطة. تتم مناقشة الطرق التي يمكن من خلالهاتقديم الجدول الزمني في القسم التالي من الفصل.

7. *آليات الرصد والإبلاغ*يحدد تقارير الإدارة التي يجب إنتاجها ، ومتى يجب إنتاجها ، وآليات مراقبةالمشروع التي سيتم استخدامها.

يجبأن تتضمن خطة المشروع الرئيسية دائماً تقييماً لمخاطر المشروع وجدولاً زمنياً للمشروع.بالإضافة إلى ذلك ، يمكنك تطوير عدد من الخطط التكميلية لأنشطة مثل الاختبار وإدارةالتكوين. يوضح الشكل 23.2 بعض الخطط التكميلية التي يمكن تطويرها. هذه كلها مطلوبةعادة في المشاريع الكبيرة التي تعمل على تطوير أنظمة كبيرة ومعقدة.

23.2.2عملية التخطيط

تخطيطالمشروع هو عملية تكرارية تبدأ عند إنشاء خطة مشروع أولية أثناء مرحلة بدء المشروع. الشكل23.3 هو مخطط نشاط UML يوضح سير عمل نموذجي لعملية تخطيط المشروع. تغييراتالخطة أمر لا مفر منه. مع توفر المزيد من المعلومات حول النظام وفريق المشروع



الشكل23.3عملية تخطيط المشروع

أثناءالمشروع ، يجب عليك مراجعة الخطة بانتظام لتعكس المتطلبات والجدول الزمني وتغييراتالمخاطر. يؤدي تغيير أهداف العمل أيضاً إلى تغييرات في خطط المشروع. مع تغير أهدافالعمل ، يمكن أن يؤثر ذلك على جميع المشاريع ، والتي قد يتعين إعادة التخطيط لها بعدذلك.

فيبداية عملية التخطيط ، يجب عليك تقييم القيود التي تؤثر على المشروع. هذه القيود هيتاريخ التسليم المطلوب ، والموظفين المتاحين ، والميزانية الإجمالية ، والأدوات المتاحة ، ومإلى ذلك. بالتزامن مع هذا التقييم ، يجب عليك أيضاً تحديد معالم المشروع والتسليمات. المعالمالرئيسية هي نقاط في الجدول يمكنك على أساسها تقييم التقدم ، على سبيل المثال ،تسليم النظام للاختبار. التسليمات هي منتجات العمل التي يتم تسليمها إلى العميل ، علىسبيل المثال ، مستند المتطلبات للنظام.

ثمتدخل العملية حلقة تنتهي عند اكتمال المشروع. تقوم بوضع جدول تقديري للمشروع ، ويتمبدء الأنشطة المحددة في الجدول أو الموافقة عليها للمتابعة. بعد مرور بعض الوقت (عادةً حوالي أسبوعين إلى ثلاثة أسابيع) ، يجب عليك مراجعة التقدم وملاحظة التناقضات في الجدولالزمني المخطط. نظراً لأن التقديرات الأولية لمعلمات المشروع تقريبية حتماً ، فإن الانزلاقاتالطفيفة أمر طبيعي وسيتعين عليك إجراء تعديلات على الخطة الأصلية.

يجبأن تضع افتراضات واقعية وليست متفائلة عندما تحدد خطة مشروع. تظهر مشاكل بعضالوصف دائماً أثناء المشروع ، وتؤدي إلى تأخير المشروع. لذلك يجب أن تكون افتراضاتكوجدولتك الأولية متشائمة وأن تأخذ في الاعتبار المشكلات غير المتوقعة. يجب عليكتضمين الطوارئ في خطتك بحيث إذا ساءت الأمور ، فلن يتم تعطيل جدول التسليم بشكلخطير.

إذاكانت هناك مشاكل خطيرة في أعمال التطوير من المحتمل أن تؤدي إلى تأخيرات كبيرة ،فأنت بحاجة إلى بدء إجراءات تخفيف المخاطر لتقليل مخاطر فشل المشروع. بالتزامن معهذه الإجراءات ، عليك أيضاً إعادة تخطيط المشروع. قد يتضمن ذلك إعادة التفاوض بشأن قيودالمشروع والتسليمات مع العميل. يجب أيضاً وضع جدول زمني جديد لموعد الانتهاء من العملوالاتفاق عليه مع العميل.

إذالم تنجح عملية إعادة التفاوض هذه أو كانت إجراءات تخفيف المخاطر غير فعالة ، فيجب عليك الترتيب لإجراء مراجعة فنية رسمية للمشروع. تهدف هذه المراجعة إلى إيجاد نهج بديل يسمح للمشروع بالاستمرار. يجب أن تتحقق المراجعات أيضاً من أن أهداف العميل لم تتغير وأن المشروع يظل متوافقاً مع هذه الأهداف.

قد تكون نتيجة المراجعة قراراً بإلغاء المشروع. قد يكون هذا نتيجة لفشل فني أو إداري ولكن في كثير من الأحيان يكون نتيجة للتغييرات الخارجية التي تؤثر على المشروع. غالباً ما يكون وقت تطوير مشروع برمجيات كبير عدة سنوات. خلال ذلك الوقت ، تتغير أهداف العمل وأولوياته حتما. قد تعني هذه التغييرات أن البرنامج لم يعد مطلوباً أو أن متطلبات المشروع الأصلية غير مناسبة. قد تقرر الإدارة بعد ذلك إيقاف تطوير البرامج أو إجراء تغييرات كبيرة على المشروع لتعكس التغييرات في الأهداف التنظيمية.

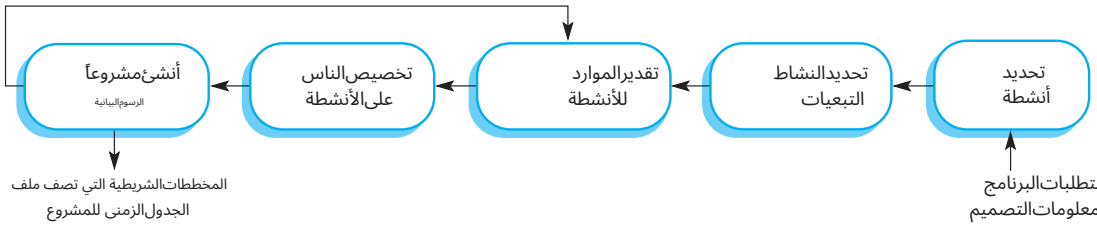
23.3 جدولة المشروع

جدولة المشروع هي عملية تحديد كيفية تنظيم العمل في المشروع كمهام منفصلة ، ومتى وكيف سيتم تنفيذ هذه المهام. أنت تقدر وقت التقويم اللازم لإكمال كل مهمة والجهد المطلوب ، وتقترح من سيعمل على المهام التي تم تحديدها. يجب عليك أيضاً تقدير موارد الأجهزة والبرامج اللازمة لإكمال كل مهمة. على سبيل المثال ، إذا كنت تقوم بتطوير نظام مضمن ، فيجب عليك تقدير الوقت الذي تحتاجه على الأجهزة المتخصصة وتكاليف تشغيل محاكي النظام. فيما يتعلق بمراحل التخطيط التي قدمتها في مقدمة هذا الفصل ، يتم عادةً إنشاء جدول زمني أولي للمشروع أثناء مرحلة بدء المشروع. ثم يتم تنقيح هذا الجدول الزمني وتعديله أثناء التخطيط للتنمية.

تحتاج كل من العمليات المستندة إلى الخطة والعملية الرشيقة إلى جدول زمني أولي للمشروع ، على الرغم من تضمين تفاصيل أقل في خطة مشروع رشيقة. يتم استخدام هذا الجدول الزمني الأولي لتخطيط كيفية تخصيص الأشخاص للمشاريع وللتحقق من تقدم المشروع مقابل التزاماته التعاقدية. في عمليات التطوير التقليدية ، يتم في البداية تطوير الجدول الكامل ثم تعديله مع تقدم المشروع. في العمليات الرشيقة ، يجب أن يكون هناك جدول زمني شامل يحدد متى سيتم الانتهاء من المراحل الرئيسية من المشروع. ثم يتم استخدام نهج تكراري للجدولة للتخطيط لكل مرحلة.

تتضمن الجدولة في المشاريع القائمة على الخطة (الشكل 23.4) تقسيم إجمالي العمل المتضمن في المشروع إلى مهام منفصلة وتقدير الوقت المطلوب لإكمال كل مهمة. يجب أن تستمر المهام عادة لمدة أسبوع على الأقل ولا تزيد عن شهرين. التقسيم الدقيق يعني أنه يجب إنفاق قدر غير متناسب من الوقت في إعادة تخطيط خطة المشروع وتحديثها. يجب أن يكون الحد الأقصى للوقت لأي مهمة من 6 إلى 8 أسابيع. إذا كانت المهمة تستغرق وقتاً أطول من ذلك ، فيجب تقسيمها إلى مهام فرعية لتخطيط المشروع وجدولته.

يتم تنفيذ بعض هذه المهام بالتوازي ، حيث يعمل أشخاص مختلفون على مكونات مختلفة من النظام. يجب عليك تنسيق هذه المهام المتوازية وتنظيم العمل بحيث يتم استخدام القوى العاملة على النحو الأمثل ولا تقدم



الشكل23.4عملية جدولة المشروع

التبعياتغير الضرورية بين المهام. من المهم تجنب الموقف الذي يتأخر فيه المشروع بأكمله لأنالمهمة الحاسمة لم تكتمل.

إذاكان المشروع متقدماً تقنياً ، فمن شبه المؤكد أن التقديرات الأولية ستكون متفائلة حتىعندما تحاول النظر في جميع الاحتمالات. في هذا الصدد ، لا تختلف جدولة البرامج عن جدولةأي نوع آخر من المشاريع الكبيرة المتقدمة. غالباً ما تتأخر الطائرات الجديدة والجسور وحتىالنماذج الجديدة من السيارات بسبب مشاكل غير متوقعة. لذلك ، يجب تحديث الجداولباستمرار كلما توفرت معلومات تقدم أفضل. إذا كان المشروع الذي تتم جدولته مشابهاًلمشروع سابق ، فقد يعُاد استخدام التقديرات السابقة. ومع ذلك ، قد تستخدم المشاريعطرق تصميم ولغات تنفيذ مختلفة ، لذلك قد لا تكون الخبرة المكتسبة من المشاريع السابقةقابلة للتطبيق في التخطيط لمشروع جديد.

عندماتقوم بتقدير الجداول ، يجب أن تأخذ في الاعتبار احتمالية أن تسوء الأمور. قد يمرض الأشخاصالذين يعملون في مشروع ما أو يغادرون ، وقد تتعطل الأجهزة ، وقد يتم تسليم برامجأو أجهزة الدعم الأساسية في وقت متأخر. إذا كان المشروع جديداً ومتقدماً تقنياً ، فقد تصبحأجزاء منه أكثر صعوبة وتستغرق وقتاً أطول مما كان متوقعاً في الأصل.

تتمثلإحدى القواعد العامة الجيدة في التقدير كما لو أن شيئاً لن يحدث بشكل خاطئ ، ثم قمبزيادة تقديرك لتغطية المشكلات المتوقعة. يمكن أيضاً إضافة عامل طوارئ آخر لتغطية المشكلاتغير المتوقعة إلى التقدير. يعتمد عامل الطوارئ الإضافي هذا على نوع المشروع ، ومعاييرالعملية (الموعد النهائي ، والمعايير ، وما إلى ذلك) ، وجودة وخبرة مهندسي البرمجياتالعاملين في المشروع. قد تضيف تقديرات الطوارئ من 30 إلى 50٪ إلى الجهد والوقتاللازمين للمشروع.

23.3.1جدول العرض

يمكنببساطة توثيق جداول المشروع في جدول أو جدول بيانات يوضح المهام والجهد المقدر والمدةوتبعيات المهام (الشكل 23.5). ومع ذلك ، فإن أسلوب العرض هذا يجعل من الصعب رؤيةالعلاقات والتبعيات بين الأنشطة المختلفة. لهذا السبب ، تم تطوير تصورات رسومية بديلةلجداول المشروع والتي غالباً ما تكون أسهل في القراءة والفهم. يشيع استخدام نوعين منالتصور:

1.تظُهر المخططات الشريطية المستندة إلى التقويم المسؤول عن كل نشاط ، والوقت المنقضيالمتوقع ، وموعد بدء النشاط وانتهائه. تسمى المخططات الشريطية أيضاً مخططاتجانت ، نسبة لمخترعها ، هنري جانت.

مهمة	الجهد(أيام عمل)	المدة(أيام)	التبعيات
T1	15	10	
T2	8	15	
T3	20	15	M1(T1)
T4	5	10	
T5	5	10	M3(T2، T4)
T6	10	5	M4(T1، T2)
T7	25	20	M1(T1)
T8	75	25	M2(T4)
T9	10	15	M5(T3، T6)
T10	20	15	M6(T7، T8)
T11	10	10	M7(T9)
T12	20	10	M8(T10، T11)

الشكل23.5مهام، المددو التبعيات

2.تظهر شبكات النشاط التبعيات بين الأنشطة المختلفة التي يتكون منها المشروع. يتم وصفهذه الشبكات في قسم الويب المرتبط.

أنشطةالمشروع هي عنصر التخطيط الأساسي. كل نشاط له:

- المدةبالأيام أو الأشهر التقويمية ؛
- تقديرللجهد ، والذي يوضح عدد أيام عمل الشخص أو شهره لإكمال العمل ؛
- الموعدالنهائي الذي يجب أن يكتمل فيه النشاط ؛ و
- نقطةنهاية محددة ، والتي قد تكون مستنداً ، أو عقد اجتماع مراجعة ، أو تنفيذ ناجح لجميع الاختبارات ،أو ما شابه ذلك.

عندالتخطيط لمشروع ما ، قد تقرر تحديد معالم المشروع. المعلم الرئيسي هو نهاية منطقيةلمرحلة من المشروع حيث يمكن مراجعة تقدم العمل. يجب توثيق كل حدث رئيسي بتقريرموجز (غالباً ما يكون مجرد بريد إلكتروني) يلخص العمل المنجز وما إذا كان العمل قد اكتملكما هو مخطط أم لا. قد ترتبط المعالم بمهمة واحدة أو بمجموعات من الأنشطة ذات الصلة.على سبيل المثال ، في الشكل 23.5 ، يرتبط المعلم الرئيسي M1 بالمهمة T1 ويمثل نهايةهذا النشاط. يرتبط Milestone M3 بزوج من المهام T2 و T4 ؛ لا يوجد معلم فردي في نهايةهذه المهام.

مخططاتالنشاط

مخططالنشاط هو تمثيل لجدول المشروع الذي يقدم خطة المشروع كرسم بياني موجه. إنه يوضح المهام التي يمكن تنفيذها بالتوازي وتلك التييجب تنفيذها بالتسلسل نظراً لاعتمادها على الأنشطة السابقة. إذا كانت المهمة تعتمد على عدة مهام أخرى ، فيجب إكمال كل هذه المهامقبل أن تبدأ. "المسار الحرج" عبر مخطط النشاط هو أطول سلسلة من المهام التابعة. هذا يحدد مدة المشروع.

http://software-engineering-book.com/web/planning-activities/

تقومبعض الأنشطة بإنشاء مخرجات المشروع - المخرجات التي يتم تسليمها إلى عميل البرنامج.عادة ، يتم تحديد المخرجات المطلوبة في عقد المشروع ، وتعتمد رؤية العميل لتقدم المشروععلى هذه التسليمات. المعالم والمخرجات ليست هي نفسها. المعالم الرئيسية هي تقاريرقصيرة تسُتخدم للإبلاغ عن التقدم المحرز ، في حين أن النواتج هي مخرجات مشروع أكثر جوهريةمثل وثيقة المتطلبات أو التنفيذ الأولي للنظام.

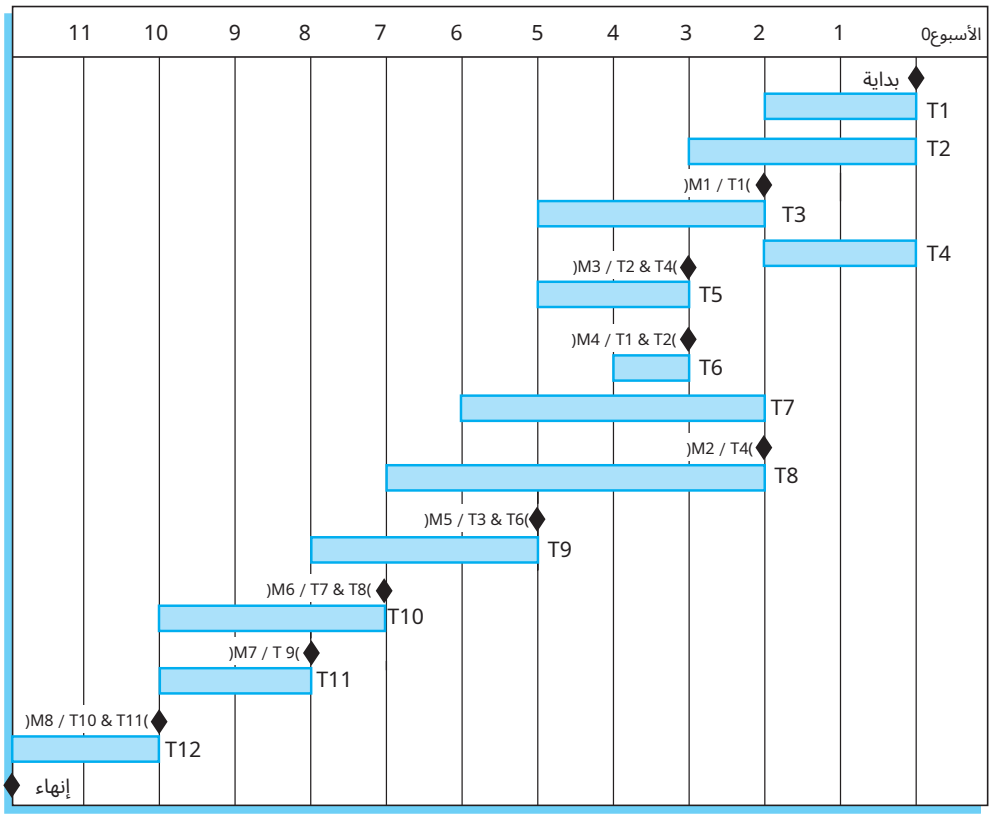
يوضحالشكل 23.5 مجموعة افتراضية من المهام وجهودها المقدرة ومدتها وتبعيات المهام.من هذا الجدول ، يمكنك أن ترى أن المهمة T3 تعتمد على المهمة T1. هذا يعني أن المهمةT1 يجب أن تكتمل قبل أن تبدأ T3. على سبيل المثال ، قد يكون T1 هو اختيار نظام لإعادةالاستخدام و T3 ، تكوين النظام المحدد. لا يمكنك بدء تكوين النظام حتى تختار وتثبيت نظامالتطبيق المراد تعديله.

لاحظأن المدة المقدرة لبعض المهام أكثر من المجهود المطلوب والعكس صحيح. إذا كان الجهدأقل من المدة ، فلن يعمل الأشخاص المخصصون لهذه المهمة بدوام كامل. إذا تجاوز الجهدالمدة ، فهذا يعني أن العديد من أعضاء الفريق يعملون على المهمة في نفس الوقت.

يأخذالشكل 23.6 المعلومات الواردة في الشكل 23.5 ويعرض الجدول الزمني للمشروع كمخططشريطي يعرض تقويم المشروع وتواريخ البدء والانتهاء للمهام. عند القراءة من اليسار إلىاليمين ، يظهر الرسم البياني الشريطي بوضوح متى تبدأ المهام وتنتهي. تظهر المعالم (M2 ،M1 ، إلخ) أيضاً على المخطط الشريطي. لاحظ أنه يمكن تنفيذ المهام المستقلة بشكل متوازٍعلى سبيل المثال ، تبدأ المهام T1 و T2 و T4 في بداية المشروع.

بالإضافةإلى تخطيط جدول التسليم للبرنامج ، يتعين على مديري المشاريع تخصيص الموارددللمهام. المورد الرئيسي ، بالطبع ، هو مهندسو البرمجيات الذين سيقومون بهذا العمل.يجب أن يتم تكليفهم بأنشطة المشروع. يمكن تحليل تخصيص الموارد من خلال أدواتإدارة المشروع ، ويمكن إنشاء مخطط شريطي يوضح وقت عمل الموظفين في المشروع(الشكل 23.7). قد يعمل الأشخاص على أكثر من مهمة في نفس الوقت ، وأحياناً لا يعملونفي المشروع. قد يكونون في إجازة أو يعملون في مشاريع أخرى أو يحضرون دورات تدريبية.أعرض مهام بدوام جزئي باستخدام خط قطري يتقاطع مع الشريط.

