Предмет: «Технология разработки программного обеспечения»

Тема: «Оценка размера и трудоемкости проекта »

Теоретический материал

Естественный подход к решению любой проблемы, если она слишком сложная, заключается в ее разделении на более простые, пока они не ока­жутся пригодными для решения. Затем каждая из подпроблем решается от­дельно, при этом предполагается, что все решения могут быть объединены в одно общее. Оценка проекта программного продукта - это также форма реше­ния проблемы и в большинстве случаев эта проблема также слишком слож­ная. Поэтому проблема , которую требуется решить, должна быть предвари­тельно представлена в виде более простых и более управляемых подпроб­лем.

Размер программного продукта, оцениваемый числом строк иходного кода, относится, как уже отмечалось, к размеро-ориентированным метрикам, которые могут быть непосредственно измерены для каждого разработанного проекта.

Как правило, организации, занятые разработкой программного обеспе­чения, ведут регистрацию основных параметров каждого выполненного про­екта-в виде записей базы данных. О каждом проекте собирается и храниться обычно следующая информация:

* трудозатраты (в человеко-месяцах);
* стоимость разработки (в соответствующей валюте);
* объем написанных программ (в строках исходного кода);
* численность разработчиков (человек);
* объем документации (в страницах) и т.п.

В результате обработки этих данных, собранных по многим проектам, может быть получен, кроме основных, ряд производных базовых метрик, по­зволяющих проводить впоследствии оценки средней производительности труда разработчиков, стоимости разработки одной строки кода, относительно­го числа страниц документации на тысячу строк кода и т.д.

Перечисленные метрики позволяют прогнозировать трудоемкость и стоимость разработки новых проектов. Однако прежде всего возникает необ­ходимость определения размеров будущего программного продукта, выра­женных в тысячах строк исходного кода.

При разработке программного проекта менеджеру требуется оценить параметры проекта уже на самом начальном этапе системного анализа. Вме­сте с тем использование числа строк исходного кода (СИК) для оценки раз­меров программного изделия требует определенной степени детальности описания будущего продукта, что может быть достигнуто только в результате предварительной функциональной декомпозиции изделия.

В связи с вышесказанным последовательность действий для получения основных оценок программного проекта можно себе представить следующим образом:

1. На основе предварительного системного анализа должны быть опре­делены основные функциональные подсистемы будущего программного про­дукта и ассмотрено их содержание с точки зрения сложности их программной реализации как отдельного функционального блока программ.
2. Каждый функциональный блок необходимо оценить, используя экс­пертную оценку его размера в строках исходного кода. Очевидно, что после декомпозиции новной функции системы оценка каждого блока может быть выполнена более точно, чем оценка всей системы в целом.
3. В процессе экспертной оценки размеров каждого блока должны опре­деляться доверительные границы каждой оценки.
4. Суммирование оценок числа строк кода по всем блокам и вычислениеобщей дисперсии суммарного среднего позволяет определить среднее число строк исходного кода программного продукта и границы доверительного ин­тервала полученной оценки. Эти данные становятся исходными для после­дующих рачетов.
5. Для определения стоимости и трудоемкости разработки проекта требуются данные о производительности труда исполнителей (строк кода в чело­веко-месяцах) и средней стоимости создания строки кода. Эти данные обычно характеризуют уровень производства, достигнутый в организации на данный период времени и должны быть взяты из базы данных о предыдущих проек­тах подобного типа. Такая база данных может создаваться в организации, где планируется разработка программного проекта, но может использоваться ин­формация из хранилищ других организаций. На практике использование этих исторических данных обычно дополняется их экспертной корректировкой, учитывающей особенности, сложность и объемы каждого из блоков.
6. Используя данные об оценках размера каждого блока и данные о производительности труда исполнителей и о затратах на одну строку кода (также применительно к конфетному блоку), легко определить трудоемкость и стоимость разработки каждого блока с указанием значений на границах доверительного интервала.
7. В результате суммирования данных по всем блокам получаем оценки трудоемкости и стоимости разработки всего изделия и степень доверия этих оценок.

**Метод оценки**

В этой части даны пошаговые рекомендации для оценки трудоемкости проекта. Практически за каждым пунктом стоят годы опыта, успехов и ошибок. Предложенный метод в основном применим к проектам на этапе, когда требования проработаны до уровня пользовательских или функциональных. Многие же советы и рекомендации подойдут к любым проектам разработки и инженерным проектам вообще.

**Шаг 1. Подготовка (Prerequisites)**

1. Выделяйте либо требуйте ресурсы для оценки. К сожалению, далеко не все менеджеры и клиенты понимают всю сложность и трудоемкость качественного процесса оценки. Требуйте, чтобы у оценщиков было достаточно времени и прочих ресурсов для работы. Чем больше времени инвестировано в оценку, тем она, как правило, точнее. Даже несколько минут анализа задачи могут существенно повысить точность по сравнению с оценкой «на ходу».

