Лекция 2. Ранние этапы жизненного цикла

ТЕСТИРОВАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

2014

Павел Степанов

Старший преподаватель

Кафедра компьютерной математики и программирования ГУАП

1. Содержание

- Анализ требований
- Анализ проекта

2. Минутка страха

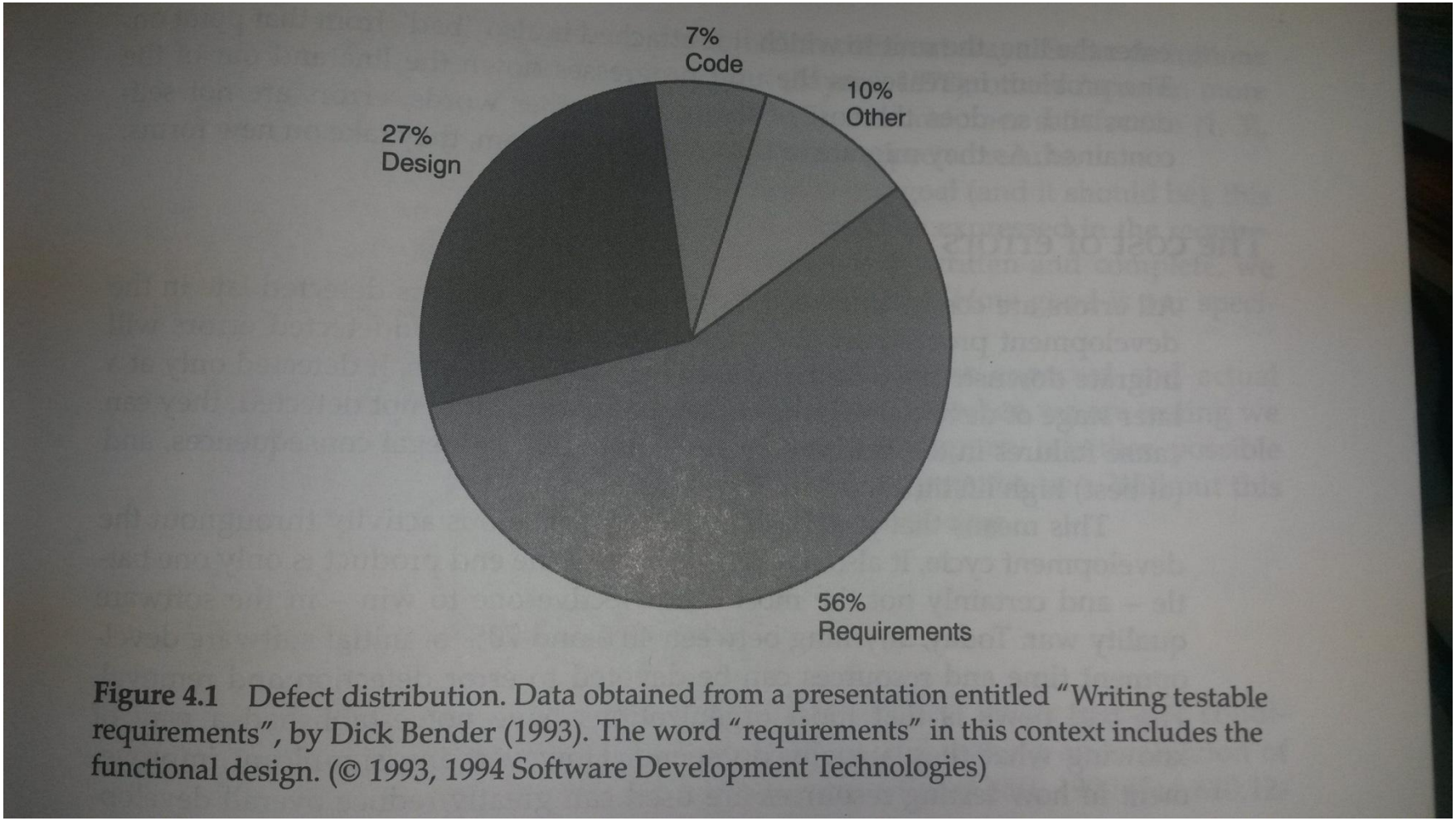

Figure 4.1 Defect distribution. Data obtained from a presentation entitled "Writing testable requirements", by Dick Bender (1993). The word "requirements" in this context includes the functional design. (© 1993, 1994 Software Development Technologies)

3. Немного теории

- Требования часто поступают к разработчикам в форме, не позволяющей приступить к их немедленной реализации
  - Потому что обычно формулируются требования не инженерами, а маркетологами и пользователями
- Поэтому каждое бизнес-требование необходимо преобразовать в функциональное требование
- Обычно уровней требований три и поступать требования могут на каждый из них
  - Бизнес-требования
  - Требования пользователей
  - Функциональные требования

