1. Автоматизация процесса документирования

Наиболее полной и согласованной получается документация при автоматизации процесса документирования. *Автоматизированное документирование* — одна из самых важных сторон разработки программного кода. Идея автоматизированного документирования заключается в том, что в исходном коде пишутся комментарии в специальном формате. Далее запускается специальная утилита, которая разбирает исходный код, вытаскивая комментарии и объединяя их в единый структурированный документ. Таким образом, с плеч программиста снимается вся черновая работа по созданию программной документации. Кроме того, такой способ документирования позволяет более строго контролировать версии программного кода и документации, поскольку при изменении программного кода программист должен изменять расположенные рядом комментарии.

Для документирования программного кода используют *программы-генераторы*, сканирующие файлы проекта в поисках комментариев. Без комментариев даже небольшую программу трудно понять не только постороннему программисту, но и самому автору.

Многие среды разработки (Zend, Visual Studio, Netbeans) уже включают в себя средства документирования, но никакого общего стандарта до сих пор нет. В своём большинстве, цель генераторов — составить нечто типа энциклопедии всех функций, классов, объектов, иерархий классов, связей между файлами, используемые компоненты и третьи программные продукты.

Генераторы можно условно разделить на:

- Универсальные
  - — Doxygen
  - — Natural Docs На выходе — HTML.
  - — AsciiDoc (GNU GPLv2) На выходе — HTML(4,5), XHTML, DocBook, LaTeX, PDF, EPUB, DVI, PS, man, HTML Help, txt.
  - — Doc-O-Matic для C/C++, C#, Delphi, VB.NET, IDL, Java, PHP, JavaScript, ASPX, JSP, MatLab. Doc-O-Matic ships with link databases for Microsoft Visual Studio 2010, 2008, 2005, MSDN, Embarcadero RAD Studio XE, 2010 as well as previous CodeGear and Borland Delphi versions of Delphi and C++Builder. V7 — $49.

  Специфичные

  - — javadoc (Free) для Java.
  - — phpDocumentor (Open Source) для PHP.
  - — Docutils (Open Source) для Python.
  - — NDoc, Sandcastle, GhostDoc для .NET.
  - — PasDoc (GNU GPLv2) для Pascal (Object Pascal) На выходе — HTML, HtmlHelp, LaTeX, ...

Генераторы, как правило, по-своему хранят данные и комментарии, завися от языка, однако последние тенденции, например в Visual Studio, показывают популяризацию xml. DocBook — одна из предлагаемых универсальных схем.

2. Doxygen

Doxygen — мощная и распространённая утилита для документирования программного кода, написанного на C, C++, Objective-C, Java, Python, IDL (Corba и Microsoft) и некоторых версий PHP, C#, и D. Выходная документация представляет собой файлы в формате HTML, XML, LaTeX (в частности, для дальнейшего создания PDF-файлов), RTF, справочного руководства в стиле кроссплатформенной библиотеки Qt, справочного руководства в стиле Unix — man и т. д.

Даже, если не оформалять специальным образом исходные тексты, Doxygen сгенерирует из них замечательные справочники программного интрефейса приложения. В выходной документ могут быть включены разделы с описанием используемых пространств имён, классов, файлов и их указатели на соответствующие разделы. В случае использования пакета визуализации графов — Graphviz, описание классов может сопровождаться диаграммами UML.

Распространяется по лицензии GNU GPL.

Существует два вида комментариев:

Комментарии для автоматизированного документирования исходного кода. Эти комментарии оформляются с помощью специальных форматов (doxygen, javadoc, MS XML Documentation Comments).

Комментарии исходного кода — сопровождающие исходный код, объясняющие его работу, раскрывающие непонятные и сомнительные моменты.

В отличие от комментариев для автоматизированного документирования, эти комментарии используются в любом программном коде. *Стоимость владения* исходным кодом уменьшается прямо пропорционально количеству и качеству таких комментариев. Это связано с тем, что хорошо структурированный и комментированный код легче поддерживается и сопровождается. То есть программистам просто требуется меньше времени для того, чтобы разобраться с ним. Для анализа качества исходного кода можно использовать специализированные инструменты (например, together), которые подсчитывают различные метрики кода, описывающие его качество.