عادةما توظف المنظمات الكبيرة عدداً من المتخصصين الذين يعملون في مشروع عند الحاجة.في الشكل 23.7 ، يمكنك أن ترى أن ماري متخصصة في العمل



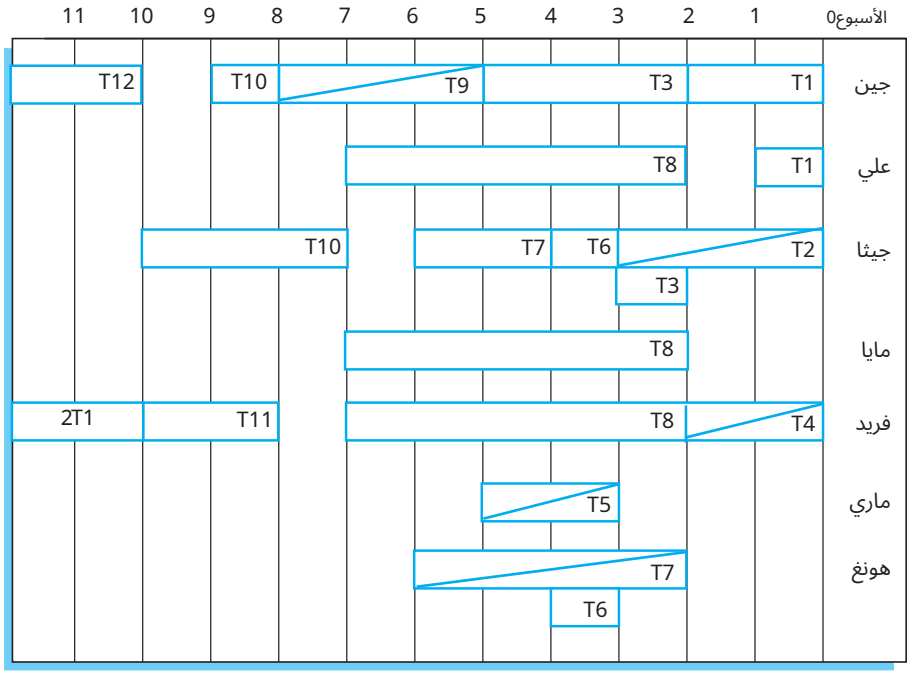
الشكل23.6نشاط

شريطالرسم البياني

فقطمهمة واحدة (T5) في المشروع. لا مفر من الاستعانة بالمتخصصين عند تطوير أنظمة معقدة ،ولكن يمكن أن يؤدي ذلك إلى مشاكل في الجدولة. إذا تأخر أحد المشاريع أثناء عمل أحدالمتخصصين عليه ، فقد يؤثر ذلك على مشاريع أخرى حيث يكون الاختصاصي مطلوباً أيضاً.قد تتأخر هذه المشاريع لأن الاختصاصي غير متوفر.

إذاتم تأخير مهمة ما ، فقد تتأثر المهام اللاحقة التي تعتمد عليها. لا يمكنهم البدء حتى تكتملالمهمة المؤجلة. يمكن أن تتسبب التأخيرات في حدوث مشكلات خطيرة في تخصيصالموظفين ، خاصةً عندما يعمل الأشخاص في عدة مشروعات في نفس الوقت. إذا تأخرتمهمة (T) ، فقد يتم تعيين الأشخاص المخصصين لها لعمل آخر (W). قد يستغرق إكمالهذا العمل وقتاً أطول من التأخير ، ولكن بمجرد تعيينهم ، لا يمكن إعادة تعيينهم مرة أخرىإلى المهمة الأصلية. قد يؤدي هذا بعد ذلك إلى مزيد من التأخير في T لأنها تكمل W.

عادة ،يجب عليك استخدام أداة تخطيط المشروع ، مثل Basecamp أو مشروع Microsoft ، لإنشاء وتحديث وتحليل معلومات جدول المشروع. تتوقع أدوات إدارة المشروع عادةً إدخال معلومات المشروع في جدول ، ويقومون بإنشاء قاعدة بيانات لمعلومات المشروع.يمكن بعد ذلك إنشاء المخططات الشريطية ومخططات النشاط تلقائياً من قاعدة البياناتهذه.



الشكل23.7العاملين مخططالتخصيص

23.4 تخطيط السريع

الأساليبالرشيقة لتطوير البرمجيات هي مناهج تكرارية حيث يتم تطوير البرنامج وتسليمه للعملاءبزيادات. على عكس النهج التي تعتمد على الخطة ، لم يتم التخطيط لوظيفة هذه الزياداتمسبقاً ولكن يتم تحديدها أثناء التطوير. يعتمد القرار بشأن ما يجب تضمينه في الزيادةعلى التقدم المحرز وعلى أولويات العميل. الحجة في هذا النهج هي أن أولويات العميل ومتطلباتهتتغير ، لذلك من المنطقي أن يكون لديك خطة مرنة يمكنها استيعاب هذه التغييرات.يعد كتاب كوهن (Cohn 2005) مقدمة ممتازة للتخطيط السريع.

تتميزطرق التطوير الرشيقة مثل)Rubin 2013(Scrum و)and Andres 2004 Beck(Extreme Programming بنهج من مرحلتين للتخطيط ، بما يتوافق مع مرحلة بدء التشغيلفي التخطيط القائم على التخطيط والتطوير:

1. *تخطيط الإصدار* ،الذي يتطلع إلى المستقبل لعدة أشهر ويقرر الميزات التي يجب تضمينها في إصدار النظام.
2. *تخطيط التكرار* ،التي لها نظرة مستقبلية قصيرة المدى وتركز على تخطيط الزيادة التالية في النظام.يمثل هذا عادة من 2 إلى 4 أسابيع من العمل للفريق.

لقدشرحت بالفعل نهج Scrum للتخطيط في الفصل 3 ، والذي يعتمد على تراكم المشاريعوالمراجعات اليومية للعمل الذي يتعين القيام به. إنه موجه في المقام الأول

الشكل 23.8 ال
"لعبة التخطيط"

لتخطيط التكرار. نهج آخر للتخطيط السريع ، والذي تم تطويره كجزء من البرمجة المتطرفة ، يعتمد على قصص المستخدمين. يمكن استخدام ما يسمى بلعبة التخطيط في كل من تخطيط الإصدار وتخطيط التكرار.

أساس لعبة التخطيط (الشكل 23.8) هو مجموعة قصص المستخدمين (انظر الفصل 3) التي تغطي جميع الوظائف التي سيتم تضمينها في النظام النهائي. يعمل فريق التطوير وعميل البرنامج معاً لتطوير هذه القصص. يقرأ أعضاء الفريق القصص ويناقشونها ويرتبونها بناءً على مقدار الوقت الذي يعتقدون أنه سيستغرقه لتنفيذ القصة. قد تكون بعض القصص أكبر من أن يتم تنفيذها في تكرار واحد ، ويتم تقسيمها إلى قصص أصغر.

تكمن مشكلة ترتيب القصص في أن الأشخاص غالباً ما يجدون صعوبة في تقدير مقدار الجهد أو الوقت اللازم للقيام بشيء ما. لتسهيل هذا الأمر ، يمكن استخدام الترتيب النسبي. يقارن الفريق القصص في أزواج ويقرر أيها سيستغرق معظم الوقت والجهد ، دون تقييم مقدار الجهد المطلوب بالضبط. في نهاية هذه العملية ، تم ترتيب قائمة القصص ، مع بذل القصص في الجزء العلوي من القائمة أقصى جهد للتنفيذ. ثم يخصص الفريق نقاط جهد نظرية لجميع القصص في القائمة. قد تحتوي القصة المعقدة على 8 نقاط والقصة البسيطة نقطتان.

بمجرد تقدير القصص ، تتم ترجمة الجهد النسبي إلى التقدير الأول للجهد الإجمالي المطلوب باستخدام فكرة "السرعة". السرعة هي عدد نقاط الجهد التي ينفذها الفريق يومياً. يمكن تقدير ذلك إما من التجربة السابقة أو من خلال تطوير قصة أو قصتين لمعرفة مقدار الوقت المطلوب. تقدير السرعة تقريبي ولكن يتم تنقيحه أثناء عملية التطوير. بمجرد أن يكون لديك تقدير للسرعة ، يمكنك حساب الجهد الإجمالي في أيام الفرد لتنفيذ النظام.

يتضمن تخطيط الإصدار اختيار القصص وتنقيحها التي ستعكس الميزات التي سيتم تنفيذها في إصدار النظام والترتيب الذي يجب أن يتم تنفيذ القصص به. يجب أن يشارك العميل في هذه العملية. ثم يتم اختيار تاريخ الإصدار ، ويتم فحص القصص لمعرفة ما إذا كان تقدير الجهد متسقاً مع ذلك التاريخ. إذا لم يكن كذلك ، تتم إضافة القصص أو إزالتها من القائمة.

تخطيط التكرار هو المرحلة الأولى في تطوير زيادة نظام التسليم. يتم اختيار القصص التي سيتم تنفيذها خلال هذا التكرار ، مع عدد القصص التي تعكس الوقت اللازم لتقديم نظام عملي (عادة 2 أو 3 أسابيع) وسرعة الفريق. عند الوصول إلى تاريخ التسليم ، يكون تكرار التطوير قد اكتمل ، حتى لو لم يتم تنفيذ جميع القصص. يأخذ الفريق بعين الاعتبار القصص التي تم تنفيذها ويضيف نقاط جهدهم. يمكن بعد ذلك إعادة حساب السرعة ، ويستخدم هذا المقياس في التخطيط للإصدار التالي من النظام.

في بداية كل تكرار تطوير ، هناك مرحلة تخطيط مهمة حيث يقسم المطورون القصص إلى مهام تطوير. يجب أن تستغرق مهمة التطوير من 4 إلى 16 ساعة. يتم سرد جميع المهام التي يجب إكمالها لتنفيذ جميع المجموعات النصية في هذا التكرار. ثم يقوم المطورون الأفراد بالتسجيل في الموقع المحدد

المهامالتي سيقومون بتنفيذها. يعرف كل مطور سرعته الفردية ، لذا يجب ألا يشترك في مهامأكثر مما يمكنه تنفيذها في الوقت المخصص.
هذاالنهج لتخصيص المهام له فائدتان هامتان:

1.يحصل الفريق بأكمله على نظرة عامة على المهام المراد إكمالها في تكرار. لذلك لديهم فهم لمايفعله أعضاء الفريق الآخرون ومن يتحدثون إليه إذا تم تحديد تبعيات المهام.

2.المطورين الفرديين يختارون المهام المراد تنفيذها. لم يتم تخصيص المهام ببساطة من قبلمدير المشروع. لذلك فإن لديهم إحساساً بالملكية في هذه المهام ، وهذا من المرجح أنيحفزهم على إكمال المهمة.

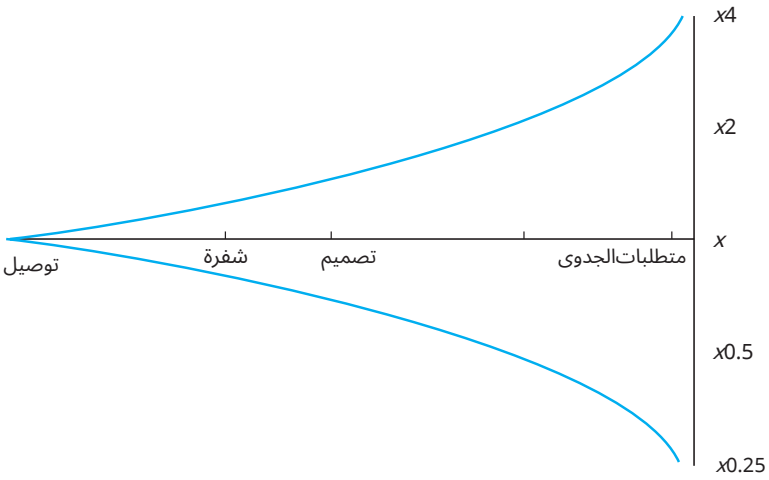
فيمنتصف الطريق خلال التكرار ، تتم مراجعة التقدم. في هذه المرحلة ، يجب إكمال نصفنقاط جهد القصة. لذلك ، إذا اشتمل التكرار على 24 نقطة قصة و 36 مهمة ، فيجب إكمال12 نقطة قصة و 18 مهمة. إذا لم يكن الأمر كذلك ، فيجب أن تكون هناك مناقشات معالعميل حول القصص التي يجب إزالتها من زيادة النظام التي يتم تطويرها.

يتميزهذا النهج في التخطيط بأنه يتم تسليم زيادة البرامج دائماً في نهاية كل تكرار للمشروع.إذا تعذر إكمال الميزات المراد تضمينها في الزيادة في الوقت المسموح به ، فسيتم تقليلنطاق العمل. لا يتم تمديد جدول التسليم. ومع ذلك ، يمكن أن يتسبب هذا في مشاكل لأنهيعني أن خطط العملاء قد تتأثر. قد يؤدي تقليل النطاق إلى إنشاء عمل إضافي للعملاء إذا كانعليهم استخدام نظام غير مكتمل أو تغيير طريقة عملهم بين إصدار واحد من النظام وآخر.

تتمثلإحدى الصعوبات الرئيسية في التخطيط السريع في أنه يعتمد على مشاركة العملاء وتوافرهم.قد يكون من الصعب ترتيب هذه المشاركة ، حيث يتعين على ممثلي العملاء في بعضالأحيان تحديد أولويات العمل الآخر ولا يكونون متاحين للعبة التخطيط. علاوة على ذلك ،قد يكون بعض العملاء أكثر دراية بخطط المشروع التقليدية وقد يجدون صعوبة في الانخراطفي عملية تخطيط رشيقة.
يعملالتخطيط السريع بشكل جيد مع فرق التطوير الصغيرة والمستقرة التي يمكنها الاجتماعومناقشة القصص التي سيتم تنفيذها. ومع ذلك ، عندما تكون الفرق كبيرة و / أو موزعةجغرافياً ، أو عندما تتغير عضوية الفريق بشكل متكرر ، فمن المستحيل عملياً أن يشاركالجميع في التخطيط التعاوني الضروري لإدارة المشروع السريع. وبالتالي ، عادة ما يتم التخطيطللمشاريع الكبيرة باستخدام الأساليب التقليدية لإدارة المشاريع.

23.5 تقنيات التقدير

تقديرجداول المشروع صعب. يجب عليك عمل تقديرات أولية على أساس تعريف غير مكتمل لمتطلباتالمستخدم. قد يتعين تشغيل البرنامج على منصات غير مألوفة أو استخدام تقنية تطويرجديدة. من المحتمل ألا يكون الأشخاص المشاركون في المشروع ومهاراتهم معروفين. هناكالكثير من أوجه عدم اليقين لدرجة أنه من المستحيل تقدير تكاليف تطوير النظام بدقة خلالالفترة المبكرة



الشكل 23.9 تقدير ريبة

مراحل المشروع. ومع ذلك ، تحتاج المؤسسات إلى بذل جهود البرامج وتقديرات التكلفة. يمكن استخدام نوعين من التقنيات لعمل التقديرات:

1. *التقنيات القائمة على الخبرة* يعتمد تقدير متطلبات الجهد المستقبلي على خبرة المدير في المشاريع السابقة ومجال التطبيق. بشكل أساسي ، يصدر المدير حكماً مستنيراً حول متطلبات الجهد المحتمل أن تكون.

2. *نمذجة التكلفة الحسابية* في هذا النهج ، يتم استخدام نهج معادلات لحساب جهد المشروع بناءً على تقديرات سمات المنتج ، مثل الحجم وخصائص العملية وخبرة الموظفين المعنيين.

في كلتا الحالتين ، تحتاج إلى استخدام حكمك لتقدير الجهد مباشرة أو خصائص المشروع والمنتج. في مرحلة بدء المشروع ، يكون لهذه التقديرات هامش خطأ واسع. استناداً إلى البيانات التي تم جمعها من عدد كبير من المشاريع ، Boehm et al. اكتشف (B. Boehm وآخرون 1995) أن تقديرات بدء التشغيل تختلف اختلافاً كبيراً. إذا كان التقدير الأولي للجهد المطلوب هو *x* أشهر من الجهد ، وجدوا أن النطاق قد يكون من 0.25*x* ل 4*x* من الجهد الفعلي كما تم قياسه عند تسليم النظام. أثناء التخطيط للتنمية ، تصبح التقديرات أكثر دقة مع تقدم المشروع (الشكل 23.9).

تعتمد التقنيات القائمة على الخبرة على خبرة المدير في المشاريع السابقة والجهد الفعلي المبذول في هذه المشاريع على الأنشطة المتعلقة بتطوير البرمجيات. عادةً ما تحدد النواتج التي سيتم إنتاجها في مشروع ومكونات أو أنظمة البرامج المختلفة التي سيتم تطويرها. تقوم بتوثيقها في جدول بيانات ، وتقديرها بشكل فردي ، وتحسب الجهد الإجمالي المطلوب. عادة ما يكون من المفيد إشراك مجموعة من الأشخاص في تقدير الجهد ومطالبة كل عضو في المجموعة بشرح تقديرهم. يكشف هذا غالباً عن عوامل لم يأخذها الآخرون في الاعتبار ، ثم تقوم بالتكرار نحو تقدير مجموعة متفق عليه.

تكمنالصعوبة في التقنيات القائمة على الخبرة في أن مشروع برمجيات جديد قد لا يكون لهالكثير من القواسم المشتركة مع المشاريع السابقة. يتغير تطوير البرامج بسرعة كبيرة ، وغالبأما يستخدم المشروع تقنيات غير مألوفة مثل خدمات الويب أو تكوين نظام التطبيق أو HTML5. إذا لم تكن قد عملت باستخدام هذه الأساليب ، فقد لا تساعدك تجربتك السابقة في تقدير الجهد المطلوب ، مما يزيد من صعوبة إنتاج تكاليف دقيقة وجدولة التقديرات.

من المستحيل تحديد ما إذا كانت المقاربات القائمة على الخبرة أو الخوارزميات أكثر دقة. غالبأما تكون تقديرات المشروع محققة لذاتها. يتم استخدام التقدير لتحديد ميزانية المشروع ، ويتم تعديل المنتج بحيث يتم تحقيق رقم الميزانية. قد يكون المشروع ضمن الميزانية قد حقق ذلك على حساب الميزات الموجودة في البرنامج الذي يتم تطويره.

لإجراء مقارنة بين دقة هذه التقنيات ، ستكون هناك حاجة لعدد من التجارب الخاضعة للرقابة حيث تم استخدام العديد من التقنيات بشكل مستقل لتقدير جهد المشروع وتكاليفه. لن يسُمح بأي تغييرات على المشروع ، ويمكن مقارنة الجهد النهائي بينهما. لن يعرف مدير المشروع تقديرات الجهود ، لذلك لن يتم إدخال أي تحيز. ومع ذلك ، فإن هذا السيناريو مستحيل تماماً في المشاريع الحقيقية ، لذلك لن يكون لدينا مقارنة موضوعية لهذه الأساليب.

23.5.1 نمذجة التكلفة الحسابية

تستخدم نمذجة التكلفة الحسابية صيغة رياضية للتنبؤ بتكاليف المشروع بناءً على تقديرات حجم المشروع ونوع البرنامج الذي يتم تطويره وعوامل الفريق والعملية والمنتج الأخرى. يتم تطوير نماذج التكلفة الحسابية من خلال تحليل تكاليف وسمات المشاريع المنجزة ، ثم إيجاد الصيغة الأقرب للتكاليف الفعلية المتكبدة.

تسُتخدم نماذج التكلفة الخوارزمية بشكل أساسي لعمل تقديرات لتكاليف تطوير البرمجيات. ومع ذلك ، فإن Boehm ومعاونيه (BW Boehm وآخرون 2000) يناقشون مجموعة من الاستخدامات الأخرى لهذه النماذج ، مثل إعداد التقديرات للمستثمرين في شركات البرمجيات ، والاستراتيجيات البديلة للمساعدة في تقييم المخاطر ولإبلاغ القرارات حول إعادة الاستخدام وإعادة التطوير. أو الاستعانة بمصادر خارجية.
تعتمد معظم النماذج الخوارزمية لتقدير الجهد في مشروع برمجي على صيغة بسيطة:

الجهد = أ3 بحجم ب3 م

أ: عامل ثابت يعتمد على الممارسات التنظيمية المحلية ونوع البرمجيات التي يتم تطويرها.