Помимо времени, может понадобиться доступ к существующим системам, их коду и документации, данным, лицензиям, экспертам и прочим ресурсам.

2. Знайте стейкхолдеров проекта и людей, принимающих решения. Часто у разных людей (например, представителей заказчика) разное видение и интерпретация того, что и как нужно сделать. Постарайтесь определить все картины мира либо обозначить основные конфликты в них.

3. Внимательно читайте контракт, если он есть, пусть и в форме черновика. Вы можете открыть для себя много нового в отношении формальных обязательств по проекту. Например, требования к качеству работ, поставке, срокам, документации. Это напрямую влияет на объем работ и их оценку.

4. Налаживайте взаимодействие с отделом продаж. Рассинхронизация между продавцами и исполнителями — весьма частая причина проектных проблем. Как правило, главный KPI продавцов — объем продаж, проблемы выполнения проектов их беспокоят намного меньше. Убедитесь, что вы знаете, что они продают, а они знают как вы собираетесь это выполнять.

**Шаг 2. Опишите требования и рамки проекта (scope)**

Четыре категории требований по степени известности

В момент оценки требования можно условно разделить на 4 части по степени известности, понятности и количеству рисков.

Known knowns. Заявленное в явном виде, понятное и достаточное для точной оценки. Что с этим делать? Оценить.

Unknown knowns. Косвенное заявленное либо не заявленное, но легко доступное. Не ленитесь прочитать требования и спецификации, «прокликать» гиперссылки в них, зайти на сайт клиента, спросить SME. Это не требует больших трудозатрат, но позволяет намного лучше понять требования и выявить скрытые риски. Что с этим делать? Найти, прочитать, спросить, прояснить, перевести в Known knowns, оценить.

Known unknowns. Недостаточные требования, неготовая документация и тому подобное. Например, заказчик знает, что будет интеграция с другими системами, но не знает протоколов, форматов и объемов обмена данными.

Что с этим делать? Прежде всего, решить, хотим ли мы брать на себя риски по оценке и выполнению этих требований. Если да, то неплохо бы сделать предположения насчет задач, которые мы не знаем, и добавить буфер времени на всякий случай. Если же нет, выписать явно, что эти требования не оценены либо остались вне рамок проекта.

Unknown unknowns. Практически во всех проектах происходит что-то, о чем и не подозревали вначале. Например, баги во внешних библиотеках, сюрпризы от поставщиков браузеров и операционных систем, сложности в требованиях либо реализации, упущенные требования. Изменения в требованиях сюда напрямую не относятся, но часто покрываются за счет этого буфера, поскольку запускать процедуры управления изменениями на каждый клиентский запрос весьма накладно. Мало что можно с этим сделать, кроме добавления буфера.

**Действия по подготовке требований к оценке**

Категоризируйте требования и все, что может ими считаться, как указано выше.

Помимо спецификаций и официальных документов, стоит проанализировать также протоколы встреч, письма с намерениями, все вплоть до разговоров в курилках, «поскольку ожидания в будущем часто становятся требованиями либо влияют на них.

Зафиксируйте ожидания, формальные и неформальные, функциональные и нефункциональные требования, ссылки на другие документы, спецификации и стандарты.

По возможности проверьте актуальность предоставленной документации, поскольку нередки случаи оценки по устаревшим документам.

Постарайтесь подтвердить список требований и предположений с заказчиком. Это может выйти не сразу. В крайнем случае можно прибегнуть к приему «согласование по умолчанию»: высылайте письмо с просьбой подтвердить либо прокомментировать скоуп, аккуратно намекнув, что отсутствие ответа означает молчаливое подтверждение.

Явно опишите то, что в скоуп не входит: что вы не оцениваете и не собираетесь делать, поскольку любое, даже самое детальное описание подвержено различным интерпретациям, сознательным или несознательным. Типичный пример: клиент может ожидать бесконечной поддержки программы, которую вы пишете, если гарантийный период поддержки не прописан явно.

Если же вы по какой-то причине не оценили задачу, необходимо прямо написать об этом( таблица 1):

Таблица 1 – трудности оценки проекта

|  |  |  |
| --- | --- | --- |
| **Feature** | **Estimate** |  |
| Login page | 24mh |

|  |
| --- |
| — оцениваем |
|  |

 |
| Single sign-on | NOT ESTIMATED |

|  |
| --- |
| —оценим позже, когда будет понятнее |
|  |

 |
| Sign-up link | Out of scope | — не собираемся оценивать и делать |

Домашнее задание

Описать, на Ваше мнение, основные характеристики оценки размера проекта.

Результат скинуть на почту с названием по образцу — Предмет\_Тема\_датаОтправки.

В теме письма указать ФИО преподавателя.

Дата исполнения до следующей пары.