Doxygen позволяет из комментариев кода, записанных в специальном формате, генерировать документы HTML, LaTeX, RTF, PDF с ссылками, диаграммами классов и прочими удобствами.

В Doxygen существуют несколько вариантов комментариев для автоматизированного документирования:

1. блочный стиль JavaDoc

```
/**
комментарий
*/
```

2. блочный стиль Qt

```
/*!
комментарий
*/
```

3. строчный стиль C++

```
/// комментарий
```

4. строчный стиль C++

```
//! комментарий
```

Всё, что написано внутри комментариев, оформленных таким образом, выносится в документацию. Для единообразия лучше использовать `///` для коротких комментариев и `/** */` для длинных.

Служебные слова

В Doxygen есть служебные слова, которые задают правила отображения частей документации (выделение в отдельный раздел, специальное форматирование и шрифт для выделения примера кода, вынесение текста на заглавную страницу и т.п.).

Автор — `@author` *имя*

Текст на главной странице — `@mainpage`

Создание новой страницы — `@page` *идентификатор название*

Создание раздела — `@section` *идентификатор название*

Вставка примера кода — `@code` текст программы `@endcode`

```
@page pagereq Требования к системе
@section common Общие требования
```

Заметка — `@note`

Предупреждение — `@warning`

Описание к файлу — `@file`

Краткое описание — `@brief` Можно настроить файл опций таким образом, чтобы в краткое описание выносилось первое предложение до точки подробного описания (по умолчанию описание считается подробным). Для этого нужно включить опцию JAVADOC_AUTOBRIEF — в утилите `doxywizard` она находится на первой странице.

Подробное описание — `@full`

Именованная группа — `@name` *имя*. После этого группа в скобках @{ ... @}

Внутри парного тега `\f` можно задавать формулу в нотации LaTeX. Этот тег будет проигнорирован, если в качестве выходного формата используется не LaTeX. [формулы в Doxygen]

```
Расстояние между \f$(x_1,y_1)\f$ и \f$(x_2,y_2)\f$
вычисляется, как \f$\sqrt{(x_2-x_1)^2+(y_2-y_1)^2}\f$.
```

Расстояние между (x_1, y_1) и (x_2, y_2) вычисляется, как $\sqrt{(x_2-x_1)^2+(y_2-y_1)^2}$.

Документирование функций `@function` *имя-функции*
[функции в Doxygen]

```
  /**
  @function MyFunction

  Тестовая функция, имеет два входных параметра.

@param in\_Parameter1 --- параметр, отвечающий за что-то
@param out\_Parameter2 --- параметр, отвечающий за что-то ещё
@return true если всё хорошо, false если всё плохо
  */

bool MyFunction (int in\_Parameter1, std::string out\_Parameter2);
```

Синтаксис имеет следующий вид:

1. Для генерации шаблона конфигурационного файла

   ```
   doxygen [-s] -g [configName]
   ```

   If — is used for configName doxygen will write to standard output.

2. Для обновления шаблона конфигурационного файла

   ```
   doxygen [-s] -u [configName]
   ```

3. Генерация документации на основе конфигурационного файла

   ```
   doxygen [configName]
   ```

   If — is used for configName doxygen will read from standard input.

4. Для создания файла шаблона управления макетом при генерации документации

   ```
   doxygen -l layoutFileName.xml
   ```

5. Для генерации файла документации с заданным расширением (на примере `.rtf`)

   ```
   RTF:   doxygen -e rtf extensionsFile
   ```

6. Для создания шаблона стиля для RTF, HTML или LaTeX

```
RTF:   doxygen -w rtf styleSheetFile
HTML:  doxygen -w html headerF footerF styleSheetF [configFile]
LaTeX: doxygen -w latex headerF footerF styleSheetF [configFile]
```

If -s is specified the comments in the config file will be omitted. If configName is omitted ‘Doxyfile’ will be used as a default.

Файл настроек можно редактировать в любом текстовом редакторе или с помощью специальных утилит с графическим интерфейсом. Одна из них, Doxywizard, поставляется вместе с Doxygen.