بحجم: تقييم لحجم رمز البرنامج أو تقدير وظيفي معبر عنه في نقاط الوظيفة أو التطبيق.

ب: يمثل تعقيد البرنامج وعادة ما يقع بين 1 و 1.5.

م: هو عامل يأخذ في الاعتبار سمات العملية والمنتج والتطوير ، مثل متطلبات الموثوقية للبرنامج وتجربة فريق التطوير. قد تزيد هذه السمات أو تقلل من الصعوبة الإجمالية لتطوير النظام.

عددسطور التعليمات البرمجية المصدر (SLOC) في النظام المقدم هو مقياس الحجم الأساسيالمستخدم في العديد من نماذج التكلفة الحسابية. لتقدير عدد سطور التعليمات البرمجيةفي نظام ما ، يمكنك استخدام مجموعة من الأساليب:

1.قارن النظام المراد تطويره بأنظمة مماثلة واستخدم حجم الكود الخاص بهم كأساس لتقديرك.

2.قم بتقدير عدد نقاط الوظيفة أو التطبيق في النظام (انظر القسم التالي) وقم بتحويلها إلى سطورمن التعليمات البرمجية في لغة البرمجة المستخدمة.

3.رتب مكونات النظام باستخدام الحكم على أحجامها النسبية واستخدم مكوناً مرجعياً معروفاً لترجمةهذا الترتيب إلى أحجام الكود.

تحتويمعظم نماذج التقدير الخوارزمية على مكون أسي (بفي المعادلة أعلاه) التي تزيد مع حجموتعقيد النظام. وهذا يعكس حقيقة أن التكاليف لا تزداد عادة بشكل خطي مع حجم المشروع.مع زيادة حجم البرنامج وتعقيده ، يتم تكبد تكاليف إضافية بسبب عبء الاتصال للفرقالأكبر ، وإدارة التكوين الأكثر تعقيداً ، وتكامل النظام الأكثر صعوبة ، وما إلى ذلك. كلما زادتعقيد النظام ، زادت هذه العوامل التي تؤثر على التكلفة.

تعدفكرة استخدام نهج علمي وموضوعي لتقدير التكلفة فكرة جذابة ، لكن جميع نماذج التكلفةالخوارزمية تعاني من مشكلتين رئيسيتين:

1.من المستحيل عمليا التقديربحجمبدقة في مرحلة مبكرة من المشروع ، عندما تكون المواصفاتفقط متاحة. تعد تقديرات نقطة الوظيفة ونقطة التطبيق (انظر لاحقاً) أسهل فيالإنتاج من تقديرات حجم الشفرة ولكنها أيضاً غير دقيقة في العادة.

2.تقديرات التعقيد وعملية العوامل المساهمة فيبوهي شخصية. تختلف التقديرات من شخصإلى آخر ، اعتماداً على خلفيتهم وخبرتهم في نوع النظام الذي يتم تطويره.

يعدالتقدير الدقيق لحجم الكود أمراً صعباً في مرحلة مبكرة من المشروع لأن حجم البرنامج النهائييعتمد على قرارات التصميم التي ربما لم يتم اتخاذها عندما يكون التقدير مطلوباً. علىسبيل المثال ، التطبيق الذي يتطلب إدارة بيانات عالية الأداء قد يقوم إما بتنفيذ نظام إدارةالبيانات الخاص به أو استخدام نظام قاعدة بيانات تجاري. في التقدير الأولي للتكلفة ، من غيرالمحتمل أن تعرف ما إذا كان هناك نظام قاعدة بيانات تجاري يعمل جيداً بما يكفي لتلبية متطلباتالأداء. لذلك أنت لا تعرف مقدار كود إدارة البيانات الذي سيتم تضمينه في النظام.

تؤثرلغة البرمجة المستخدمة لتطوير النظام أيضاً على عدد سطور التعليمات البرمجية المطلوبتطويرها. قد تعني لغة مثل Java أن عدد سطور التعليمات البرمجية ضرورية أكثر ممالو تم استخدام C (على سبيل المثال). ومع ذلك ، يسمح هذا الرمز الإضافي بمزيد من التحققمن وقت الترجمة ، لذلك من المرجح أن يتم تقليل تكاليف التحقق. ليس من الواضح كيفيجب أن يؤخذ ذلك في الاعتبار في عملية التقدير. إعادة استخدام الكود أيضاً

إنتاجيةالبرمجيات

إنتاجيةالبرامج هي تقدير لمتوسط مقدار أعمال التطوير التي يكملها مهندسو البرمجيات في أسبوع أو شهر. لذلك يتم التعبير عنها كسطر منالكود / الشهر ، ونقاط الوظيفة / الشهر ، وما إلى ذلك.
ومعذلك ، في حين يمكن قياس الإنتاجية بسهولة حيث توجد نتيجة ملموسة (على سبيل المثال ، عمليات المسؤولنمطالبات السفر / اليوم) ، يصعب تحديد إنتاجية البرامج. قد يقوم الأشخاص المختلفون بتنفيذ نفس الوظيفة بطرق مختلفة ، باستخدام أعداد مختلفة من سطورالتعليمات البرمجية. تعد جودة الكود مهمة أيضاً ولكنها إلى حد ما ذاتية. لذلك ، لا يمكنك حقاً مقارنة إنتاجية المهندسين الفرديين. من المنطقيفقط استخدام تدابير الإنتاجية مع مجموعات كبيرة.

http://software-engineering-book.com/web/productivity/

يحدثفرقاً ، وبعض النماذج تقدر صراحة عدد سطور الكود المعاد استخدامها. ومع ذلك ، إذا أعيداستخدام أنظمة التطبيقات أو الخدمات الخارجية ، فمن الصعب للغاية حساب عدد سطورالتعليمات البرمجية المصدر التي تحل محلها.

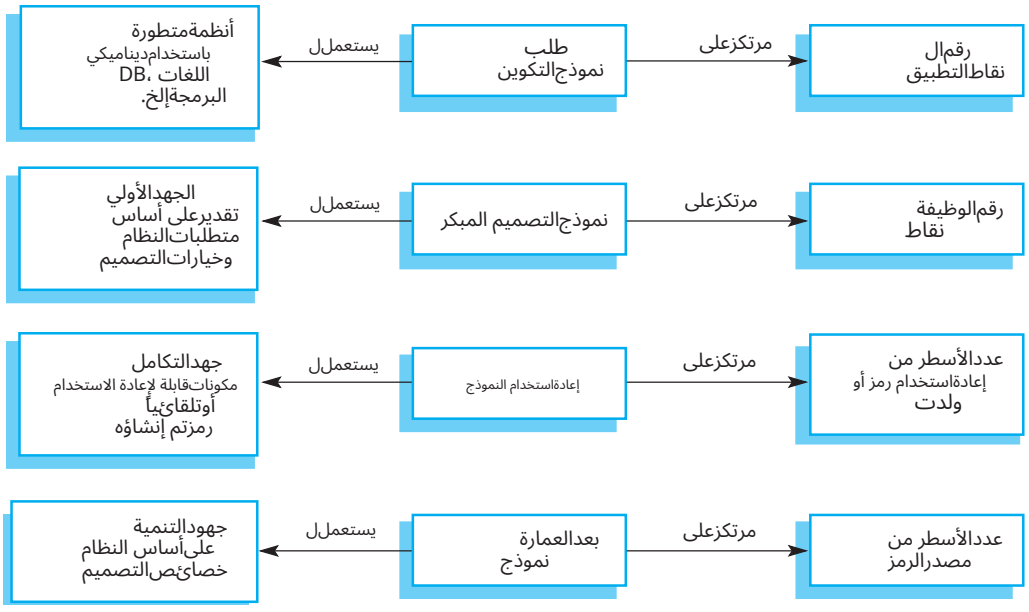
نماذجالتكلفة الحسابية هي طريقة منهجية لتقدير الجهد المطلوب لتطوير النظام. ومع ذلك ،فإن هذه النماذج معقدة وصعبة الاستخدام. هناك العديد من السمات ونطاق كبير لعدم اليقينفي تقدير قيمها. يعني هذا التعقيد أن التطبيق العملي لنمذجة التكلفة الخوارزمية قد اقتصرعلى عدد صغير نسبياً من الشركات الكبيرة ، والتي تعمل في الغالب في هندسة أنظمة الدفاعوالفضاء.

الحاجزالآخر الذي لا يشجع على استخدام النماذج الحسابية هو الحاجة إلى المعايرة. يجب علىمستخدمي النموذج معايرة نموذجهم وقيم السمات باستخدام بيانات المشروع التاريخية الخاصةبهم ، لأن هذا يعكس الممارسة والخبرة المحلية. ومع ذلك ، هناك عدد قليل جداً من المؤسساتالتي جمعت بيانات كافية من المشاريع السابقة في شكل يدعم معايرة النموذج. لذلك ،يجب أن يبدأ الاستخدام العملي للنماذج الخوارزمية بالقيم المنشورة لمعلمات النموذج.من المستحيل عمليا على المصمم أن يعرف مدى ارتباطها بمنظمته.

إذاكنت تستخدم نموذج تقدير التكلفة الخوارزمي ، فيجب عليك تطوير نطاق من التقديرات(الأسوأ ، والمتوقع ، والأفضل) بدلاً من تقدير واحد وتطبيق معادلة تقدير التكلفة علىكل منهم. من المرجح أن تكون التقديرات دقيقة عندما تفهم نوع البرنامج الذي يتم تطويره وقمتبمعايرة نموذج التكلفة باستخدام البيانات المحلية ، أو عندما تكون خيارات لغة البرمجة والأجهزةمحددة مسبقاً.

23.6 نمذجة تكلفة COCOMO

إنأفضل تقنية وأداة لنمذجة التكلفة الحسابية المعروفة هي نموذج COCOMO II. تم اشتقاقهذا النموذج التجريبي من خلال جمع البيانات من عدد كبير من مشاريع البرامج ذات الأحجامالمختلفة. تم تحليل هذه البيانات لاكتشاف الصيغ الأكثر ملاءمة للملاحظات. ربطت هذهالصيغ حجم



الشكل23.10كوكومو نماذجالتقدير

النظاموالمنتج والمشروع وعوامل الفريق للجهود المبذولة لتطوير النظام. COCOMO II هو نموذجمتاح مجاناً ومدعوم بأدوات مفتوحة المصدر.

تمتطوير COCOMO II من نماذج تقدير تكلفة COCOMO (نمذجة التكلفة البنائية) السابقة ،والتي كانت تعتمد إلى حد كبير على تطوير الكود الأصلي (BW Boehm 1981 ؛ B. Boehm and Royce1989). يأخذ نموذج COCOMO II في الاعتبار الأساليب الحديثة لتطويرالبرامج ، مثل التطوير السريع باستخدام اللغات الديناميكية ، والتطوير مع إعادة الاستخدام ،وبرمجة قواعد البيانات. يقوم COCOMO II بتضمين العديد من النماذج الفرعية بناءً على هذه التقنيات ، والتي تنتج تقديرات مفصلة بشكل متزايد.

النماذجالفرعية (الشكل 23.10) التي تعد جزءاً من نموذج COCOMO II هي:

1. *نموذج تكوين التطبيق*يمثل هذا الجهد المطلوب لتطوير الأنظمة التي تم إنشاؤها من مكوناتقابلة لإعادة الاستخدام أو البرمجة النصية أو برمجة قواعد البيانات. تعتمد تقديراتحجم البرنامج على نقاط التطبيق ، ويتم استخدام صيغة بسيطة للحجم / الإنتاجيةلتقدير الجهد المطلوب.
2. *نموذج تصميم مبكر*يستخدم هذا النموذج خلال المراحل الأولى من تصميم النظام بعد تحديدالمتطلبات. يعتمد التقدير على صيغة التقدير القياسية التي ناقشتها في مقدمة هذاالفصل ، مع مجموعة مبسطة من سبعة مضاعفات. تستند التقديرات إلى نقاط الوظيفة ،والتي يتم تحويلها بعد ذلك إلى عدد سطور التعليمات البرمجية المصدر.

النقاطالوظيفية هي طريقة مستقلة عن اللغة لقياس وظائف البرنامج. يمكنك حساب العددالإجمالي لنقاط الوظائف في برنامج عن طريق قياس أو تقدير عدد المدخلات والمخرجاتالخارجية وتفاعلات المستخدم والواجهات الخارجية والملفات أو جداول قاعدةالبيانات التي يستخدمها النظام.

3.*نموذج إعادة الاستخدام*يستخدم هذا النموذج لحساب الجهد المطلوب لدمج المكونات القابلةلإعادة الاستخدام و / أو رمز البرنامج الذي تم إنشاؤه تلقائياً. يتم استخدامه عادةً معنموذج ما بعد العمارة.

4.*نموذج ما بعد العمارة*بمجرد تصميم بنية النظام ، يمكن إجراء تقدير أكثر دقة لحجم البرنامج. مرةأخرى ، يستخدم هذا النموذج الصيغة القياسية لتقدير التكلفة التي تمت مناقشتها أعلاه.ومع ذلك ، فهو يتضمن مجموعة أكثر شمولاً من 17 مضاعفاً تعكس قدرة الأفراد والمنتجوخصائص المشروع.

بالطبع ،في الأنظمة الكبيرة ، قد يتم تطوير أجزاء مختلفة من النظام باستخدام تقنيات مختلفة ،وقد لا تضطر إلى تقدير جميع أجزاء النظام بنفس مستوى الدقة. في مثل هذه الحالات ،يمكنك استخدام النموذج الفرعي المناسب لكل جزء من النظام ودمج النتائج لإنشاءتقدير مركب.

نموذجCOCOMO II هو نموذج معقد للغاية ، ولتسهيل شرحه ، قمت بتبسيط عرضه. يمكنكاستخدام النماذج كما أوضحتها هنا لتقدير بسيط للتكلفة. ومع ذلك ، لاستخدام COCOMO بشكل صحيح ، يجب الرجوع إلى كتاب Boehm ودليل نموذج COCOMO II (BW Boehm وآخرون 2000 ؛ Abts وآخرون 2000).

23.6.1نموذج تكوين التطبيق

تمتقديم نموذج تكوين التطبيق في COCOMO II لدعم تقدير الجهد المطلوب لمشاريع النماذجالأولية وللمشاريع التي يتم فيها تطوير البرنامج من خلال تكوين المكونات الحالية. وهو يعتمدعلى تقدير نقاط التطبيق الموزونة (تسمى أحياناً نقاط الكائن) ، مقسومة على تقدير قياسيلإنتاجية نقطة التطبيق (BW Boehm وآخرون 2000). يتم اشتقاق عدد نقاط التطبيق فيالبرنامج من أربعة تقديرات أبسط:

- عددالشاشات المنفصلة أو صفحات الويب التي يتم عرضها ؛
- عددالتقارير التي يتم إنتاجها ؛
- عددالوحدات في لغات البرمجة الحتمية (مثل Java) ؛ و
- عددسطور لغة البرمجة النصية أو كود برمجة قاعدة البيانات.

ثميتم تعديل هذا التقدير وفقاً لصعوبة تطوير كل نقطة تطبيق. تعتمد الإنتاجية على خبرة المطوروقدرته بالإضافة إلى إمكانيات أدوات البرمجيات (ICASE) المستخدمة لدعم التطوير. يوضحالشكل 23.11 مستويات إنتاجية نقطة التطبيق التي اقترحها مطورو نموذج COCOMO .

يعتمدتكوين التطبيق عادةً على إعادة استخدام البرامج الموجودة وتكوين أنظمة التطبيقات.لذلك سيتم تنفيذ بعض نقاط التطبيق في النظام باستخدام مكونات قابلة لإعادة الاستخدام.وبالتالي ، يجب عليك ضبط ملف

المطور الخبرةو الإمكانية	منخفضجدا	قليل	اسمي،صوري شكلي، بالاسم فقط	عالٍ	عاليجدا
نضجICASE و الإمكانية	منخفضجدا	قليل	اسمي،صوري شكلي، بالاسم فقط	عالٍ	عاليجدا
)قيلولة / شهر(PROD	4	7	13	25	50

الشكل23.11 طلب-إنتاجيةالنقطة

تقديرلمراعاة النسبة المئوية المتوقعة لإعادة الاستخدام. لذلك ، فإن الصيغة النهائية لحسابالجهد لنماذج النظام هي:

مساءً5 (قيلولة3 (1٪2إعادة استخدام / 100)) / PROD

مساء:تقدير الجهد في شخص-أشهر.

قيلولة:العدد الإجمالي لنقاط التطبيق في النظام الذي تم تسليمه.

٪إعادةاستخدام:تقدير لمقدار الكود المعاد استخدامه في التطوير.

همز:إنتاجية نقطة التطبيق كما هو موضح في الشكل 23.11.

23.6.2نموذج التصميم المبكر

يمكناستخدام هذا النموذج خلال المراحل الأولى من المشروع ، قبل أن يتوفر تصميم معماريمفصل للنظام. يفترض نموذج التصميم المبكر أنه تم الاتفاق على متطلبات المستخدموأن المراحل الأولية لعملية تصميم النظام جارية. يجب أن يكون هدفك في هذه المرحلةهو إجراء تقدير سريع وتقريبي للتكلفة. لذلك ، يجب عليك وضع افتراضات مبسطة ، مثلافتراض عدم وجود جهد مشترك في دمج التعليمات البرمجية القابلة لإعادة الاستخدام.

تعدتقديرات التصميم المبكرة مفيدة للغاية لاستكشاف الخيارات حيث تحتاج إلى مقارنة الطرقالمختلفة لتنفيذ متطلبات المستخدم. تستند التقديرات التي تم إنتاجها في هذه المرحلةإلى الصيغة القياسية للنماذج الحسابية ، وهي:

جهد5أ3بحجمب3م

بناءً على مجموعة البيانات الكبيرة الخاصة به ، اقترح Boehm أن يكون فعالاًأيجب ان يكون 2.94يتم التعبير عن حجم النظام في KSLOC ، وهو عدد آلاف الأسطر من التعليمات البرمجية المصدر.يمكنك حساب KSLOC عن طريق تقدير عدد نقاط الوظيفة في البرنامج. يمكنك بعد ذلكاستخدام الجداول القياسية ، التي تربط حجم البرنامج بالنقاط الوظيفية للغات البرمجة المختلفة(QSM 2014) لحساب تقدير أولي لحجم النظام في KSLOC.

الأسبيعكس الجهد المتزايد المطلوب مع زيادة حجم المشروع. يمكن أن يختلف هذا من 1.1إلى 1.24 اعتماداً على حداثة المشروع ، ومرونة التطوير ، وعمليات حل المخاطر المستخدمة ،وتماسك فريق التطوير ، ومستوى نضج العملية (انظر الفصل 26 على الويب) للمؤسسة.أناقش كيفية حساب قيمة هذا الأس باستخدام هذه المعلمات في وصف نموذج العمارةاللاحقة COCOMO II.

ينتجعن هذا حساب الجهد على النحو التالي:

مساءً32.945بحجم(1.1 إلى 1.24)3م

م3PERS5رئيس الجامعة3RCPX3حيلة3PDIF3FSIL3SCED

الأشخاص:قدرة الأفراد

رئيسالجامعة:خبرة الموظفين

:RCPXموثوقية المنتج وتعقيده

حيلة:إعادة الاستخدام المطلوبة

:PDIFصعوبة النظام الأساسي

:SCEDبرنامج

:FSILمرافق الدعم

المضاعفـمعتمد على سبع سمات للمشروع والعملية تزيد أو تقلل من التقدير. أشرح هذه السماتعلى صفحات الويب للكتاب. أنت تقدر قيم هذه السمات باستخدام مقياس من ستنقاط ، حيث يتوافق 1 مع "منخفض جداً" و 6 يتوافق مع "مرتفع جداً" ؛ فمثلا،بيرس = 6 يعنيأن الموظفين الخبراء متاحون للعمل في المشروع.

23.6.3نموذج إعادة الاستخدام

يسُتخدمنموذج إعادة استخدام COCOMO لتقدير الجهد المطلوب لدمج الكود القابل لإعادة الاستخدامأو المنُشأ. كما ناقشت في الفصل الخامس عشر ، أصبحت إعادة استخدام البرامج الآن هيالمعيار في جميع عمليات تطوير البرامج. تشتمل معظم الأنظمة الكبيرة على قدر كبير من التعليماتالبرمجية التي تم إعادة استخدامها من مشاريع التطوير السابقة.

