

Мельничук С.Ф., Гемба О.В.

УНК “Институт прикладного системного анализа” НТУУ “КПИ”, Киев, Украина

Методы построения моделей программных проектов

В процессе разработки ПО часто возникает необходимость поддерживать, сопровождать или модифицировать уже готовые программные решения и системы (в том числе корпоративные). Часто такие задачи решаются силами сторонних команд, а не команд, работавших непосредственно над проектом. Таким образом, ярко выражается необходимость передачи знаний, проектных решений и деталей реализации проекта из одной команды в другую. С этой проблемой должна справляться проектная документация – спецификации, варианты использования, реализации вариантов использования и т.д.

Даже для небольшого проекта (меньше 1000 часов) таких документов накапливается значительное количество (user stories, use cases, use case realizations и т.д.), и их обработка и усваивание человеком становится нетривиальной задачей, для выполнения которой может потребоваться значительное количество времени [1].

Выражается необходимость создания единого комплекта документации, т.н. документной модели программного проекта – цель которого в предоставлении неформализованных (стандартные документы для разработки ПО содержат множество избыточных и недостаточно структурированных для использования на этапе поддержки данных), а более практически ориентированных данных, знаний о проекте для разных уровней абстракции и разных областей применения (развертывание, поддержка, модификация и т.д.)

Результатом работы является подход к решению задачи построения документной модели в иерархическом виде на основе мета-информации, описывающей связи между компонентами разных уровней абстракции, которые документируются. С задачей документирования проекта в общем виде может справиться организация документации по принципу википедии (wiki). Одна из сложностей восприятия формата wiki – это наличие, фактически, только прямых (горизонтальных) связей между узлами (страницами) [2, 3]. При добавлении иерархических меток (позволяющих создавать вертикальные связи между узлами), основанных на частичном описании предметной области разработки ПО (например, в виде онтологии), получим возможность представления срезов модели по уровням абстракции (например, визуализация связей между слоями, модулями, сервисами, транзакциями и т.д.)

При детальном рассмотрении и анализе разных моделей жизненных циклов разработки ПО (модель “водопада”, спиральная модель, итеративная модель) можно найти удачные временные точки для использования описанного подхода с документами, создание которых включает в себя модель жизненного цикла разработки. Таким образом, накладные затраты времени на создание описанной документной модели остаются минимальными, если создание модели происходит во время процесса разработки.

Результаты использования такого подхода предполагают постепенное создание общекорпоративной базы знаний о процессах разработки ПО, применяемых в компании, включающей в себя как общие понятия, так, возможно, и какие-либо специфические для данной компании или предметной области. На основе такой базы знаний (которую можно представить в виде онтологии), можно унифицировать форматы описаний программных проектов (возможно, специфических для конкретной компании) и представить отчетность по проектам. Также описанная база знаний и построенные на ее основе документные модели проектов помогут при документировании следующих проектов и поддержке/сопровождении/модификации существующих.

Литература. 1. Wikipedia (2012, Feb. 09), *Software documentation* [Online] Available: http://en.wikipedia.org/wiki/Software_documentation. 2. Wikipedia (2012, Feb. 22) *Wikipedia:Linking* [Online] Available: <http://en.wikipedia.org/wiki/Wikipedia:Linking>. 3. Wikipedia (2012, Feb. 22) *Wikipedia:External Links* [Online] Available: http://en.wikipedia.org/wiki/Wikipedia:External_links.