Схема работы Doxygen (совместно с Doxywizard) представлена на рис. 1.

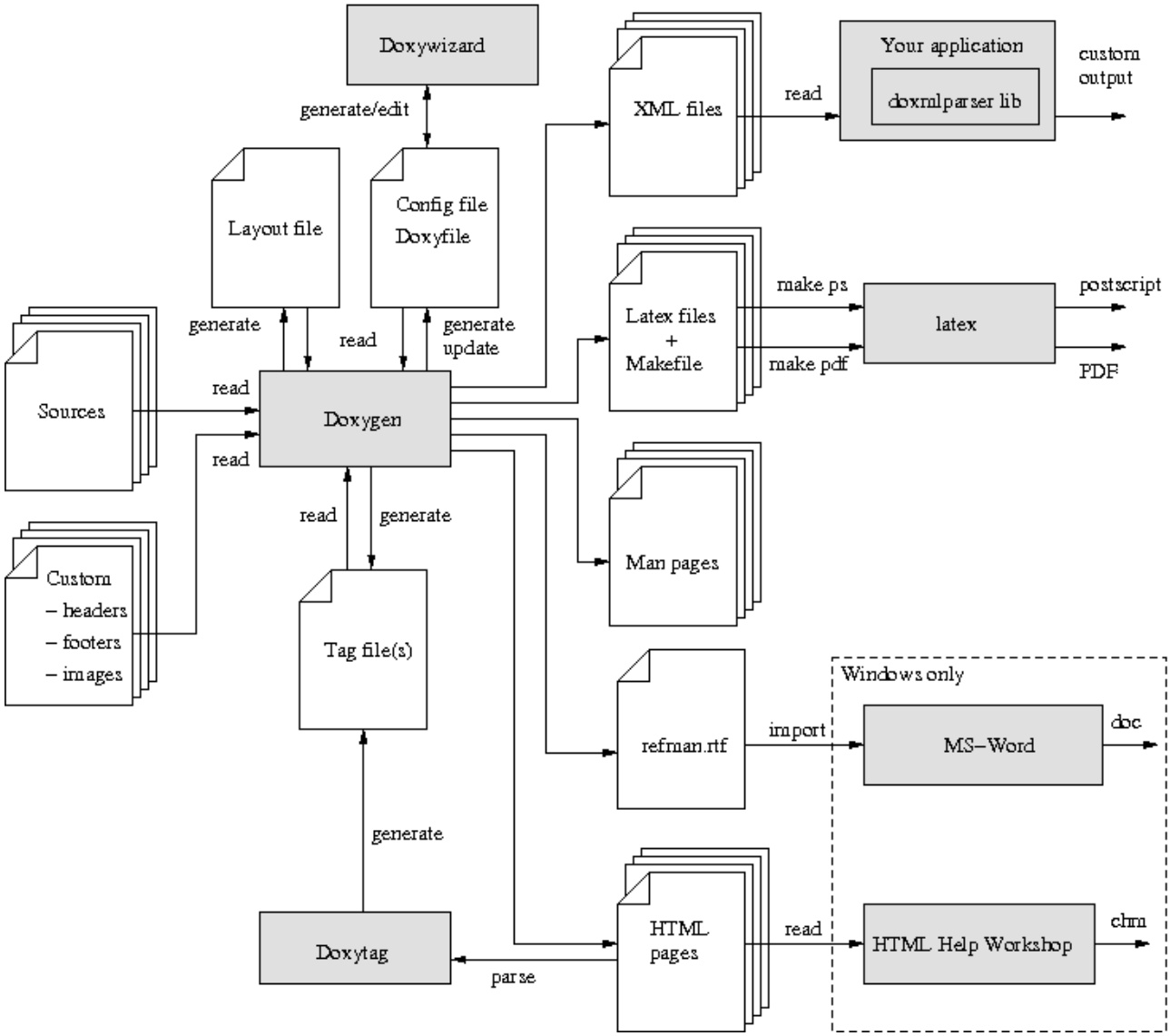

Рис. 1. Информационные потоки Doxygen