يعتبرII COCOMO نوعين من التعليمات البرمجية المعاد استخدامها. رمز الصندوق الأسودهو رمز يمكن إعادة استخدامه دون فهم الكود أو إجراء تغييرات عليه. أمثلة رمز الصندوقالأسود هي المكونات التي يتم إنشاؤها تلقائياً من نماذج UML أو مكتبات التطبيقاتمثل مكتبات الرسومات. من المفترض أن جهود تطوير كود الصندوق الأسود تساويصفراً. لا يؤخذ حجمها في الاعتبار في حساب الجهد الكلي.

كودالصندوق الأبيض عبارة عن كود قابل لإعادة الاستخدام يجب تكييفه لدمجه مع كود جديدأو مكونات أخرى معاد استخدامها. جهود التطوير مطلوبة لإعادة الاستخدام لأنه يجب فهمالكود وتعديله قبل أن يعمل بشكل صحيح في النظام. يمكن إنشاء كود الصندوق الأبيض تلقائيأوالذي يحتاج إلى تغييرات يدوية أو إضافات. بدلاً من ذلك ، يمكن إعادة استخدام مكوناتمن أنظمة أخرى يجب تعديلها في النظام الذي يتم تطويره.

هناكثلاثة عوامل تساهم في الجهد المبذول في إعادة استخدام مكونات كود الصندوق الأبيض:

1.الجهد المبذول في تقييم ما إذا كان من الممكن إعادة استخدام أحد المكونات في نظام يتم تطويره.

2.الجهد المطلوب لفهم الكود الذي يتم إعادة استخدامه.

3.الجهد المطلوب لتعديل الكود المعاد استخدامه لمواءمته مع النظام الجاري تطويره.

يتمحساب جهد التطوير في نموذج إعادة الاستخدام باستخدام نموذج التصميم المبكر COCOMO ويستند إلى العدد الإجمالي لأسطر الكود في النظام. يتضمن حجم الكود رمزاً جديداًتم تطويره للمكونات التي لم يتم إعادة استخدامها بالإضافة إلى عامل إضافي يسمح بالجهدالمبذول في إعادة استخدام التعليمات البرمجية الحالية ودمجها. هذا العامل الإضافي يسمى، ESLOCالعدد المكافئ لأسطر شفرة المصدر الجديدة. أي أنك تعبر عن جهد إعادة الاستخدامباعتباره الجهد الذي سيشُارك في تطوير بعض التعليمات البرمجية المصدر الإضافية.

الصيغةالمستخدمة لحساب معادلة كود المصدر هي:

AAM(31-AT / 100() 3ASLOC) 5ESLOC

:ESLOCالعدد المكافئ لأسطر شفرة المصدر الجديدة.

:ASLOCتقدير لعدد سطور التعليمات البرمجية في المكونات المعاد استخدامها التي يجب تغييرها.

في:النسبة المئوية للرمز المعاد استخدامه والذي يمكن تعديله تلقائياً.

آم:مضُاعفِ تعديل التكيف الذي يعكس الجهد الإضافي المطلوب لإعادة استخدام المكونات.

فيبعض الحالات ، تكون التعديلات المطلوبة لإعادة استخدام الشفرة نحوية ويمكن تنفيذهابواسطة أداة آلية. لا تتطلب هذه مجهوداً كبيراً ، لذا يجب عليك تقدير جزء التغييرات التيتم إجراؤها على التعليمات البرمجية المعاد استخدامها التي يمكن أتمتتها (في).هذا يقلل منالعدد الإجمالي لأسطر التعليمات البرمجية التي يجب تكييفها.

مضُاعفِتعديل التكيفٌ ()AAMيضبط التقدير ليعكس الجهد الإضافي المطلوب لإعادة استخدامالكود. تناقش وثائق نموذج)COCOMO)Abts et al. 2000 بالتفصيل كيفية القيامبذلكآميجب أن تحسب. بشكل مبسط ،آمهو مجموع ثلاثة مكونات:

1.عامل تقييم (يشار إليه باسم)AAيمثل الجهد المبذول في تقرير ما إذا كان سيتم إعادة استخدامالمكونات أم لا.AAيختلف من 0 إلى 8 اعتماداً على مقدار الوقت الذي تحتاجه للبحثعن المرشحين المحتملين وتقييمهم لإعادة الاستخدام.

2.مكون الفهم (يشار إليه باسم)SUالتي تمثل تكاليف فهم الكود المراد إعادة استخدامه ومعرفةالمهندس بالكود الذي يتم إعادة استخدامه.SUتتراوح من 50 للشفرة المعقدة وغيرالمهيكلة إلى 10 للتعليمات البرمجية المكتوبة جيداً والموجهة للكائنات.

3.مكون التكيف (يشار إليه باسم)AAFالتي تمثل تكاليف إجراء تغييرات على الكود المعاد استخدامه.وتشمل هذه التغييرات التصميم والرمز والتكامل.

بمجردحساب قيمة لـ، ESLOCتقوم بتطبيق صيغة التقدير القياسية لحساب الجهد الإجماليالمطلوب ، حيث تكون معلمة الحجم =.ESLOC لذلك ، فإن الصيغة لتقدير جهد إعادة الاستخدامهي:

جهد5أ3ESLOCب3م

أينأ ، ب ،ومهلها نفس القيم المستخدمة في نموذج التصميم المبكر.

تكلفةCOCOMO السائقين

محركاتتكلفة COCOMO II هي سمات تعكس بعض المنتجات والفريق والعملية والعوامل التنظيمية التي تؤثر على مقدار الجهد المطلوبلتطوير نظام برمجي. على سبيل المثال ، إذا كانت هناك حاجة إلى مستوى عالٍمن الموثوقية ، فستكون هناك حاجة إلى بذل جهد إضافي ؛إذا كانت هناك حاجة للتسليم السريع ، فسيلزم بذل جهد إضافي ؛ إذا تغير أعضاء الفريق ، فسيلزم بذل جهد إضافي.

هناك17 سمة من هذه السمات في نموذج COCOMO II ، والتي تم تخصيص قيم تقديرية لها من قبل مطوري النموذج.

http://software-engineering-book.com/web/cost-drivers/

23.6.4مستوى ما بعد العمارة

نموذجما بعد العمارة هو الأكثر تفصيلاً في نماذج COCOMO II. يتم استخدامه عندما يكون لديكتصميم معماري أولي للنظام. نقطة البداية للتقديرات المنتجة على مستوى ما بعد العمارةهي نفس الصيغة الأساسية المستخدمة في تقديرات التصميم المبكرة:

مساءً5أ3بحجمب3م

في هذه المرحلة من العملية ، يجب أن تكون قادراً على إجراء تقدير أكثر دقة لحجم المشروع ،كما تعلم كيف سيتحلل النظام إلى أنظمة فرعية ومكونات. يمكنك إجراء هذا التقدير لحجمالشفرة الكلي عن طريق إضافة ثلاثة تقديرات لحجم الشفرة:

1. تقدير للعدد الإجمالي لأسطر الكود الجديد المطلوب تطويره (SLOC).
2. تقدير لتكاليف إعادة الاستخدام بناءً على عدد مكافئ من سطور مصدر الشفرة (ESLOC،) محسوبةباستخدام نموذج إعادة الاستخدام.
3. تقدير لعدد سطور التعليمات البرمجية التي يمكن تغييرها بسبب التغييرات في متطلبات النظام.

المكونالأخير في التقدير - عدد سطور الكود المعدل يعكس حقيقة أن متطلبات البرامج تتغيردائماً. هذا يؤدي إلى إعادة صياغة وتطوير التعليمات البرمجية الإضافية ، والتي يجب أنتأخذها في الاعتبار. بالطبع سيكون هناك غالباً عدم يقين في هذا الرقم أكثر من تقديرات الكودالجديد الذي سيتم تطويره.

مصطلحالأس (ب)في صيغة حساب الجهد يرتبط بمستويات تعقيد المشروع. عندما تصبحالمشاريع أكثر تعقيداً ، تصبح تأثيرات زيادة حجم النظام أكثر أهمية. قيمة الأسبيعتمد علىخمسة عوامل ، كما هو موضح في الشكل 23.12. تم تصنيف هذه العوامل على مقياس منست نقاط من 0 إلى 5 ، حيث يعني 0 "مرتفع جداً" ويعني 5 "منخفض جداً". لكي يحسب ب،يمكنك إضافة التصنيفات ، وتقسيمها على 100 ، وإضافة النتيجة إلى 1.01 للحصول على الأسالذي يجب استخدامه.

عامل المقياس	تفسير
الهيكلية / حل المخاطر	يعكس مدى تحليل المخاطر المنفذة. منخفض جداً يعني القليل من التحليل ؛ تعني كلمة "high-Extra" إجراء تحليل كامل وشامل للمخاطر.
مرونة التطوير	يعكس درجة المرونة في عملية التطوير. منخفض جداً يعني استخدام عملية محددة ؛ تعني كلمة " عالية جداً" أن العميل يضع أهدافاً عامة فقط.
الأسبقية	يعكس الخبرة السابقة للمنظمة مع هذا النوع من المشاريع. منخفض جداً يعني عدم وجود خبرة سابقة ؛ تعني كلمة "عالٍ جداً" أن المؤسسة على دراية تامة بمجال التطبيق هذا.
تماسك الفريق	يعكس مدى معرفة فريق التطوير ببعضهم البعض والعمل معاً. منخفض جداً يعني تفاعلات صعبة للغاية ؛ فائق الارتفاع يعني وجود فريق متكامل وفعال بدون مشاكل في الاتصال.
نضج العملية	يعكس نضج عملية المنظمة كما تمت مناقشته في فصل الويب 26. يعتمد حساب هذه القيمة على استبيان نضج CMM ، ولكن يمكن تحقيق تقدير عن طريق طرح مستوى نضج عملية CMM من 5.

الشكل 23.12 مقياس العوامل المستخدمة في حساب الأس في نموذج ما بعد العمارة

على سبيل المثال ، تخيل أن منظمة تتولى مشروعاً في مجال ليس لديها خبرة سابقة فيه. لم يحدد عميل المشروع العملية التي سيتم استخدامها أو الوقت المسموح به في الجدول الزمني للمشروع لتحليل المخاطر الكبيرة. يجب تشكيل فريق تطوير جديد لتنفيذ هذا النظام. وضعت المنظمة مؤخراً برنامجاً لتحسين العملية وتم تصنيفها كمنظمة من المستوى 2 وفقاً لتقييم قدرة SEI ، كما تمت مناقشته في الفصل 26 (فصل الويب). تؤدي هذه الخصائص إلى تقديرات التصنيفات المستخدمة في حساب الأس على النحو التالي:

1. *أسبقية* منخفضة التصنيف (4). هذا مشروع جديد للمنظمة.
2. *مرونة التطوير* ، مصنفة عالية جدا (1). لا يوجد أي مشاركة للعميل في عملية التطوير ، لذلك هناك القليل من التغييرات المفروضة من الخارج.
3. *تحليل العمارة / المخاطر* ، مصنفة منخفضة جدا (5). لم يتم إجراء تحليل للمخاطر.
4. *تماسك الفريق*، التصنيف الاسمي (3). هذا فريق جديد ، لذا لا توجد معلومات متاحة حول التماسك.
5. *نضج العملية* ، التصنيف الاسمي (3). بعض عمليات التحكم في مكانها الصحيح.

مجموع هذه القيم هو 16. ثم تقوم بحساب القيمة النهائية للأس عن طريق قسمة هذا المجموع على 100 وإضافة 0.01 إلى النتيجة. القيمة المعدلة ب لذلك هو 1.17.

يتم تنقيح تقدير الجهد الإجمالي باستخدام مجموعة واسعة من 17 منتجاً وعملية وسمات تنظيمية (انظر مربع الاختصار) بدلاً من السمات السبع المستخدمة في نموذج التصميم المبكر. يمكنك تقدير قيم هذه السمات لأن لديك المزيد من المعلومات حول البرنامج نفسه ، ومتطلباته غير الوظيفية ، وفريق التطوير ، وعملية التطوير.

قيمةالأس	1.17
حجمالنظام (بما في ذلك عوامل إعادة الاستخدام وتقلبالمتطلبات)	128 KLOC
تقديرCOCOMO الأولي بدون محركات تكلفة	**730شخص - شهر**
مصداقية	مرتفعجدا ، المضاعف = 1.39
تعقيد	مرتفعجدا ، المضاعف = 1.3
قيودالذاكرة	عالية ،المضاعف = 1.21
استخدامالأداة	منخفض ،المضاعف = 1.12
برنامج	المعجل ،المضاعف = 1.29
تقديرCOCOMO المعدل	**2306شخص شهر**
مصداقية	منخفضجدا ، المضاعف = 0.75
تعقيد	منخفضجدا ، المضاعف = 0.75
قيودالذاكرة	لاشيء ، المضاعف = 1
استخدامالأداة	مرتفعجدا ، المضاعف = 0.72
برنامج	عادي ،مضاعف = 1
تقديرCOCOMO المعدل	**295شخص - شهر**

الشكل23.13 تأثيرمحركات التكلفة علىالجهد التقديرات

يوضحالشكل 23.13 كيف تؤثر سمات محرك التكلفة على تقديرات الجهود. افترض أن قيمةالأس هي 1.17 كما تمت مناقشته في المثال أعلاه. مصداقية (يعتمد)،تعقيد (،)CPLXتخزين (،)STORأدوات (أداة)،والجدول الزمني ()SCEDهي محركات التكلفةالرئيسية في المشروع. جميع محركات التكلفة الأخرى لها قيمة اسمية قدرها 1 ، لذا فهيلا تؤثر على حساب الجهد.

في الشكل 23.13 ، قمت بتعيين القيم القصوى والدنيا لمحركات التكلفة الرئيسية لإظهاركيفية تأثيرها على تقدير الجهد. القيم المستخدمة هي تلك من الدليل المرجعي COCOMO II)Abts et al. 2000(. يمكنك أن ترى أن القيم العالية لمحركات التكلفة تؤدي إلىتقدير جهد يزيد عن ثلاثة أضعاف التقدير الأولي ، بينما تقلل القيم المنخفضة التقدير إلى حواليثلث القيمة الأصلية. وهذا يسلط الضوء على الاختلافات الكبيرة بين الأنواع المختلفة للمشروعوصعوبات نقل الخبرة من مجال تطبيق إلى آخر.

23.6.5مدة المشروع والتوظيف

بالإضافةإلى تقدير التكاليف الإجمالية للمشروع والجهد المطلوب لتطوير نظام برمجي ، يجب علىمديري المشروع أيضاً تقدير المدة التي سيستغرقها تطوير البرنامج ومتى ستكون هناك حاجةإلى موظفين للعمل في المشروع. تطالب المؤسسات بشكل متزايد بجداول تطوير أقصر بحيثيمكن طرح منتجاتها في السوق قبل منافسيها.

يتضمن نموذج COCOMO صيغة لتقدير وقت التقويم المطلوب لإكمال المشروع:

TDEV335 (مساءً) = (0.210.33 * (ب1.012))

TDEV: الجدول الزمني الاسمي للمشروع ، في الأشهر التقويمية ، مع تجاهل أي مضاعف مرتبط بجدول المشروع.

مساءً: الجهد المحسوب بواسطة نموذج COCOMO.

ب: الأس المرتبط بالتعقيد ، كما تمت مناقشته في القسم 23.5.2. إذا ب = 1.175 ثم

TDEV335 (60) = 1350.36 (60) شهر

لايتوافق الجدول الزمني الاسمي للمشروع الذي تنبأ به نموذج COCOMO بالضرورة مع الجدول الزمني المطلوب من قبل عميل البرنامج. قد تضطر إلى تسليم البرنامج في وقت أبكر أو (نادراً) بعد التاريخ الذي يقترحه الجدول الاسمي. إذا كان الجدول الزمني مضغوطاً (على سبيل المثال ، يتم تطوير البرنامج بسرعة أكبر) ، فإن هذا يزيد من الجهد المطلوب للمشروع. يؤخذ هذا في الاعتبار من قبل SCED المضاعف في حساب تقدير الجهد.

افترض أن المشروع مقدّر TDEV13 شهراً ، كما هو مقترح أعلاه ، لكن الجدول الفعلي المطلوب كان 10 أشهر. يمثل هذا ضغط جدول بنسبة 25٪ تقريباً. استخدام قيم SCED المضاعف كما اشتق بواسطة فريق Boehm ، نرى أن مضاعف الجهد لهذا المستوى من ضغط الجدول هو 1.43. لذلك ، فإن الجهد الفعلي المطلوب إذا تم الوفاء بهذا الجدول الزمني المعجل يزيد بنسبة 50٪ تقريباً عن الجهد المطلوب لتسليم البرنامج وفقاً للجدول الاسمي.

هناك علاقة معقدة بين عدد الأشخاص الذين يعملون في المشروع ، والجهد الذي سيتم تكريسه للمشروع. والجدول الزمني لتسليم المشروع. إذا كان بإمكان أربعة أشخاص إكمال مشروع في 13 شهراً (أي 52 شخصاً - شهراً من الجهد) ، فقد تعتقد أنه من خلال إضافة شخص آخر ، يمكنك إكمال العمل في 11 شهراً (55 شهراً من الجهد للشخص). ومع ذلك ، يشير نموذج COCOMO إلى أنك ستحتاج ، في الواقع ، إلى ستة أشخاص لإنهاء العمل في 11 شهراً (66 شخصاً من الجهد).

والسبب في ذلك هو أن إضافة أشخاص إلى مشروع يقلل من إنتاجية أعضاء الفريق الحاليين. مع زيادة حجم فريق المشروع ، يقضي أعضاء الفريق وقتاً أطول في التواصل وتحديد الواجهات بين أجزاء النظام التي طورها أشخاص آخرون. وبالتالي ، فإن مضاعفة عدد الموظفين (على سبيل المثال) لا يعني أن مدة المشروع ستنخفض إلى النصف.

وبالتالي ،عندما تضيف شخصاً إضافياً ، فإن الزيادة الفعلية للجهد المضاف تكون أقل من شخص واحد لأن الآخرين يصبحون أقل إنتاجية. إذا كان فريق التطوير كبيراً ، فإن إضافة المزيد من الأشخاص إلى مشروع يؤدي أحياناً إلى زيادة جدول التطوير بدلاً من تقليله بسبب التأثير الكلي على الإنتاجية.

لايمكنك ببساطة تقدير عدد الأشخاص المطلوبين لفريق المشروع عن طريق قسمة الجهد الإجمالي على جدول المشروع المطلوب. عادة ، هناك حاجة إلى عدد قليل من الأشخاص في بداية المشروع لتنفيذ التصميم الأولي. الفريق بعد ذلك

يصلإلى الذروة أثناء تطوير النظام واختباره ، ثم ينخفض حجمه مع استعداد النظام للنشر. تبينأن التراكم السريع لموظفي المشروع يرتبط بالتأخر في الجدول الزمني للمشروع. بصفتك مديرمشروع ، يجب عليك بالتالي تجنب إضافة عدد كبير جداً من الموظفين إلى مشروع في وقتمبكر من عمره.

النقاطالرئيسية

- لايعتمد السعر الذي يتم تحصيله لنظام ما على تكاليف التطوير المقدرة والأرباح المطلوبة من قبل شركة التطوير. قدتعني العوامل التنظيمية زيادة السعر للتعويض عن زيادة المخاطر أو تقليله لاكتساب ميزة تنافسية.
- غالبأما يتم تسعير البرامج للحصول على عقد ، ومن ثم يتم تعديل وظيفة النظام لتلبية السعر المقدر.
- يتمتنظيم التطوير المستند إلى الخطة حول خطة مشروع كاملة تحدد أنشطة المشروع والجهد المخطط وجدول النشاطوالمسؤول عن كل نشاط.
- تتضمنجدولة المشروع إنشاء عروض بيانية مختلفة لجزء من خطة المشروع. المخططات الشريطية ، التي تظُهرمدة النشاط والجداول الزمنية للتوظيف ، هي أكثر تمثيلات الجدول الزمني استخداماً.
- معلمالمشروع هو نتيجة يمكن التنبؤ بها لنشاط أو مجموعة من الأنشطة. في كل مرحلة ، يجب تقديم تقرير رسميعن التقدم المحرز إلى الإدارة. التسليم هو منتج عمل يتم تسليمه إلى عميل المشروع.
- تتضمنلعبة التخطيط الرشيقة الفريق بأكمله في تخطيط المشروع. يتم تطوير الخطة بشكل تدريجي ، وفي حالةظهور مشكلات ، يتم تعديلها بحيث يتم تقليل وظائف البرنامج بدلاً من تأخير تسليم الزيادة.
- قدتكون تقنيات تقدير البرامج قائمة على الخبرة ، حيث يحكم المديرون على الجهد المطلوب ، أو خوارزمية ، حيثيتم حساب الجهد المطلوب من معلمات المشروع المقدرة الأخرى.
- نموذجتكلفة COCOMO II هو نموذج تكلفة خوارزمي ناضج يأخذ المشروع والمنتج والأجهزة والسمات الشخصيةفي الاعتبار عند صياغة تقدير التكلفة.

