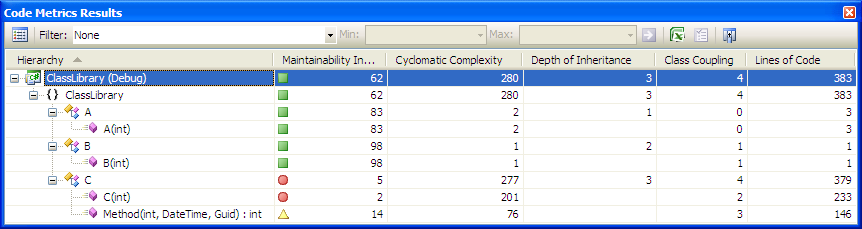
Средство оценки качества кода в MS Visual Studio

* [.NET](https://habr.com/ru/hub/net/)
* [Из песочницы](https://habr.com/ru/sandbox/)

Введение

Достаточно часто программистам приходится поддерживать чужой код, очень часто этот код выглядит не самым лучшим образом, и сопровождать его очень сложно. Если это приложение не придется выбросить в скором времени, то естественно его стоит привести в человеческий вид, т.е. отрефакторить. Было бы хорошо иметь какую-нибудь метрику, которая позволила бы оценить качество кода и определить места, которые стоит улучшить. Такая метрика позволила бы оценить, например, то, как программист пишет исходный код или то, насколько качественен код в том приложении, которое Вы собираетесь поддерживать.  
  
Microsoft предоставляет встроенное в Visual Studio средство, которое позволяет оценить код вашего проекта.  
  
  
  
Получить оценку вашего кода можно нажав правой кнопкой на проекте и выбрав пункт “Calculate Code Metrics” (Эта функциональность доступна в Visual Studio 2008 Team System и Visual Studio 2010 начиная с Premium версии).

Описание метрик

Результаты содержат 5 метрик для вашего кода.

1. **Maintainability Index** – комплексный показатель качества кода. Этот показатель разработан специалистами из Carnegie Mellon Software Engineering Institute. Рассчитывается метрика по следующей формуле:  
     
   MI = MAX(0, (171 — 5.2 \* ln(HV) — 0.23 \* CC — 16.2 \* ln(LoC)) \* 100 / 171)
   * HV – Halstead Volume, вычислительная сложность. Чем больше операторов, тем больше значение этой метрики;
   * CC – Cyclomatic Complexity. Эта метрика описана ниже;
   * LoC – количество строк кода.

Эта метрика может принимать значения от 0 до 100 и показывает относительную сложность поддержки кода. Чем больше значение этой метрики, тем легче поддерживать код.  
  
Visual Studio помечает методы/классы зеленым цветом, если значение метрики находится в пределах от 20 до 100, желтым цветом, если значение находится в пределах от 10 до 20, и красным цветом, когда значение меньше 10.

1. **Cyclomatic Complexity** – показывает структурную сложность кода, т.е. количество различных ветвей в коде. Чем больше этот показатель, тем больше тестов должно быть написано, для полного покрытия кода.
2. **Depth of Inheritance** – глубина наследования. Эта метрика показывает для каждого класса, какой он по счету в цепочке наследования. Например, есть 3 класса A, B, C, B унаследован от А, а С унаследован от В, то значение этой метрики для классов A,B и C будет равно соответственно 1, 2 и 3.
3. **Class Coupling** – показывает степень зависимости классов друг с другом. В расчет берутся уникальные классы из параметров, локальных переменных, возвращаемого типа, базового класса, атрибутов (полный список можно найти в [MSDN](http://msdn.microsoft.com/en-us/library/bb385914.aspx)). Хороший дизайн программного обеспечения предполагает небольшое количество связанных классов. Чем их больше, тем сложнее в дальнейшем переиспользовать этот класс, а также поддерживать, т.к. существует очень много зависимостей.
4. **Lines of Code** – показывает количество строк кода. Этот показатель показывает не точное количество строк в вашем файле, т.к. подсчет основан на IL-коде. В расчет не берутся пустые строчки, комментарии, строчки со скобками, объявление типов и пространств имен. Большое количество строк в методе/классе может показывать на ошибки в проектировании и на то, что этот код можно разделить на несколько частей.

Более подробно про то, как считаются эти метрики, можно прочитать [здесь](http://blogs.msdn.com/b/codeanalysis/archive/2007/10/03/new-for-visual-studio-2008-code-metrics.aspx).

Реальное использование

Когда я первый раз запустил анализ на одном из проектов, все значения Maintainability Index были зеленые. Это казалось несколько странным, т.к. там явно был код, который надо было бы переписать. Значения MI для таких участков кода были около 30-40. Получается, что показатели по умолчанию являются, скорее всего, субъективными, и решение о том, какой код считать некачественным, придется принимать самим программистам.  
  
Для своих проектов я стараюсь для большинства методов поддерживать показатель MI около 70-90. Бывают методы, у которых этот показатель равен 50-60, и их можно переписать, но стоит оценивать затраты и выгоды.  
  
Благодаря этому инструменту можно достаточно легко провести code review большого проекта, найти места, которые необходимо переписать. Также достаточно полезно следить за процессом изменения вышеописанных метрик. Это может показать руководителю об отношении программистов к разработке того или иного проекта, а также динамику изменения качества кода по каждому программисту, что немаловажно в нашей профессии. Другой причиной слежения за метриками, являются, определенные программистами, пороговые значения, при достижении которых, необходимо заняться рефакторингом.  
  
Спасибо большое.

Источники

1. [MSDN](http://msdn.microsoft.com/en-us/library/bb385914.aspx)
2. [PC Week](http://www.pcweek.ru/themes/detail.php?ID=83423)
3. [Code Analysis Team Blog](http://blogs.msdn.com/b/codeanalysis/archive/2007/10/03/new-for-visual-studio-2008-code-metrics.aspx)