4. Жизненный цикл требований

- Требования могут меняться на любом этапе жизненного цикла программного продукта
- Меняются они по-прежнему маркетологами и пользователями
    - Поэтому необходимо постоянно поддерживать связь между бизнес требованиями и функциональными требованиями

5. Отслеживаемость (трассируемость) требований

- В процессе эволюционирования бизнес-требований связь между функциональными требованиями и бизнес требованиями может теряться
  - Появляются бизнес требования, не имеющие функциональных требований
  - Появляются висящие функциональные требования, не имеющие связи с бизнес-требованиями
- Эта проблема называется нетрассируемостью требований

6. Пример нетрассируемости

- Маркетинг заказывает разработку интернет-магазина бытовой техники
- Были сформулированы следующие пользовательские требования*
  - Веб-сайт должен уметь показывать ассортимент магазина, производить по нему поиск, сравнивать два наименования бытовой техники из одной категории, оставлять и читать отзывы покупателей, делать заказ и оплачивать его через интернет, указывать детали доставки по территории РФ

* Требования могут быть в дальнейшем детализированы еще сильнее, но к сути примера это не относится

7. Пример нетрассируемости – cont.

- Посмотрев на происходящее, маркетинг решает, что оставлять отзыв можно не абы кому, а только проверенным лучшим покупателям (с рейтингом 4 и выше)
- Добавляются следующие требования
  - Необходимо уметь регистрировать пользователя и указывать его рейтинг

8. Пример - окончание

- Маркетинг подумал и решил, что рейтинг ничего на самом деле не значит и для бизнеса нужно, чтобы все отзывы рассматривал администратор сайта.
- Добавляются следующие требования
  - Администратор сайта должен уметь разрешать публикацию отзыва
- Убирается требование
  - Необходимо уметь указывать рейтинг пользователя
- Кто найдет нетрассируемое требование?

9. Нетрассируемость - summary

- Основаная причина возникновения нетрассируемости – изменение требований

10. Другие проблемы требований

- Некорректность
- Двусмысленность
- Неполнота
- Непроверяемость
- Непонятность

11. Некорректность требований

- Требование противоречит другим требованиям либо не может быть исполнено по каким-то иным причинам
  - Нарисовать семь взаимно перпендикулярных красных линий, две из которых должны быть нарисованы зеленой краской, а две – прозрачной.

12. Двусмысленность требований

- Требование можно интерпретировать различным образом
  - В помещении должны быть пол, потолок и стены. Они должны быть красного цвета.
    - Кто должен быть красного цвета – стены или все помещение?

13. Неполнота

- Требования оставляют слишком большой простор для фантазии
  - Разработать телефон, который при открытой крышке позволит снять трубку, нажав на зеленую кнопку
    - А что делает зеленая кнопка при закрытой крышке?

14. Непроверяемость (нетестируемость)

- Соблюдение требования невозможно проверить формальными методами
  - Форма кресла должна быть удобной.
    - Удобной – кому? Как это проверить?

15. Непонятность

- Необходимо разработать программу с кандибобером
  - Все же ясно написано – или вы что, не знаете, что такое кандибобер?
  - А знает ли заказчик?

16. Кандибобер

- Кандибобер – используется только в выражении “с кандибобером” и означает “лихо, отлично, на славу” (толковый словарь Ушакова)
    - Стало ли требование понятнее?

17. Дополнительные проблемы

- Разные источники выделяют дополнительные классы проблем требований
  - Не реализуемость в рамках текущего уровня технологии
  - Ненужность и бесполезность
  - Немодифицируемость
  - Неупорядоченность по важности
  - Отсутствие количественных метрик
  - другие

18. Вывод

- Никогда не забывайте, что заказчик хочет заплатить вам денег. Выяснить, что именно ему надо – это ваша задача, а не его.

19. Тестирование проекта

- Вы собрали требования и создаете дизайн продукта. Здесь тоже есть место тестированию

20. Основные проблемы проекта

- Потеря связи с требованиями
  - Невозможность реализации требования
  - Избыточная функциональность
- Неполнота или двусмысленность
- Излишняя сложность
- Неиспользование подходящих паттернов
- Использование паттернов не к месту

21. Потеря связи с требованиями

- В проект не заложены механизмы реализации требования
- Например
  - В базе данных отсутствуют поля для хранения информации
  - На формах нет кнопок для решения задач
- Либо наоборот, в проект включены лишние детали, которые никто не просил

22. Неполнота или двусмысленность

- Неполнота – непонятно, как именно должно реализовываться требование
- Двусмысленность – требование может быть реализовано более, чем одним способом

23. Излишняя сложность

- Проект можно улучшить за счет упрощения

24. Паттерны

- Паттерны очень полезны, если их правильно применять
  - Упрощают понимание кода
  - Упрощают написание кода
  - Снижают количество ошибок
- Паттерны могут быть крайне вредны, если применять их не по делу

25. Как искать ошибки в проекте?

- Путем использования своих знаний, навыков, опыта и здравого смысла.

26. Q&A