قراءةمتعمقة

القراءةالإضافية المقترحة في الفصل 22 ذات صلة أيضاً بهذا الفصل.

" عشرة أخطاء في تقدير المشروع ". مقال عملي يناقش الصعوبات العملية لتقدير المشروع ويتحدى بعض الافتراضاتالأساسية في هذا المجال. (P. Armor، *بالاتصالات ACM*، 45 (11) ، نوفمبر 2002). http://dx.doi.org/10.1145/581571.581582

رشيقةالتقدير والتخطيط. هذا الكتاب هو وصف شامل للتخطيط القائم على القصة كما هو مستخدم في XP ، بالإضافةإلى الأساس المنطقي لاستخدام نهج رشيق لتخطيط المشروع. يتضمن الكتاب أيضاً مقدمة جيدة وعامة لقضاياتخطيط المشروع. (إم كوهن ، 2005 ، برنتيس هول).

"الإنجازات والتحديات في تقدير موارد البرامج المستندة إلى COCOMO." تقدم هذه المقالة تاريخاً لنماذج COCOMO وتأثيراتها على هذه النماذج ، وتناقش متغيرات هذه النماذج التي تم تطويرها. كما يحدد المزيد من التطوراتالمحتملة في نهج BW Boehm(.COCOMO و R. Valeridi ،*برامج IEEE*، 25 (5) ، سبتمبر / أكتوبر 2008) http://dx.doi.org/10.1109/MS.2008.133.

كلشيء عن Agile. رشيقة التخطيط.يتضمن موقع الويب هذا حول الأساليب الرشيقة مجموعة ممتازة من المقالات حول التخطيطالسريع من عدد من المؤلفين المختلفين. (2007-2012). agile-Planning / category / http://www.allaboutagile.com /

معرفةإدارة المشروع: تخطيط المشروع.يحتوي هذا الموقع على عدد من المقالات المفيدة حول إدارة المشاريع بشكلعام. هذه تستهدف الأشخاص الذين ليس لديهم خبرة سابقة في هذا المجال. (2009-2014 ، P. Stoemmer). http://www.project-management-knowhow.com/project_ Planning.html

موقعالكتروني

شرائحPowerPoint لهذا الفصل:

www.pearsonglobaleditions.com/Sommerville

روابطلمقاطع الفيديو الداعمة:

http://software-engineering-book.com/videos/software-management/

E x E r C i SES

23.1.صف العوامل التي تؤثر على تسعير البرنامج. تحديد نهج "التسعير للفوز" في تسعير البرامج.

23.2.اشرح لماذا تكون عملية تخطيط المشروع متكررة ولماذا يجب مراجعة الخطة باستمرار أثناء مشروع البرنامج.

23.3.تحديد جدولة المشروع. ما هي الأشياء التي يجب مراعاتها عند تقدير الجداول؟

23.4.ما هي النمذجة الحسابية؟ ما هي المشاكل التي تعاني منها عند مقارنتها مع الأساليب الأخرى لتقدير التكلفة؟

23.5.يوضح الشكل 23.14 عدداً من المهام ومددها وتبعياتها. ارسم مخططاً شريطياً يوضح الجدول الزمني للمشروع.

مهمة	مدة	التبعيات
T1	10	
T2	15	T1
T3	10	T1 ، T2
T4	20	
T5	10	
T6	15	T3 ، T4
T7	20	T3
T8	35	T7
T9	15	T6
T10	5	T5 ، T9
T11	10	T9
T12	20	T10
T13	35	T3 ، T4
T14	10	T8 ، T9
T15	20	T12 ، T14
T16	10	T15

الشكل23.14
مثالعلى الجدولة

23.6.يوضح الشكل 23.14 مدد المهام لأنشطة مشروع البرمجيات. افترض حدوث نكسة خطيرة وغير متوقعة ، وبدلاً من أن تستغرق 10 أيام ، تستغرق المهمة T5 40 يوماً. ارسم مخططات شريطية جديدة توضح كيف يمكنإعادة تنظيم المشروع.

23.7.تعتمد لعبة التخطيط على فكرة التخطيط لتنفيذ القصص التي تمثل متطلبات النظام. اشرح المشاكل المحتملةمع هذا الأسلوب عندما يكون للبرنامج متطلبات عالية الأداء أو الاعتمادية.

23.8يتولى مدير البرنامج مسؤولية تطوير نظام برمجيات حرجة للسلامة ، وهو مصمم للتحكم في آلة العلاج الإشعاعيلعلاج المرضى الذين يعانون من السرطان. هذا النظام مضمن في الجهاز ويجب أن يعمل على معالجخاص الغرض مع مقدار ثابت من الذاكرة (256 ميغا بايت). يتواصل الجهاز مع نظام قاعدة بيانات المريضللحصول على تفاصيل المريض ، وبعد العلاج ، يسجل تلقائياً جرعة الإشعاع التي يتم تسليمها وتفاصيلالعلاج الأخرى في قاعدة البيانات.

يتماستخدام طريقة COCOMO لتقدير الجهد المطلوب لتطوير هذا النظام ، ويتم حساب تقدير 26 شخص -شهر. تم تعيين جميع مضاعفات محرك التكلفة على 1 عند إجراء هذا التقدير.

اشرحسبب تعديل هذا التقدير لأخذ عوامل المشروع والموظفين والمنتج والعوامل التنظيمية في الاعتبار.
اقترحأربعة عوامل قد يكون لها تأثيرات كبيرة على تقدير COCOMO الأولي واقترح القيم المحتملة لهذه العوامل.قم بتبرير سبب تضمين كل عامل.

23.9تتضمن بعض مشاريع البرامج الكبيرة جداً كتابة ملايين أسطر التعليمات البرمجية. اشرح لماذا قد لا تعمل نماذجتقدير الجهد ، مثل COCOMO ، بشكل جيد عند تطبيقها على أنظمة كبيرة جداً.

23.10.هل من الأخلاقي أن تقدم الشركة سعراً منخفضاً لعقد برنامج مع العلم أن المتطلبات غامضة ويمكنها فرض سعرمرتفع للتغييرات اللاحقة التي يطلبها العميل؟

المراجع

أبتس ،سي ، بي كلارك ، إس. ديفناني شولاني ، بي دبليو بوهم. 2000. "دليل تعريف نموذج COCOMO II." مركز هندسةالبرمجيات ، جامعة جنوب كاليفورنيا. CII_modelman2000.0.pdf / cocomo2000.0 / COCOMOII / http://csse.usc.edu/csse/ research

بيك ،ك ، وسي أندريس. 2004.*شرح البرمجة المتطرفة: الطبعة الثانية*. بوسطن: أديسون ويسلي.

بوم ،ب ، ب. كلارك ، إ. هورويتز ، سي ويستلاند ، ر. ماداتشي ، ور. سيلبي. 1995. "نماذج التكلفة لعمليات دورةحياة البرامج المستقبلية: COCOMO 2."*حوليات هندسة البرمجيات*: 31-1. دوى: 10.1007 / BF02249046.

بوهم ،ب ، و دبليو رويس. 1989. "Ada COCOMO ونموذج عملية Ada." في*بروك. الاجتماع الخامس لمجموعة مستخدمي COCOMO*. بيتسبرغ: معهد هندسة البرمجيات. a243476.pdf / u2 / fulltext /dtic/tr/ http://www.dtic.mil

بوم ،BW 1981.*اقتصاديات هندسة البرمجيات*. إنجليوود كليفس ، نيوجيرسي: برنتيس هول.

C. Abts، AW Brown، S. Chulani، B K. Clark، E. Horowitz، R. Madachy، D. Reifer، and B. Steece. 2000. Boehm، BW،*تقدير تكلفة البرمجيات مع COCOMO II*. إنجليوود كليفس ، نيوجيرسي: برنتيس هول.

كوهن ،م 2005.*رشيقة التقدير والتخطيط*. إنجليوود كليفس ، نيوجيرسي: برنتيس هول.

"جدول لغات النقطة الوظيفية". http://www.qsm.com/resources/function-pointlanguages-table QSM. 2014.

روبن ،كانساس 2013.*أساسي سكرم*. بوسطن: أديسون ويسلي.

إدارةality

أهداف

أهدافهذا الفصل هي تقديم إدارة جودة البرمجيات وقياس البرمجيات. عندما تقرأالفصل ، سوف:

- تمتعريفهم بعملية إدارة الجودة ومعرفة سبب أهمية تخطيط الجودة ؛
- إدراكأهمية المعايير في عملية إدارة الجودة ومعرفة كيفية استخدام المعايير فيضمان الجودة ؛
- فهمكيفية استخدام المراجعات والتفتيش كآلية لضمان جودة البرامج ؛
- فهمكيفية استناد إدارة الجودة في الأساليب الرشيقة إلى تطوير ثقافة جودة الفريق ؛
- فهمكيف قد يكون القياس مفيداً في تقييم بعض سمات جودة البرامج ، ومفهومتحليلات البرامج ، والقيود المفروضة على قياس البرامج.

محتويات

تهتمإدارة جودة البرامج بضمان أن أنظمة البرامج المطورة "مناسبة للغرض". بمعنى ، يجب أنتلبي الأنظمة احتياجات مستخدميها ، ويجب أن تعمل بكفاءة وموثوقية ، ويجب تسليمها فيالوقت المحدد وفي حدود الميزانية. أدى استخدام تقنيات إدارة الجودة جنباً إلى جنب مع تقنياتالبرامج الجديدة وطرق الاختبار إلى تحسينات كبيرة في مستوى جودة البرامج على مدارالعشرين عاماً الماضية.

تعتبرإدارة الجودة الرسمية (QM) مهمة بشكل خاص في الفرق التي تقوم بتطوير أنظمة كبيرةوطويلة العمر والتي تستغرق عدة سنوات لتطويرها. تم تطوير هذه الأنظمة للعملاء الخارجيين ،وعادةً ما يتم ذلك باستخدام عملية قائمة على الخطة. بالنسبة لهذه الأنظمة ، تعتبرإدارة الجودة مسألة تنظيمية وفردية في نفس الوقت:

1. على المستوى التنظيمي ، تهتم إدارة الجودة بإنشاء إطار للعمليات التنظيمية والمعايير التي منشأنها أن تؤدي إلى برامج عالية الجودة. يجب أن يتحمل فريق إدارة الجودة مسؤولية تحديدعمليات تطوير البرامج التي سيتم استخدامها والمعايير التي يجب تطبيقها على البرامجوالوثائق ذات الصلة ، بما في ذلك متطلبات النظام والتصميم والرمز.

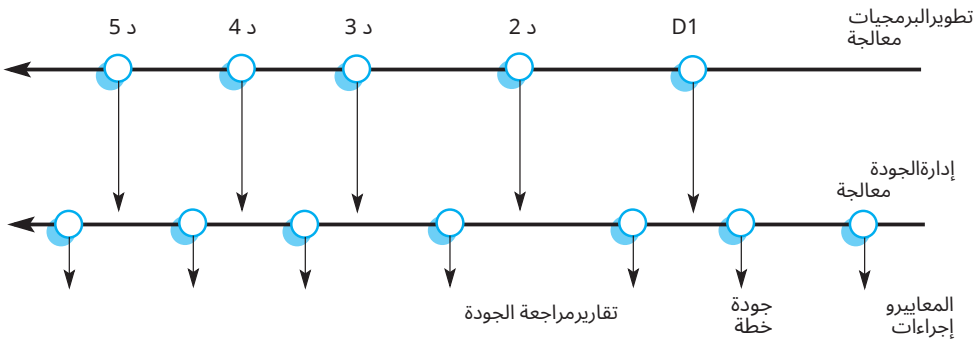
2. على مستوى المشروع ، تتضمن إدارة الجودة تطبيق عمليات جودة محددة ، والتحقق من اتباعهذه العمليات المخططة ، والتأكد من أن مخرجات المشروع تلبي معايير المشروع المحددة.قد تتضمن إدارة جودة المشروع أيضاً تحديد خطة جودة للمشروع. يجب أن تحددخطة الجودة أهداف الجودة للمشروع وأن تحدد العمليات والمعايير التي يجب استخدامها.

تقنياتإدارة جودة البرمجيات لها جذورها في الأساليب والتقنيات التي تم تطويرها في الصناعاتالتحويلية ، حيث الشروط*تاكيد الجودة*و *رقابة جودة*تستخدم على نطاق واسع. ضمانالجودة هو تعريف العمليات والمعايير التي يجب أن تؤدي إلى منتجات عالية الجودة وإدخالعمليات الجودة في عملية التصنيع. مراقبة الجودة هي تطبيق عمليات الجودة هذه للتخلصمن المنتجات التي ليست بمستوى الجودة المطلوب. يعد كل من ضمان الجودة ومراقبةالجودة جزءاً من إدارة الجودة.

فيصناعة البرمجيات ، ترى بعض الشركات أن ضمان الجودة هو تعريف للإجراءات والعملياتوالمعايير لضمان تحقيق جودة البرامج. في الشركات الأخرى ، يشمل ضمان الجودةأيضاً جميع أنشطة إدارة التكوين والتحقق والتحقق من الصحة التي يتم تطبيقها بعد تسليمأحد المنتجات بواسطة فريق التطوير.

توفرإدارة الجودة فحصاً مستقلاً لعملية تطوير البرامج. يتحقق فريق إدارة الجودة من مخرجاتالمشروع للتأكد من توافقها مع المعايير والأهداف التنظيمية (الشكل 24.1). كما يقومونبفحص وثائق العملية ، والتي تسجل المهام التي تم إكمالها من قبل كل فريق يعمل فيهذا المشروع. يستخدم فريق إدارة الجودة الوثائق للتحقق من عدم نسيان المهام المهمة أوأن إحدى المجموعات لم تضع افتراضات غير صحيحة حول ما قامت به المجموعات الأخرى.

عادةً ما يكون فريق إدارة الجودة في الشركات الكبيرة مسؤولاً عن إدارة عملية اختبار الإصدار.كما ناقشت في الفصل الثامن ، هذا يعني أنهم يديرون اختبار البرنامج قبل إصداره للعملاء.بالإضافة إلى ذلك ، فهم مسؤولون



الشكل24.1جودة الإدارةو تطويرالبرمجيات

للتحققمن أن اختبارات النظام توفر تغطية للمتطلبات وأن السجلات المناسبة لعملية الاختباريتم الاحتفاظ بها.

يجبأن يكون فريق إدارة الجودة مستقلاً وليس جزءاً من مجموعة تطوير البرامج حتى يتمكنوامن إلقاء نظرة موضوعية على جودة البرنامج. يمكنهم الإبلاغ عن جودة البرامج دون أن يتأثروابقضايا تطوير البرمجيات. من الناحية المثالية ، يجب أن يتحمل فريق إدارة الجودة مسؤوليةإدارة الجودة على مستوى المؤسسة. يجب أن يقدموا تقارير إلى الإدارة فوق مستوى مديرالمشروع.

نظراًلأنه يتعين على مديري المشاريع الحفاظ على ميزانية المشروع والجدول الزمني ، فقد يميلونإلى التنازل عن جودة المنتج للوفاء بهذا الجدول الزمني. يضمن فريق إدارة الجودة المستقلأن الأهداف التنظيمية للجودة لا تتأثر بالميزانية قصيرة الأجل واعتبارات الجدول الزمني.في الشركات الصغيرة ، هذا مستحيل عمليا. تتشابك إدارة الجودة وتطوير البرامج بشكلحتمي مع الأشخاص الذين يتحملون مسؤوليات التطوير والجودة.

تخطيطالجودة الرسمي هو جزء لا يتجزأ من عمليات التنمية القائمة على الخطة. إنها عمليةتطوير خطة الجودة لمشروع ما. يجب أن تحدد خطة الجودة صفات البرامج المرغوبة وتصفكيفية تقييم هذه الصفات. إنه يحدد ما تعنيه البرامج "عالية الجودة" في الواقع لنظام معين.لذلك ، يمتلك المهندسون فهماً مشتركاً لأهم سمات جودة البرامج.

همفري(همفري 1989) ، في كتابه الكلاسيكي عن إدارة البرمجيات ، يقترح هيكلاً تفصيلياً لخطةالجودة. يتضمن هذا المخطط التفصيلي ما يلي:

1. *مقدمة المنتج*وصف المنتج والسوق المقصود وتوقعات جودة المنتج.

2. *خطط المنتج*تواريخ الإصدار والمسؤوليات الهامة للمنتج ، إلى جانب خطط التوزيع وخدمة المنتج.

3. *أوصاف العملية*عمليات ومعايير التطوير والخدمة التي يجب استخدامها لتطوير المنتج وإدارته.

4. *أهداف الجودة*أهداف الجودة وخطط المنتج ، بما في ذلك تحديد وتبرير سمات جودة المنتج المهمة.

5. *المخاطر وإدارة المخاطر*المخاطر الرئيسية التي قد تؤثر على جودة المنتج والإجراءات الواجباتخاذها لمواجهة هذه المخاطر.

تختلف خطط الجودة ، التي يتم تطويرها كجزء من عملية التخطيط العام للمشروع ، في التفاصيل اعتماداً على حجم ونوع النظام الذي يتم تطويره. ومع ذلك ، عند كتابة خطط الجودة ، يجب أن تحاول جعلها قصيرة قدر الإمكان. إذا كان المستند طويلاً جداً ، فلن يقرأه الأشخاص ، مما يؤدي إلى إفشال الغرض من إنتاج خطة الجودة.

إدارة الجودة التقليدية هي عملية رسمية تعتمد على الاحتفاظ بوثائق مكثفة حول الاختبار والتحقق من صحة النظام وكيفية اتباع العمليات. في هذا الصدد ، فإنه يتعارض تماماً مع التطوير السريع ، حيث يتمثل الهدف في قضاء أقل وقت ممكن في كتابة المستندات وإضفاء الطابع الرسمي على كيفية تنفيذ أعمال التطوير. لذلك ، يجب أن تتطور تقنيات إدارة الجودة عند استخدام الأساليب الرشيقة. أناقش إدارة الجودة والتطوير السريع في القسم 24.4.

24.1 جودة البرمجيات

أرست الصناعة التحويلية أساسيات إدارة الجودة في محاولة لتحسين جودة المنتجات التي تم تصنيعها. كجزء من هذا الجهد ، طورت الصناعة تعريفاً للجودة يستند إلى التوافق مع مواصفات المنتج التفصيلية. كان الافتراض الأساسي هو أنه يمكن تحديد المنتجات بالكامل ويمكن وضع إجراءات يمكن أن تتحقق من المنتج المصنّع مقابل مواصفاته. بالطبع ، لن تتوافق المنتجات أبداً مع المواصفات تماماً ، لذلك تم السماح ببعض التسامح. إذا كان المنتج "مناسباً تقريباً" ، فقد تم تصنيفه على أنه مقبول.

جودة البرمجيات لا يمكن مقارنتها مباشرة بالجودة في التصنيع. فكرة التفاوتات قابلة للتطبيق في الأنظمة التناظرية ولكنها لا تنطبق على البرامج. علاوة على ذلك ، غالباً ما يكون من المستحيل التوصل إلى نتيجة موضوعية حول ما إذا كان نظام البرنامج يفي بمواصفاته أم لا:

1. من الصعب كتابة متطلبات برامج كاملة لا لبس فيها. قد يفسر مطورو البرامج والعملاء المتطلبات بطرق مختلفة ، وقد يكون من المستحيل التوصل إلى اتفاق حول ما إذا كان البرنامج يتوافق مع مواصفاته أم لا.

2. عادة ما تدمج المواصفات متطلبات من عدة فئات من أصحاب المصلحة. هذه المتطلبات هي حتماً حل وسط وقد لا تشمل متطلبات جميع مجموعات أصحاب المصلحة. لذلك قد ينظر أصحاب المصلحة المستبعدون إلى النظام على أنه نظام ذو جودة رديئة ، على الرغم من أنه ينفذ المتطلبات المتفق عليها.

3. من المستحيل قياس خصائص جودة معينة (على سبيل المثال ، قابلية الصيانة) مباشرة ، وبالتالي لا يمكن تحديدها بطريقة لا لبس فيها. أناقش صعوبات القياس في القسم 24.4.

بسبب هذه المشاكل ، فإن تقييم جودة البرمجيات هو عملية ذاتية. يستخدم فريق إدارة الجودة حكمهم لتقرير ما إذا كان قد تم تحقيق مستوى مقبول من الجودة. يقررون ما إذا كان البرنامج مناسباً أم لا

أمان	القابليةللفهم	قابليةالتنقل
حماية	قابليةالاختبار	سهولةالاستخدام
مصداقية	القدرةعلى التكيف	إعادةالاستخدام
تكيف	نمطية	نجاعة
المتانة	تعقيد	قابليةالتعلم

الشكل24.2برمجة علامات الجودة

الغرضالمقصود منه. يتضمن هذا القرار الإجابة على أسئلة حول خصائص النظام. فمثلا:

1.هل تم اختبار البرنامج بشكل صحيح ، وهل تم إثبات أن جميع المتطلبات قد تم تنفيذها؟

2.هل يمكن الاعتماد على البرنامج بدرجة كافية ليتم استخدامه؟

3.هل أداء البرنامج مقبول للاستخدام العادي؟

4.هل البرنامج قابل للاستخدام؟

5.هل البرنامج جيد التنظيم ومفهوم؟

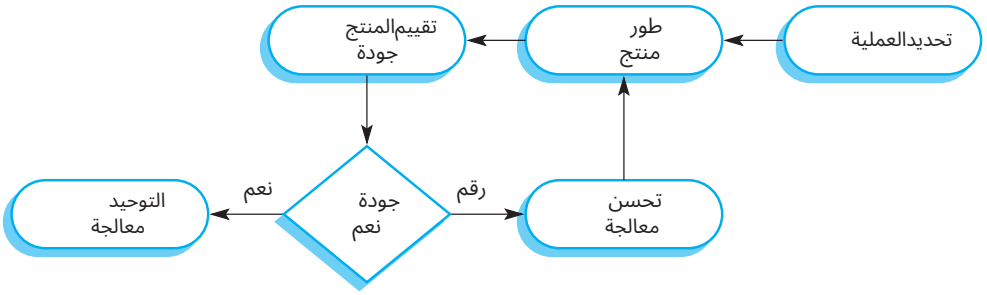
6.هل تم اتباع معايير البرمجة والتوثيق في عملية التطوير؟

هناكافتراض عام في إدارة جودة البرامج بأن النظام سيتم اختباره وفقاً لمتطلباته. يجب أن يعتمدالحكم على ما إذا كان يوفر الوظيفة المطلوبة أم لا على نتائج هذه الاختبارات. لذلك ، يجبعلى فريق إدارة الجودة مراجعة الاختبارات التي تم تطويرها وفحص سجلات الاختبار للتحققمن إجراء الاختبار بشكل صحيح. في بعض الشركات ، تجري مجموعة QM اختبار النظامالنهائي ؛ في حالات أخرى ، يقدم فريق اختبار نظام مخصص تقاريره إلى مدير جودة النظام.

تعتمدالجودة الذاتية لنظام البرمجيات إلى حد كبير على خصائصه غير الوظيفية. يعكس هذاتجربة المستخدم العملية - إذا لم تكن وظائف البرنامج كما هو متوقع ، فغالباً ما يعمل المستخدمونفقط على حل هذا النقص وإيجاد طرق أخرى للقيام بما يريدون القيام به. ومع ذلك ،إذا كان البرنامج غير موثوق به أو بطيئاً جداً ، فمن المستحيل عملياً بالنسبة لهم تحقيقأهدافهم.

لذلك ،لا تتعلق جودة البرامج فقط بما إذا كانت وظيفة البرنامج قد تم تنفيذها بشكل صحيح ،ولكنها تعتمد أيضاً على سمات النظام غير الوظيفية كما هو موضح في الشكل 24.2. تعكسهذه السمات موثوقية البرامج وقابليتها للاستخدام والكفاءة وقابلية الصيانة.

لايمكن تحسين أي نظام لكل هذه السمات. على سبيل المثال ، قد يؤدي تحسين الأمان إلىفقدان الأداء. لذلك يجب أن تحدد خطة الجودة أهم سمات الجودة للبرنامج الذي يتم تطويره.قد تكون الكفاءة أمراً بالغ الأهمية ويجب التضحية بعوامل أخرى لتحقيقها. إذا كنت قد أكدتعلى أهمية الكفاءة في خطة الجودة ، فيمكن للمهندسين العاملين في التطوير العمل معاًلتحقيق ذلك. يجب أن تتضمن الخطة أيضاً تعريفاً لعملية تقييم الجودة.



الشكل24.3معالجة-الجودةالقائمة

يجبأن تكون هذه العملية طريقة متفق عليها لتقييم ما إذا كانت بعض الجودة ، مثل قابلية الصيانةأو المتانة ، موجودة في المنتج.

تعتمدإدارة جودة البرامج التقليدية على افتراض أن جودة البرنامج مرتبطة مباشرة بجودة عمليةتطوير البرمجيات. يأتي هذا الافتراض من أنظمة التصنيع حيث ترتبط جودة المنتج ارتباطاًوثيقاً بعملية الإنتاج. تتضمن عملية التصنيع تكوين وإعداد وتشغيل الآلات المشاركة فيالعملية. بمجرد تشغيل الآلات بشكل صحيح ، تتبع جودة المنتج بشكل طبيعي. أنت تقيسجودة المنتج وتغير العملية حتى تصل إلى مستوى الجودة الذي تحتاجه. يوضح الشكل 24.3هذا النهج القائم على العملية لتحقيق جودة المنتج.

هناكارتباط واضح بين العملية وجودة المنتج في التصنيع لأن العملية سهلة نسبياً لتوحيدهاومراقبتها. بمجرد معايرة أنظمة التصنيع ، يمكن تشغيلها مراراً وتكراراً لإنتاج منتجاتعالية الجودة. ومع ذلك ، تم تصميم البرامج بدلاً من تصنيعها ، وتكون العلاقة بين جودةالعملية وجودة المنتج أكثر تعقيداً. تصميم البرمجيات هو عملية إبداعية ، لذا فإن تأثير المهاراتوالخبرات الفردية كبير. العوامل الخارجية ، مثل حداثة التطبيق أو الضغط التجاري للإصدارالمبكر للمنتج ، تؤثر أيضاً على جودة المنتج بغض النظر عن العملية المستخدمة.

لاشك أن عملية التطوير المستخدمة لها تأثير كبير على جودة البرنامج ، ومن المرجح أن تؤديالعمليات الجيدة إلى برامج عالية الجودة. يمكن أن تؤدي إدارة جودة العملية وتحسينها إلىعدد أقل من العيوب في البرامج التي يتم تطويرها. ومع ذلك ، من الصعب تقييم سمات جودةالبرامج ، مثل الموثوقية وقابلية الصيانة ، دون استخدام البرنامج لفترة طويلة. وبالتالي ، منالصعب معرفة كيف تؤثر خصائص العملية على هذه السمات. علاوة على ذلك ، بسبب دورالتصميم والإبداع في عملية البرمجيات ، يمكن أن يؤدي توحيد العمليات في بعض الأحيانإلى خنق الإبداع ، مما قد يؤدي إلى برامج ذات جودة أكثر فقراً بدلاً من برامج ذات جودةأفضل.

تعتبرالعمليات المحددة مهمة ، ولكن يجب أن يهدف مديرو الجودة أيضاً إلى تطوير "ثقافة الجودة" حيث يلتزم كل شخص مسؤول عن تطوير البرامج بتحقيق مستوى عالٍ من جودة المنتج.يجب عليهم تشجيع الفرق على تحمل مسؤولية جودة عملهم وتطوير أساليب جديدة لتحسينالجودة. في حين أن المعايير والإجراءات هي أساس إدارة الجودة ، يدرك مديرو الجودة الجيدةأن هناك جوانب غير ملموسة لجودة البرامج (الأناقة ، وسهولة القراءة ، وما إلى ذلك) لا يمكنتجسيدها في المعايير. يجب عليهم دعم الأشخاص المهتمين بالجوانب غير الملموسة للجودةوتشجيع السلوك المهني في جميع أعضاء الفريق.

معاييرالتوثيق

وثائقالمشروع هي طريقة ملموسة لوصف التمثيلات المختلفة لنظام برمجيات (المتطلبات ، UML ، الكود ، إلخ) وعملية الإنتاج الخاصة به.تحدد معايير التوثيق تنظيم أنواع مختلفة من المستندات بالإضافة إلى تنسيق المستند. إنها مهمة لأنها تجعل من السهل التحقق من عدمحذف المواد المهمة من المستندات والتأكد من أن مستندات المشروع لها "شكل وأسلوب" مشترك. يمكن وضع معايير لعملية كتابة الوثائق ،للوثائق نفسها ولتبادل الوثائق.

http://software-engineering-book.com/web/documentation-standards/

24.2 معايير البرمجيات

تلعبمعايير البرمجيات دوراً مهماً في إدارة جودة البرامج المستندة إلى الخطة. كما ناقشت ، يتمثلجزء مهم من ضمان الجودة في تعريف أو اختيار المعايير التي ينبغي تطبيقها على عمليةتطوير البرمجيات أو منتج البرنامج. كجزء من هذه العملية ، يمكن أيضاً اختيار الأدوات والطرقلدعم استخدام هذه المعايير. بمجرد اختيار المعايير للاستخدام ، يجب تحديد العمليات الخاصةبالمشروع لرصد استخدام المعايير والتحقق من اتباعها.

معاييرالبرمجيات مهمة لثلاثة أسباب:

1. تلتقط المعايير الحكمة ذات القيمة بالنسبة للمنظمة. إنها تستند إلى المعرفة حول أفضل الممارساتأو أنسبها للشركة. غالباً ما يتم اكتساب هذه المعرفة فقط بعد قدر كبير من التجربةوالخطأ. يساعد بناءها في معيار الشركة على إعادة استخدام هذه التجربة وتجنب الأخطاءالسابقة.
2. توفر المعايير إطاراً لتحديد ما تعنيه الجودة في بيئة معينة. كما ناقشت ، جودة البرامج ذاتية ،وباستخدام المعايير ، فإنك تضع أساساً لتقرير ما إذا كان قد تم تحقيق المستوى المطلوبمن الجودة. بالطبع ، يعتمد هذا على وضع معايير تعكس توقعات المستخدم فيمايتعلق بالاعتمادية على البرامج وقابليتها للاستخدام والأداء.
3. تساعد المعايير على الاستمرارية عندما يتولى شخص ما العمل ويواصله شخص آخر. تضمنالمعايير أن جميع المهندسين داخل المنظمة يتبنون نفس الممارسات. وبالتالي ، يتمتقليل جهد التعلم المطلوب عند بدء عمل جديد.

يمكنتحديد نوعين مرتبطين من معايير هندسة البرمجيات واستخدامهما في إدارة جودة البرامج:

1. *معايير المنتج*تنطبق هذه على منتج البرنامج الذي يتم تطويره. وهي تشمل معايير المستندات ،مثل هيكل مستندات المتطلبات ، ومعايير التوثيق ، مثل عنوان التعليق القياسيلتعريف فئة الكائن ، ومعايير الترميز ، التي تحدد كيفية استخدام لغة البرمجة.

معاييرالمنتج	معاييرالعملية
نموذجمراجعة التصميم	سلوكمراجعة التصميم
هيكلوثيقة المتطلبات	تقديمكود جديد لبناء النظام
طريقةتنسيق رأس	عمليةإصدار الإصدار
أسلوببرمجة جافا	عمليةالموافقة على خطة المشروع
تنسيقخطة المشروع	تغييرعملية التحكم
تغييراستمارة الطلب	اختبارعملية التسجيل

الشكل24.4منتج والعملية المعايير

2.*معايير العملية*تحدد هذه العمليات التي يجب اتباعها أثناء تطوير البرامج. يجب أن تشمل ممارساتالتنمية الجيدة. قد تتضمن معايير العملية تعريفات للمواصفات ، والتصميم ، وعملياتالتحقق من الصحة ، وأدوات دعم العمليات ، ووصفاً للمستندات التي يجب كتابتهاأثناء هذه العمليات.

أمثلةلمعايير المنتج والعملية التي يمكن استخدامها موضحة في الشكل 24.4.

يجبأن تقدم المعايير قيمة ، في شكل زيادة جودة المنتج. لا جدوى من تحديد معايير مكلفةمن حيث الوقت والجهد لتطبيقها والتي لا تؤدي إلا إلى تحسينات هامشية في الجودة. يجبتصميم معايير المنتج بحيث يمكن تطبيقها وفحصها بطريقة فعالة من حيث التكلفة ، ويجبأن تتضمن معايير العملية تعريف العمليات التي تتحقق من اتباع معايير المنتج.

عادةما يتم تكييف معايير هندسة البرمجيات المستخدمة داخل الشركة من المعايير الوطنيةأو الدولية الأوسع. تم تطوير المعايير الوطنية والدولية التي تغطي مصطلحات هندسة البرمجيات ،ولغات البرمجة مثل Java و C ++ ، والرموز مثل رموز الرسوم البيانية ، وإجراءات اشتقاقوكتابة متطلبات البرامج ، وإجراءات ضمان الجودة ، وعمليات التحقق من البرامج والتحققمنها (IEEE 2003). تم تطوير معايير أكثر تخصصاً لأنظمة السلامة والأمن الحرجة.

يعتبرمهندسو البرمجيات أحياناً أن المعايير وصفية أكثر من اللازم وغير ذات صلة بالنشاط الفنيلتطوير البرمجيات. هذا محتمل بشكل خاص عندما تتطلب معايير المشروع وثائق مملةوتسجيل العمل. على الرغم من أنهم يتفقون عادةً على الحاجة العامة للمعايير ، إلا أن المهندسينغالباً ما يجدون أسباباً وجيهة لعدم ملاءمة المعايير بالضرورة لمشروعهم الخاص. لذلكيجب على مديري الجودة الذين يضعون المعايير النظر في الإجراءات الممكنة لإقناع المهندسينبقيمة المعايير:

1.*إشراك مهندسي البرمجيات في اختيار معايير المنتج*إذا فهم المطورون سبب اختيار المعايير ،فمن المرجح أن يلتزموا بهذه المعايير. من الناحية المثالية ، يجب ألا تحدد وثيقةالمعايير المعيار الواجب اتباعه فحسب ، بل يجب أن تتضمن أيضاً تعليقاً يشرح سببهاتخاذ قرارات التقييس.

2.*مراجعة المعايير وتعديلها بانتظام لتعكس التقنيات المتغيرة* تعتبر المعايير مكلفة لتطويرها ،وهي تميل إلى أن تكون مكرسة في دليل معايير الشركة. بسبب التكاليف والمناقشةالمطلوبة ، غالباً ما يكون هناك إحجام عن تغييرها. دليل المعايير ضروري ، ولكنيجب أن يتطور ليعكس الظروف المتغيرة والتكنولوجيا.

3.*تأكد من أن دعم الأداة متاح لدعم التطوير المستند إلى المعايير* غالباً ما يجد المطورون المعاييرلتكون مصدر قلق عندما يتضمن التوافق معها عملاً يدوياً مملاً يمكن أن تقوم به أداةبرمجية. في حالة توفر دعم الأداة ، يمكن اتباع المعايير دون بذل الكثير من الجهد الإضافي.على سبيل المثال ، يمكن تحديد معايير تخطيط البرنامج وتنفيذها بواسطة نظامتحرير برنامج موجه نحو بناء الجملة.

تحتاجالأنواع المختلفة من البرامج إلى عمليات تطوير مختلفة ، لذلك يجب أن تكون المعاييرقابلة للتكيف. لا جدوى من وصف طريقة عمل معينة إذا كانت غير مناسبة لمشروع أوفريق المشروع. يجب أن يتمتع كل مدير مشروع بالسلطة لتعديل معايير العملية وفقاً للظروفالفردية. ومع ذلك ، عند إجراء التغييرات ، من المهم التأكد من أن هذه التغييرات لا تؤديإلى فقدان جودة المنتج.

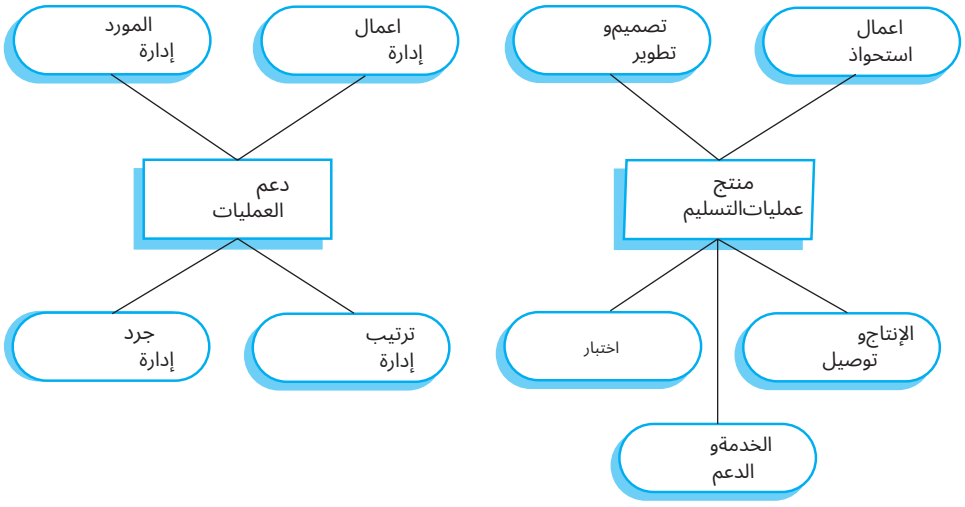
يمكنلمدير المشروع ومدير الجودة تجنب مشاكل المعايير غير الملائمة من خلال التخطيطالدقيق للجودة في وقت مبكر من المشروع. يجب أن يقرروا أي من المعايير التنظيميةيجب استخدامها دون تغيير ، وأيها يجب تعديله ، وأيها يجب تجاهله. قد يتعين إنشاءمعايير جديدة استجابة لمتطلبات العملاء أو المشروع. على سبيل المثال ، قد تكون معاييرالمواصفات الرسمية مطلوبة إذا لم يتم استخدام هذه المعايير في المشاريع السابقة.

24.2.1إطار معايير ISO 9001

مجموعةالمعايير الدولية المستخدمة في تطوير أنظمة إدارة الجودة في جميع الصناعات تسمىISO 9000. يمكن تطبيق معايير ISO 9000 على مجموعة من المنظمات من التصنيع إلىالصناعات الخدمية. تنطبق ISO 9001 ، وهي أكثر هذه المعايير عمومية ، على المؤسساتالتي تصمم المنتجات وتطورها وتحافظ عليها ، بما في ذلك البرامج. تم تطوير معيارISO 9001 في الأصل في عام 1987. أشرح هنا إصدار 2008 من المعيار ، ولكن قد يتغيرالمعيار في عام 2015 عندما تتم جدولة إصدار جديد للإصدار.

لايعد معيار ISO 9001 معياراً لتطوير البرامج ولكنه يمثل إطاراً لتطوير معايير البرامج. ويحددمبادئ الجودة العامة ، ويصف عمليات الجودة بشكل عام ، ويحدد المعايير والإجراءات التنظيميةالتي ينبغي تحديدها. يجب توثيقها في دليل الجودة التنظيمية.

أعادتمراجعة رئيسية لمعيار ISO 9001 في عام 2000 توجيه المعيار حول تسع عمليات أساسية(الشكل 24.5). إذا كانت المنظمة تريد أن تكون متوافقة مع ISO 9001 ، فيجب عليهاتوثيق كيفية ارتباط عملياتها بهذه العمليات الأساسية. يجب أيضاً أن تحدد وتحافظ على السجلاتالتي توضح أن العمليات التنظيمية المحددة لها



الشكل 24.5 العمليات الأساسية ISO 9001

تم اتباعه. يجب أن يصف دليل جودة الشركة العمليات ذات الصلة وبيانات العملية التي يجب جمعها وصيانتها.

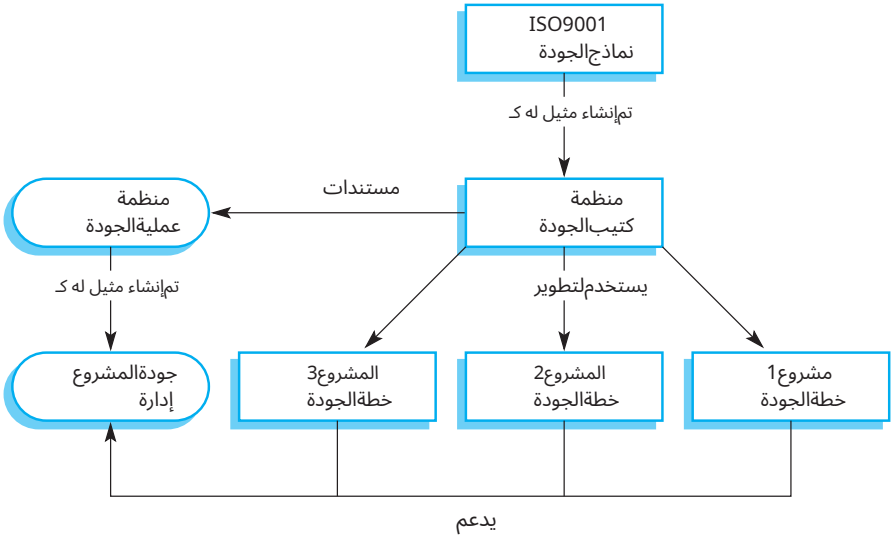
لا يحدد معيار ISO 9001 أو يصف عمليات الجودة المحددة التي يجب على الشركة استخدامها. لكي تتوافق مع ISO 9001 ، يجب على الشركة تحديد أنواع العمليات الموضحة في الشكل 24.5 وأن يكون لديها إجراءات مطبقة توضح أن عمليات الجودة الخاصة بها يتم اتباعها. هذا يسمح بالمرونة عبر القطاعات الصناعية وأحجام الشركات.

يمكن تحديد معايير الجودة المناسبة لنوع البرامج التي يتم تطويرها. يمكن أن يكون للشركات الصغيرة عمليات بسيطة بدون الكثير من الوثائق وأن تظل متوافقة مع ISO 9001. ومع ذلك ، فإن هذه المرونة تعني أنه لا يمكنك وضع افتراضات حول أوجه التشابه أو الاختلافات بين العمليات في مختلف الشركات المتوافقة مع ISO 9001. قد يكون لدى بعض الشركات عمليات جودة صارمة للغاية تحتفظ بسجلات مفصلة بينما قد يكون البعض الآخر أقل رسمية ، مع الحد الأدنى من الوثائق الإضافية.

يوضح الشكل 24.6 العلاقات بين ISO 9001 وأدلة الجودة التنظيمية وخطط جودة المشاريع الفردية. تم تكييف هذا الرسم البياني من نموذج قدمه Ince (Ince 1994) ، الذي يشرح كيف يمكن استخدام معيار ISO 9001 العام كأساس لعمليات إدارة جودة البرامج. يشرح Bamford and Deibler (Bamford and Deibler 2003) كيف يمكن تطبيق معيار ISO 9001:2000 الأحدث في شركات البرمجيات.

يطلب بعض عملاء البرامج أن يكون موردوهم حاصلين على شهادة ISO 9001. يمكن للعملاء بعد ذلك أن يكونوا على ثقة من أن شركة تطوير البرمجيات لديها نظام إدارة جودة معتمد. تفحص سلطات الاعتماد المستقلة عمليات إدارة الجودة ووثائق العملية وتقرر ما إذا كانت هذه العمليات تغطي جميع المجالات المحددة في ISO 9001. إذا كان الأمر كذلك ، فإنها تشهد بأن عمليات جودة الشركة ، على النحو المحدد في دليل الجودة ، تتوافق مع معيار ISO9001 .

يعتقد بعض الناس خطأً أن شهادة ISO 9001 تعني أن جودة البرامج التي تنتجها الشركات المعتمدة ستكون دائماً أفضل من تلك التي تنتجها



الشكل24.6والجودة 9001 ISO
إدارة

شركاتغير مصدق عليها. يركز معيار ISO 9001 على ضمان أن لدى المنظمة إجراءات إدارة الجودةالمعمول بها وأنها تتبع هذه الإجراءات. ليس هناك ما يضمن أن الشركات الحاصلة علىشهادة ISO 9001 تستخدم أفضل ممارسات تطوير البرامج أو أن عملياتها تؤدي إلى برامجعالية الجودة.

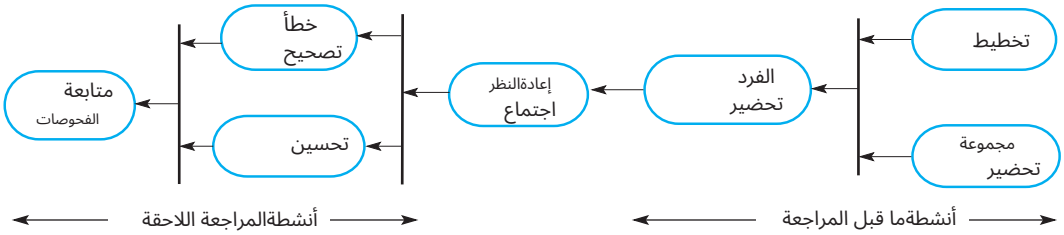
شهادة9001 ISO غير كافية ، في رأيي ، لأنها تحدد الجودة على أنها مطابقة للمعايير. لا يأخذفي الاعتبار الجودة التي يتمتع بها مستخدمو البرنامج. على سبيل المثال ، يمكن للشركة تحديدمعايير تغطية الاختبار التي تحدد أنه يجب استدعاء جميع الطرق في الكائنات مرة واحدةعلى الأقل. لسوء الحظ ، يمكن استيفاء هذا المعيار عن طريق اختبار البرامج غير المكتملالذي لا يتضمن اختبارات بمعلمات طريقة مختلفة. طالما يتم اتباع إجراءات الاختبار المحددةويتم الاحتفاظ بسجلات الاختبار ، يمكن أن تكون الشركة حاصلة على شهادة 9001 ISO.

24.3 مراجعات والتفتيش

المراجعاتوعمليات التفتيش هي أنشطة ضمان الجودة التي تتحقق من جودة مخرجات المشروع.يتضمن ذلك فحص البرنامج ووثائقه وسجلات العملية لاكتشاف الأخطاء والسهو بالإضافةإلى انتهاكات المعايير. كما أوضحت في الفصل الثامن ، يتم استخدام المراجعات وعمليةالتفتيش جنباً إلى جنب مع اختبار البرنامج كجزء من العملية العامة للتحقق من البرنامجوالتحقق من صحته.

أثناءالمراجعة ، يقوم العديد من الأشخاص بفحص البرنامج والوثائق المرتبطة به ، بحثاً عنالمشكلات المحتملة وعدم التوافق مع المعايير. يقوم فريق المراجعة بإصدار أحكام مستنيرةحول مستوى جودة البرنامج أو وثائق المشروع. يمكن لمديري المشاريع بعد ذلك استخدامهذه التقييمات لاتخاذ قرارات التخطيط وتخصيص الموارد لعملية التطوير.

تستندمراجعات الجودة إلى المستندات التي تم إنتاجها أثناء عملية تطوير البرامج. بالإضافةإلى مواصفات البرامج ، والتصاميم ، والكود ، ونماذج العمليات ، وخطط الاختبار ، وإجراءاتإدارة التكوين ، ومعايير العملية ،



الشكل24.7ال مراجعةالبرامج معالجة

وكتيباتالمستخدم قد تتم مراجعتها. يجب أن تتحقق المراجعة من اتساق واكتمال المستنداتأو الكود قيد المراجعة ، وإذا تم تحديد المعايير ، فتأكد من اتباع معايير الجودة هذه.

المراجعاتلا تتعلق فقط بفحص التوافق مع المعايير. يتم استخدامها أيضاً للمساعدة في اكتشافالمشكلات والسهو في البرنامج أو وثائق المشروع. يجب تسجيل استنتاجات المراجعةرسمياً كجزء من عملية إدارة الجودة. إذا تم اكتشاف مشاكل ، يجب إرسال تعليقات المراجعينإلى مؤلف البرنامج أو أي شخص مسؤول عن تصحيح الأخطاء أو السهو.

الغرضمن المراجعات والتفتيش هو تحسين جودة البرامج ، وليس تقييم أداء الأشخاص فيفريق التطوير. تعد المراجعة عملية عامة لاكتشاف الأخطاء ، مقارنة بعملية اختبار المكوناتالأكثر خصوصية. حتما ، يتم الكشف عن الأخطاء التي يرتكبها الأفراد لفريق البرمجة بأكمله.للتأكد من أن جميع المطورين يشاركون بشكل بناء في عملية المراجعة ، يجب أن يكون مديروالمشروع حساسين للمخاوف الفردية. يجب عليهم تطوير ثقافة عمل توفر الدعم دون لومعند اكتشاف الأخطاء.

مراجعاتالجودة ليست مراجعات تقدم الإدارة ، على الرغم من أنه يمكن استخدام المعلوماتحول جودة البرنامج في اتخاذ قرارات الإدارة. تقارن مراجعات التقدم التقدم الفعلي فيمشروع برمجي مع التقدم المخطط له. همهم الأساسي هو ما إذا كان المشروع سيقدم برامجمفيدة في الوقت المحدد وفي حدود الميزانية أم لا. تأخذ مراجعات التقدم العوامل الخارجيةفي الاعتبار ، وقد تعني الظروف المتغيرة أن البرنامج قيد التطوير لم يعد مطلوباً أو يجبتغييره بشكل جذري. قد يتعين إلغاء المشروعات التي طورت برامج عالية الجودة بسبب التغييراتالتي تطرأ على العمل أو بيئة التشغيل.

24.3.1عملية المراجعة

علىالرغم من وجود العديد من الاختلافات في تفاصيل المراجعات ، فإن عمليات المراجعة (الشكل24.7) مقسمة إلى ثلاث مراحل:

1. *أنشطة ما قبل المراجعة*هذه أنشطة تحضيرية ضرورية لكي تكون المراجعة فعالة. عادة ، أنشطةما قبل المراجعة معنية بتخطيط المراجعة وإعداد المراجعة. يتضمن تخطيط المراجعةتشكيل فريق للمراجعة ، وترتيب الوقت والمكان للمراجعة ، وتوزيع المستندات المرادمراجعتها. أثناء إعداد المراجعة ، قد يجتمع الفريق للحصول على نظرة عامة على البرنامجالمراد مراجعته. يقوم أعضاء فريق المراجعة الفردي بقراءة وفهم البرامج أو المستنداتوالمعايير ذات الصلة.

الأدوارفي عملية التفتيش

عندماتم إنشاء فحص البرنامج في شركة IBM (فاجان ، 1986) ، تم تحديد عدد من الأدوار الرسمية لأعضاء فريق التفتيش. وشملت هذه الوسيطوقارئ الشفرة والناسخ. قام مستخدمو عمليات التفتيش الآخرون بتعديل هذه الأدوار ، ولكن من المقبول عموماً أن يشمل التفتيش مؤلفالكود ومفتشاً وكاتباً ويجب أن يرأسه وسيط.

http://software-engineering-book.com/web/qm-roles

إنهميعملون بشكل مستقل للعثور على الأخطاء والسهو والحيد عن المعايير. يجوز للمراجعينتقديم تعليقات مكتوبة على البرنامج إذا لم يتمكنوا من حضور اجتماع المراجعة.

2. *اجتماع المراجعة* أثناء اجتماع المراجعة ، يجب على مؤلف المستند أو البرنامج الذي تتم مراجعته "استعراض" المستند مع فريق المراجعة. يجب أن تكون المراجعة نفسها قصيرة نسبياً- ساعتان على الأكثر. يجب أن يرأس أحد أعضاء الفريق المراجعة ، بينما يجب أن يقومآخر بتسجيل جميع قرارات المراجعة والإجراءات التي يتعين اتخاذها. أثناء المراجعة ، يكونالرئيس مسؤولاً عن ضمان مراعاة جميع التعليقات المقدمة. يجب أن يوقع رئيسالمراجعة على سجل بالتعليقات والإجراءات المتفق عليها أثناء المراجعة.

3. *أنشطة المراجعة اللاحقة* بعد انتهاء اجتماع المراجعة ، يجب معالجة المشكلات والمشكلات التيأثيرت أثناء المراجعة. قد تتضمن الإجراءات إصلاح أخطاء البرامج أو إعادة هيكلة البرامجبحيث تتوافق مع معايير الجودة أو إعادة كتابة المستندات. في بعض الأحيان ، تكونالمشكلات التي يتم اكتشافها في مراجعة الجودة ضرورية أيضاً لتقرير ما إذا كان ينبغيتوفير المزيد من الموارد لتصحيحها. بعد إجراء التغييرات ، قد يتحقق رئيس المراجعةمن أن جميع تعليقات المراجعة قد تم أخذها في الاعتبار. في بعض الأحيان ، يلزمإجراء مراجعة أخرى للتحقق من أن التغييرات التي تم إجراؤها تغطي جميع تعليقات المراجعةالسابقة.

يجبأن يكون لفرق المراجعة عادةً مجموعة أساسية من ثلاثة إلى أربعة أشخاص يتم اختيارهمكمراجعين رئيسيين. يجب أن يكون أحد الأعضاء مصمماً ذا خبرة ويتولى مسؤولية اتخاذالقرارات الفنية المهمة. قد يدعو المراجعون الرئيسيون أعضاء المشروع الآخرين ، مثل مصمميالأنظمة الفرعية ذات الصلة ، للمساهمة في المراجعة. قد لا يشاركون في مراجعة الوثيقةبأكملها ولكن يجب التركيز على تلك الأقسام التي تؤثر على عملهم. بدلاً من ذلك ، قد يوزعفريق المراجعة الوثيقة ويطلب تعليقات مكتوبة من مجموعة واسعة من أعضاء المشروع. لايلزم مشاركة مدير المشروع في المراجعة ، ما لم يكن من المتوقع حدوث مشكلات تتطلب تغييراتفي خطة المشروع.

تفترضالعمليات المقترحة للمراجعات أن فريق المراجعة لديه اجتماع وجهاً لوجه لمناقشة البرامجأو المستندات التي يراجعونها. ومع ذلك ، غالباً ما يتم توزيع فرق المشروع الآن ، أحياناً عبرالبلدان أو القارات ، لذلك من غير العملي أن يلتقي أعضاء الفريق وجهاً لوجه. يمكن دعم المراجعةعن بعُد باستخدام المستندات المشتركة حيث يمكن لكل عضو من أعضاء فريق المراجعةالتعليق على المستند بتعليقاتهم. قد تكون الاجتماعات وجهاً لوجه مستحيلة

بسببجداول العمل أو حقيقة أن الناس يعملون في مناطق زمنية مختلفة. رئيس المراجعة مسؤولعن تنسيق التعليقات ومناقشة التغييرات بشكل فردي مع أعضاء فريق المراجعة.

24.3.2 عمليات التفتيش على البرنامج

عمليات التفتيش على البرنامج هي مراجعات الأقران حيث يتعاون أعضاء الفريق للعثور على الأخطاءفي البرنامج الذي يتم تطويره. كما ناقشت في الفصل الثامن ، قد تكون عمليات التفتيشجزءاً من عمليات التحقق من البرامج والتحقق منها. إنها تكمل الاختبار لأنها لا تتطلبتنفيذ البرنامج. يمكن التحقق من الإصدارات غير الكاملة من النظام ، ويمكن التحقق منالتمثيلات مثل نماذج UML. يمكن مراجعة اختبارات البرنامج. غالباً ما تجد مراجعات الاختبارمشكلات في الاختبارات وبالتالي تعمل على تحسين فعاليتها في اكتشاف أخطاء البرنامج.

تشملعمليات التفتيش على البرنامج أعضاء الفريق من خلفيات مختلفة الذين يقومون بإجراءمراجعة دقيقة سطراً بسطر للكود المصدري للبرنامج. يبحثون عن العيوب والمشاكل ويصفونهافي اجتماع التفتيش. قد تكون العيوب أخطاء منطقية أو شذوذ في الكود قد يشير إلىحالة خاطئة أو ميزات تم حذفها من الكود. يفحص فريق المراجعة نماذج التصميم أو رمز البرنامجبالتفصيل ويسلط الضوء على العيوب والمشاكل التي يجب إصلاحها.

أثناءالفحص ، غالباً ما يتم استخدام قائمة تحقق بأخطاء البرمجة الشائعة للتركيز على البحثعن الأخطاء. قد تستند قائمة التحقق هذه إلى أمثلة من الكتب أو من معرفة العيوب الشائعةفي مجال تطبيق معين. أنت تستخدم قوائم تحقق مختلفة للغات البرمجة المختلفةلأن لكل لغة أخطاء مميزة خاصة بها. قدم همفري (همفري ، 1989) ، في مناقشة شاملةلعمليات التفتيش ، عدداً من الأمثلة لقوائم التفتيش.

الفحوصاتالمحتملة التي يمكن إجراؤها أثناء عملية الفحص موضحة في الشكل 24.8. يجبعلى المنظمات تطوير قوائم التفتيش الخاصة بها بناءً على المعايير والممارسات المحلية.يجب تحديث قوائم المراجعة هذه بانتظام ، حيث يتم العثور على أنواع جديدة من العيوب.تختلف العناصر الموجودة في قائمة التحقق وفقاً للغة البرمجة نظراً لمستويات الفحصالمختلفة التي يمكن إجراؤها في وقت الترجمة. على سبيل المثال ، يتحقق مترجم Java من أن الوظائف لديها العدد الصحيح من المعلمات ؛ مترجم لغة سي لا يفعل ذلك.

وجدتالشركات التي تستخدم عمليات التفتيش أنها فعالة في اكتشاف الأخطاء. في العملالمبكر ، أفاد Fagan 1986) Fagan) أنه تم اكتشاف أكثر من 60٪ من الأخطاء في البرنامجباستخدام عمليات التفتيش غير الرسمية للبرنامج. يقارن McConnell 2004) McConnell) اختبار الوحدة ، حيث يبلغ معدل اكتشاف الخلل حوالي 25٪ ، مع عمليات التفتيش ، حيث كان معدل اكتشاف العيب 60٪. تم إجراء هذه المقارنات قبل اختبار آليواسع النطاق. لا نعرف كيف يمكن مقارنة عمليات التفتيش بهذا النهج.

علىالرغم من فعاليتها من حيث التكلفة المعلنة جيداً ، فإن العديد من شركات تطوير البرمجياتتتردد في استخدام عمليات التفتيش أو مراجعات الأقران. أحياناً لا يرغب مهندسو البرمجياتمن ذوي الخبرة في اختبار البرنامج في قبول حقيقة أن عمليات التفتيش يمكن أن تكونأكثر فعالية لاكتشاف العيوب من الاختبار. قد يكون المدراء مرتابين لأن عمليات التفتيشتتطلب تكاليف إضافية أثناء التصميم والتطوير. قد لا يرغبون في المخاطرة بعدم وجودوفورات مقابلة في تكاليف اختبار البرنامج.

فئةخطأ	فحصالتفتيش
أخطاءالبيانات	- هلتمت تهيئة جميع متغيرات البرنامج قبل استخدام قيمها؟ - هلتم تسمية كل الثوابت؟ - هليجب أن يكون الحد العلوي للمصفوفات مساوياً لحجم المصفوفة أو الحجم12؟ - إذاتم استخدام سلاسل الأحرف ، فهل تم تعيين محدد بشكل صريح؟ - هلهناك أي احتمال لتجاوز المخزن المؤقت؟
أخطاءالتحكم	- لكلعبارة شرطية ، هل الشرط صحيح؟ - هلكل حلقة من المؤكد أن تنتهي؟ - هلالعبارات المركبة موضوعة بين قوسين بشكل صحيح؟ - فيحالة كشوف الحساب ، هل يتم احتساب جميع الحالات الممكنة؟ - إذاكان الاستراحة مطلوباً بعد كل حالة في بيانات الحالة ، فهل تم تضمينها؟
أخطاءالإدخال / الإخراج	- هلكل متغيرات المدخلات مستخدمة؟ - هليتم تعيين قيمة لجميع متغيرات المخرجات قبل أن يتم إخراجها؟ - يمكنأن تسبب المدخلات غير المتوقعة الفساد؟
أخطاءالواجهة	- هلتحتوي جميع استدعاءات الوظائف والطريقة على العدد الصحيح من المعلمات؟ - هلتتطابق أنواع المعلمات الرسمية والفعلية؟ - هلالمعلمات بالترتيب الصحيح؟ - إذاوصلت المكونات إلى الذاكرة المشتركة ، فهل لديها نفس نموذج بنية الذاكرة المشتركة؟
أخطاءإدارة التخزين	- إذاتم تعديل هيكل مرتبط ، فهل أعيد تعيين جميع الروابط بشكل صحيح؟ - إذاتم استخدام التخزين الديناميكي ، فهل تم تخصيص المساحة بشكل صحيح؟ - هلالفضاء غير مخصص بشكل صريح بعد أن لم يعد مطلوباً؟
أخطاءإدارة الاستثناءات	- هلتم أخذ جميع حالات الخطأ المحتملة في الاعتبار؟

الشكل24.8ان
قائمةالتفتيش

24.4 إدارة الجودة والتطوير السريع

تركزالأساليب الرشيقة لهندسة البرمجيات على تطوير الكود. إنها تقلل من الوثائق والعملياتالتي لا تتعلق مباشرة بتطوير الكود وتؤكد على أهمية الاتصالات غير الرسمية بين أعضاءالفريق بدلاً من الاتصالات القائمة على وثائق المشروع. تعني الجودة ، في التطوير السريع ،أن جودة الكود وممارسات مثل إعادة البناء ، والتطوير المدفوع بالاختبار تستُخدم لضمانإنتاج كود عالي الجودة.

إدارةالجودة في التطوير السريع هي إدارة غير رسمية وليست قائمة على المستندات. يعتمدعلى إنشاء ثقافة الجودة ، حيث يشعر جميع أعضاء الفريق بالمسؤولية عن جودة البرامج واتخاذالإجراءات لضمان الحفاظ على الجودة. يعارض المجتمع الرشيق بشكل أساسي ما يعتبرهعبئاً بيروقراطياً للنهج المستندة إلى المعايير وعمليات الجودة على النحو المنصوص عليهفي ISO 9001. نادراً ما تهتم الشركات التي تستخدم أساليب التطوير الرشيقة بشهادة ISO9001.

في التطوير السريع ، تعتمد إدارة الجودة على الممارسات الجيدة المشتركة بدلاً من التوثيق الرسمي.بعض الأمثلة على هذه الممارسة الجيدة هي:

1. *تحقق قبل تسجيل الوصول*المبرمجون مسؤولون عن تنظيم مراجعات الكود الخاصة بهم معأعضاء الفريق الآخرين قبل إيداع الكود في نظام البناء.

2. *لا تكسر البناء أبداً* من غير المقبول أن يقوم أعضاء الفريق بالتحقق من التعليمات البرمجية التي تؤدي إلى فشل النظام ككل. لذلك ، يتعين على الأفراد اختبار تغييرات التعليمات البرمجية الخاصة بهم مقابل النظام بأكمله والتأكد من أن هذه الرموز تعمل كما هو متوقع. إذا تم كسر البناء ، فمن المتوقع أن يعطي الشخص المسؤول أولوية قصوى لإصلاح المشكلة.

3. *أصلح المشاكل عندما تراها* رمز النظام ينتمي إلى الفريق وليس للأفراد. لذلك ، إذا اكتشف المبرمج مشاكل أو غموض في التعليمات البرمجية التي طورها شخص آخر ، فيمكنه إصلاح هذه المشكلات مباشرةً بدلاً من إعادتها إلى المطور الأصلي.

نادراً ما تستخدم العمليات الرشيقة عمليات التفتيش أو المراجعة الرسمية. في Scrum ، يجتمع فريق التطوير بعد كل تكرار لمناقشة مشكلات الجودة ومشكلاتها. قد يقرر الفريق إجراء تغييرات على طريقة عملهم لتجنب ظهور أي مشاكل جودة. يمكن اتخاذ قرار جماعي للتركيز على إعادة البناء وتحسين الجودة أثناء العدو بدلاً من إضافة وظائف جديدة للنظام.

قد تكون مراجعات الكود مسؤولية الأفراد (تحقق قبل تسجيل الوصول) أو قد تعتمد على استخدام البرمجة الزوجية. كما ناقشت في الفصل 3 ، البرمجة الزوجية هي نهج يكون فيه شخصان مسؤولان عن تطوير الكود ويعملان معاً لتحقيق ذلك. لذلك ، تخضع التعليمات البرمجية التي طورها فرد للفحص والمراجعة باستمرار من قبل عضو آخر في الفريق. ينظر شخصان إلى كل سطر من التعليمات البرمجية ويتحققان منه قبل قبوله.

تؤدي البرمجة الزوجية إلى معرفة عميقة بالبرنامج ، حيث يتعين على كلا المبرمجين فهم البرنامج بالتفصيل لمواصلة التطوير. يصعب أحياناً تحقيق عمق المعرفة هذا في عمليات التفتيش الأخرى ، وبالتالي يمكن للبرمجة الزوجية العثور على أخطاء لا يمكن اكتشافها أحياناً في عمليات التفتيش الرسمية. ومع ذلك ، لا يمكن أن يكون الشخصان المعنيان موضوعيا مثل فريق التفتيش الخارجي بقدر ما يقومان بفحص عملهما. المشاكل المحتملة هي:

1. *سوء التفاهم المتبادل* قد يرتكب كلا العضوين نفس الخطأ في فهم متطلبات النظام. قد تعزز المناقشات هذه الأخطاء.

2. *سمعة الزوج* قد يحجم الأزواج عن البحث عن الأخطاء لأنهم لا يريدون إبطاء تقدم المشروع.

3. *علاقات العمل* من المحتمل أن تتأثر قدرة الزوج على اكتشاف العيوب من خلال علاقة العمل الوثيقة بينهما والتي غالباً ما تؤدي إلى الإحجام عن انتقاد شركاء العمل.

النهج غير الرسمي لإدارة الجودة المعتمد في الأساليب الرشيقة فعال بشكل خاص لتطوير منتجات البرمجيات حيث تتحكم الشركة التي تطور البرنامج أيضاً في مواصفاته. ليست هناك حاجة لتقديم تقارير الجودة إلى العملاء الخارجيين ، ولا داعي للتكامل مع فرق إدارة الجودة الأخرى. ومع ذلك ، عندما يتم تطوير نظام كبير لملف

العميلالخارجي ، قد تكون الأساليب الرشيقة لإدارة الجودة مع الحد الأدنى من الوثائق غير عملية:

1.إذا كان العميل شركة كبيرة ، فقد يكون لديها عمليات إدارة الجودة الخاصة بها وقد تتوقع منشركة تطوير البرمجيات أن تقدم تقريراً عن التقدم المحرز بطريقة تتوافق مع هذه العمليات.لذلك ، قد يضطر فريق التطوير إلى إعداد خطة جودة رسمية وتقارير جودة حسبطلب العميل.

2.عندما تشارك عدة فرق موزعة جغرافياً في التنمية ، ربما من شركات مختلفة ، فإن الاتصالاتغير الرسمية قد تكون غير عملية. قد يكون لدى الشركات المختلفة مناهج مختلفةلإدارة الجودة ، وقد تضطر إلى الموافقة على إنتاج بعض الوثائق الرسمية.

3.بالنسبة للأنظمة طويلة العمر ، سيتغير الفريق المشارك في التطوير بمرور الوقت. في حالة عدموجود وثائق ، قد يجد أعضاء الفريق الجدد أنه من المستحيل فهم سبب اتخاذ قراراتالتطوير.

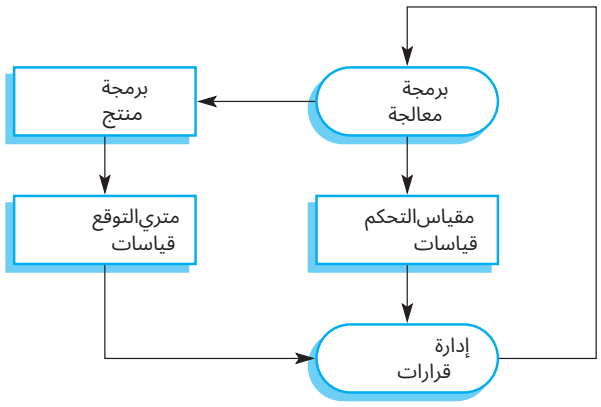
وبالتالي ،قد يتعين تكييف النهج غير الرسمي لإدارة الجودة في الأساليب الرشيقة بحيث يتمتقديم بعض الوثائق والعمليات عالية الجودة. بشكل عام ، تم دمج هذا النهج مع عملية التطويرالتكراري. بدلاً من تطوير البرامج ، يجب أن تركز إحدى سباقات السرعة أو التكرارات علىإنتاج وثائق البرامج الأساسية.

24.5 قياس البرمجيات

يهتمقياس البرامج بتحديد بعض سمات نظام برمجي مثل مدى تعقيده أو موثوقيته. من خلالمقارنة القيم المقاسة ببعضها البعض وبالمعايير المطبقة عبر المؤسسة ، قد تتمكن من استخلاصاستنتاجات حول جودة البرامج أو تقييم فعالية عمليات وأدوات وطرق البرامج. في عالممثالي ، يمكن أن تعتمد إدارة الجودة على قياسات السمات التي تؤثر على جودة البرنامج. يمكنكبعد ذلك إجراء تقييم موضوعي لتغييرات العمليات والأدوات التي تهدف إلى تحسين جودةالبرامج.

علىسبيل المثال ، لنفترض أنك تعمل في شركة تخطط لتقديم أداة اختبار برمجية جديدة. قبلتقديم الأداة ، تقوم بتسجيل عدد عيوب البرامج المكتشفة في وقت معين. هذا هو الأساسلتقييم فعالية الأداة. بعد استخدام الأداة لبعض الوقت ، تكرر هذه العملية. إذا تم العثورعلى المزيد من العيوب في نفس الفترة الزمنية ، بعد تقديم الأداة ، فقد تقرر أنها توفر دعماًمفيداً لعملية التحقق من صحة البرنامج.

الهدفطويل المدى لقياس البرامج هو استخدام القياس لإصدار أحكام حول جودة البرنامج. منالناحية المثالية ، يمكن تقييم النظام باستخدام مجموعة من المقاييس لقياس سماته. من خلالالقياسات التي تم إجراؤها ، يمكن استنتاج قيمة جودة النظام. إذا وصل البرنامج إلى حد الجودةالمطلوب ، فيمكن الموافقة عليه دون مراجعة. عند الاقتضاء ، قد تبرز أدوات القياس أيضاًمجالات البرنامج التي يمكن تحسينها.



الشكل 24.9 المتنبئ والسيطرة قياسات

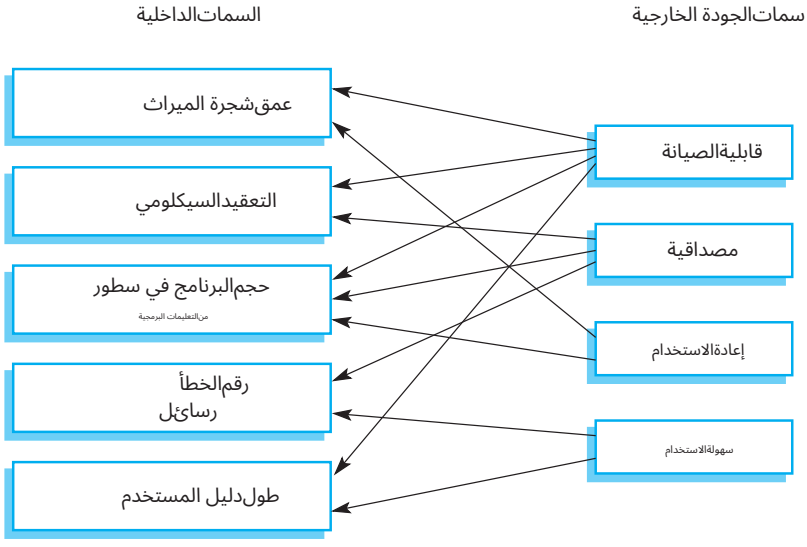
ومعذلك ، ما زلنا بعيدين عن هذا الوضع المثالي ، ومن غير المرجح أن يصبح تقييم الجودة الآلي حقيقة واقعة في المستقبل القريب.

مقياس البرنامج هو سمة من سمات نظام البرنامج أو توثيق النظام أو عملية التطوير التي يمكن قياسها بشكل موضوعي. تتضمن أمثلة المقاييس حجم المنتج في سطور التعليمات البرمجية ،ومؤشر الضباب ، وهو مقياس لسهولة قراءة النص السردي ، وعدد الأخطاء المبلغ عنها في منتج البرنامج الذي تم تسليمه ، وعدد أيام الفرد المطلوبة للتطوير أحد مكونات النظام.

قد تكون مقاييس البرامج إما مقاييس تحكم أو مقاييس توقع. كما توحي الأسماء ، تدعم مقاييس التحكم إدارة العملية ، وتساعدك مقاييس التوقع على التنبؤ بخصائص البرنامج. عادة ما ترتبط مقاييس التحكم بعمليات البرامج. من أمثلة مقاييس التحكم أو العملية متوسط الجهد والوقت المطلوب لإصلاح العيوب المبلغ عنها. يمكن استخدام ثلاثة أنواع من مقاييس العملية:

1. *الوقت المستغرق لإكمال عملية معينة* يمكن أن يكون هذا هو إجمالي الوقت المخصص للعملية ،ووقت التقويم ، والوقت الذي يقضيه مهندسون معينون في العملية ، وما إلى ذلك.
2. *الموارد المطلوبة لعملية معينة* قد تتضمن الموارد مجهوداً إجمالياً في أيام الأشخاص أو تكاليف السفر أو موارد الكمبيوتر.
3. *عدد تكرارات حدث معين* تتضمن أمثلة الأحداث التي يمكن مراقبتها عدد العيوب التي تم اكتشافها أثناء فحص الكود ، وعدد تغييرات المتطلبات المطلوبة ، وعدد تقارير الأخطاء في النظام الذي تم تسليمه ، ومتوسط عدد سطور التعليمات البرمجية التي تم تعديلها استجابة لتغيير المتطلبات.

مقاييس التوقع (تسمى أحياناً مقاييس المنتج) مرتبطة بالبرنامج نفسه. من أمثلة مقاييس التوقع التعقيد الدوري للوحدة ، ومتوسط طول المعرفات في البرنامج ، وعدد السمات والعمليات المرتبطة بفئات الكائن في التصميم. قد يؤثر كل من مقاييس التحكم والتنبؤ على اتخاذ القرار الإداري كما هو موضح في الشكل 24.9. يستخدم المديرون قياسات العملية لتحديد ما إذا كان ينبغي إجراء تغييرات على العملية وتوقع المقاييس لتحديد ما إذا كانت تغييرات البرامج ضرورية وما إذا كان البرنامج جاهزاً للإصدار.



الشكل24.10
العلاقاتبين
داخليوخارجي
سماتالبرنامج

في هذا الفصل ، أركز على مقاييس التنبؤ ، التي يتم تقييم قيمها تلقائياً من خلال تحليل الكودأو المستندات. أناقش مقاييس التحكم وكيفية استخدامها في تحسين العملية في الفصل26 على الويب.

يمكنااستخدام قياسات نظام برمجي بطريقتين:

1. *لتعيين قيمة لسمات جودة النظام*من خلال قياس خصائص مكونات النظام ثم تجميع هذه القياسات ،قد تتمكن من تقييم سمات جودة النظام ، مثل قابلية الصيانة.

2. *التعرف على مكونات النظام التي تكون جودتها دون المستوى المطلوب*يمكن للقياسات تحديدالمكونات الفردية بخصائص تختلف عن القاعدة.
على سبيل المثال ، يمكنك قياس المكونات لاكتشاف العناصر الأكثر تعقيداً. من المرجح أن تحتوي هذه المكونات على أخطاء لأن التعقيد يزيد من احتمالية ارتكاب مطور المكون للأخطاء.

من الصعب إجراء قياسات مباشرة للعديد من سمات جودة البرامج الموضحة في الشكل 24.2.سمات الجودة مثل قابلية الصيانة والفهم وسهولة الاستخدام هي سمات خارجية تتعلق بكيفيةتجربة المطورين والمستخدمين للبرامج. فهي تتأثر بعوامل ذاتية ، مثل تجربة المستخدموالتعليم ، وبالتالي لا يمكن قياسها بشكل موضوعي. لإصدار حكم بشأن هذه السمات ،يجب عليك قياس بعض السمات الداخلية للبرنامج (مثل حجمه وتعقيده) وافتراض أن هذه السمات مرتبطة بخصائص الجودة التي تهمك.

يوضح الشكل 24.10 بعض سمات جودة البرامج الخارجية والسمات الداخلية التي يمكن أن تكون مرتبطة بها بشكل حدسي. يشير الرسم البياني إلى أنه قد تكون هناك علاقات بين السماتالخارجية والداخلية ، لكنه لا يوضح كيفية ارتباط هذه السمات. اقترح كيتشنهام (كيتشنهام1990) أنه إذا كان مقياس السمة الداخلية سيكون مؤشراً مفيداً لخاصية البرنامج الخارجي ،فيجب أن تتوفر ثلاثة شروط:

1. يجب قياس السمة الداخلية بدقة. ومع ذلك ، لا يكون القياس دائماً مباشراً وقد يتطلب أدواتمطورة خصيصاً.
2. يجب أن توجد علاقة بين السمة التي يمكن قياسها وسمة الجودة الخارجية ذات الأهمية. أيأن قيمة سمة الجودة يجب أن تكون مرتبطة ، بطريقة ما ، بقيمة السمة التي يمكن قياسها.
3. يجب فهم هذه العلاقة بين السمات الداخلية والخارجية والتحقق من صحتها والتعبير عنها منخلال صيغة أو نموذج. تتضمن صياغة النموذج تحديد الشكل الوظيفي للنموذج (خطي ،أسي ، إلخ.)
منخلال تحليل البيانات التي تم جمعها ، وتحديد المعلمات التي سيتم تضمينها في النموذجومعايرة هذه المعلمات باستخدام البيانات الموجودة.

استخدمتالأعمال الحديثة في مجال تحليلات البرامج (Zhang et al. 2013) تقنيات جمعالبيانات والتعلم الآلي لتحليل مستودعات منتجات البرمجيات وبيانات العملية. الفكرة وراءتحليلات البرامج (Menzies and Zimmermann 2013) هي أننا لا نحتاج ، في الواقع ، إلىنموذج يعكس العلاقات بين جودة البرامج والبيانات التي تم جمعها. بدلاً من ذلك ، إذا كانتهناك بيانات كافية ، فيمكن اكتشاف الارتباطات ووضع تنبؤات حول سمات البرنامج. أناقشتحليلات البرامج في القسم 24.5.4.

لديناالقليل جداً من المعلومات المنشورة حول القياس المنهجي للبرامج في الصناعة. تقومالعديد من الشركات بجمع معلومات حول برامجها ، مثل عدد طلبات تغيير المتطلبات أو عددالعيوب المكتشفة في الاختبار. ومع ذلك ، ليس من الواضح ما إذا كانوا يستخدمون هذه القياساتبعد ذلك بشكل منهجي لمقارنة منتجات وعمليات البرامج أو تقييم تأثير التغييرات علىعمليات وأدوات البرامج. هناك عدة أسباب وراء صعوبة ذلك:

1. من المستحيل تحديد عائد الاستثمار لإدخال مقاييس تنظيمية أو برنامج تحليلات برمجية. لقدشهدنا تحسينات كبيرة في جودة البرامج على مدى السنوات القليلة الماضية دون استخدامالمقاييس ، لذلك من الصعب تبرير التكاليف الأولية لإدخال قياس البرامج وتقييمهابشكل منهجي.

2. لا توجد معايير لمقاييس البرامج أو عمليات موحدة للقياس والتحليل. تحجم العديد من الشركاتعن تقديم برامج القياس حتى تتوفر هذه المعايير والأدوات الداعمة.

3. قد يتطلب القياس تطوير وصيانة أدوات برمجية متخصصة. من الصعب تبرير تكاليف تطويرالأداة عندما تكون عائدات القياس غير معروفة.

4. في العديد من الشركات ، لا تكون عمليات البرمجيات موحدة وسيئة التحديد والتحكم. علىهذا النحو ، هناك الكثير من التباين في العمليات داخل نفس الشركة لاستخدام القياساتبطريقة ذات مغزى.

5. ركز الكثير من الأبحاث حول قياس البرمجيات والمقاييس على المقاييس القائمة على الكودوعمليات التطوير التي تحركها الخطة. ومع ذلك ، يتم الآن تطوير المزيد والمزيد من البرامجعن طريق إعادة استخدام وتكوين التطبيق الحالي