Session 2

Specification

Moving needs and practical concepts to reasonable models.

Ernest Cachia
Department of Computer Science

Session Contents

- Specification fundamentals
- Model examples

Ernest Cachia
Department of Computer Science

Structured Specification

A possible generic definition:

– "A pre-defined set of steps aimed at producing an efficient description of what is to be done."

A possible s/w-oriented definition:

– "A set of clear and un-ambiguous guidelines, tools and methods to aid in the production of a valid and usable description of what a software system is to achieve."

I/O requirements of specification

- Input to specification:
  - A complete feasibility study
  - A confident understanding of user requirements (usually in abstract form)
- Output from specification:
  - Graphic (and textual were necessary) descriptions of all system aspects
  - Feed-back of any system-forced constraints

Some specification properties

- If carried out MUST precede design
- Will aid the design process
- Must be structured to correctly specify user needs
- Should contain no embellishments
- Should predominantly be graphic
- Should be easily and clearly understood
- Should make provisions for prototyping

How does specification help?

- Forces one to better and more comprehensively analyse a situation
- Forces the analyst *(i.e. the person producing a description of the system)* to interface more closely and seriously with the system user(s) *(i.e. bridges the gap between the supplier's and the customer's fields of knowledge)*
- Offers the ideal starting point for sound system design

S/w specification until recently

- User requirements took second-stage to actual system implementation
- Specification was never taken seriously
- “Seasoned” programmers felt degraded when asked to specify what they thought they already knew well
- Quite often systems were specified after their implementation (often bug-driven)

Reasons for specification neglect

- Previous personal nature of software
- Previous software system size
- Previous software system demands
- Warped ideas about specification being a tool for people who can’t think well *(give the poor guys something to help organise their thoughts with)*
- Basic laziness and “cross-your-fingers-and-hope” attitude

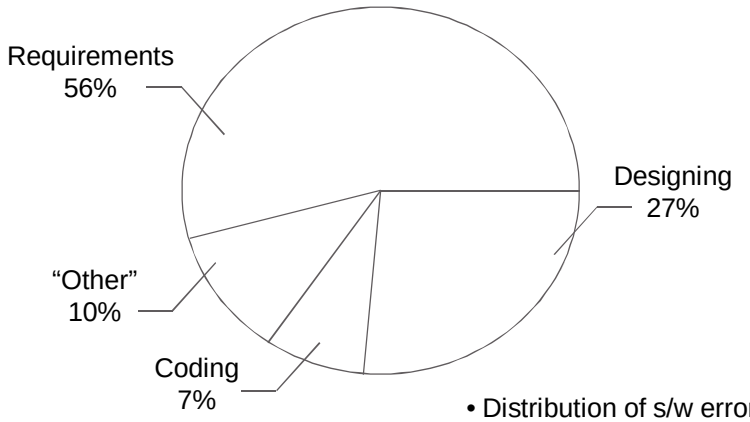
Structured specification Tools

- Graphic:
  - Data Flow Diagrams (DFDs)
  - Finite State Machines (FSMs)
  - Data Structure Diagrams (DSDs)
  - *etc.*
- Textual:
  - Natural language (e.g. English)
  - Structured Natural Language
  - Program syntax
  - *etc.*

The "Analyst"

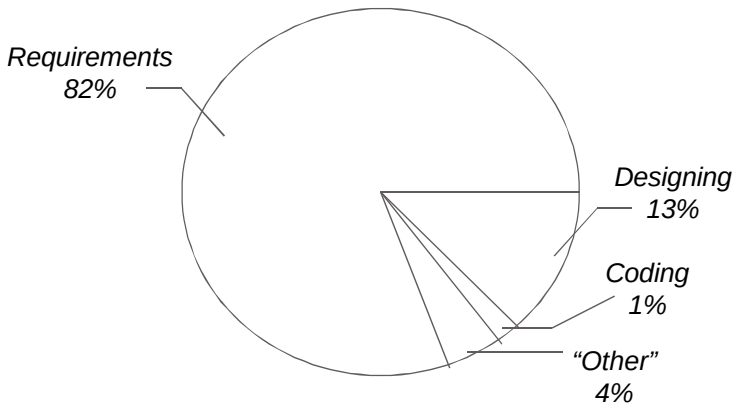
- Someone with the ability to capture user and system needs at different levels
- Someone who can envisage a system from different aspects (points of view)
- Someone who can communicate ideas on a non-technical basis
- Someone who can save (or cost) an organisation a lot of effort and money

The importance of specification (1/2)



• Distribution of s/w errors

The importance of specification (2/2)

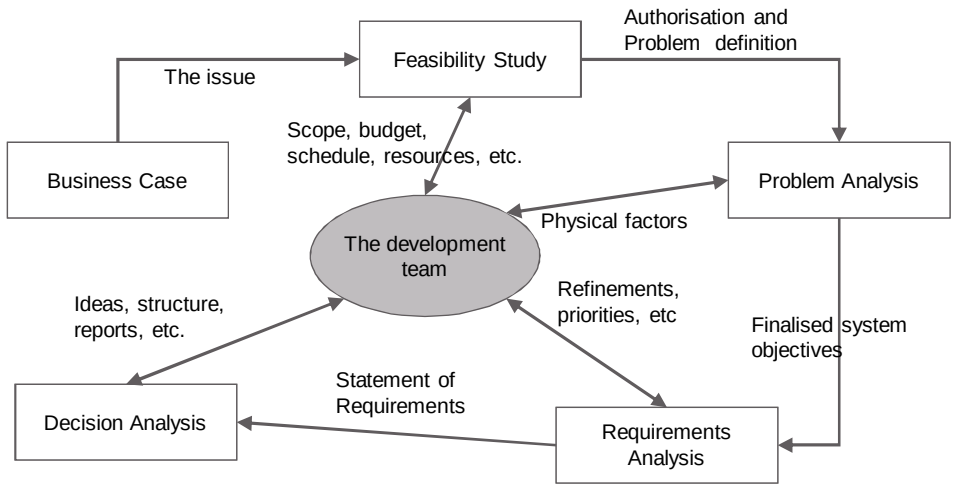


The Specification Process

- Generally referred to as *Systems Analysis*
- Systems Analysis:

  "A problem solving technique that decomposes a system into its component pieces for the purpose of studying how well those component parts work and interact to accomplish their purpose" *[J. L. Whitten]*

Visual Flow of System Analysis



The Model-Driven Approach

- Structured Analysis
  - Process-centred
- Information Engineering
  - Data-centred
- Object-Oriented Analysis
  - Both process and data-centred (i.e. object-centred)

The Accelerated Analysis Approach

- Discovery Prototyping
  - Involves the use of cheap and fast partial implementation of certain system features
  - Prototypes allow system stakeholders to learn about the final system
- Rapid Architecture
  - Based on reverse engineering principles
  - Don't re-invent the wheel – understand its function and improve on it!

System Planning

Moving on to System Planning

What is (and isn't) System Planning

- A formal definition: "An outline, sketch, or layout of the form or structure of a work." - *Random House College Dic. (1972) – modified by author*
- Software system planning: "transforming *what* is required to be done (i.e. the task) into a blueprint of *how* a solution is to be achieved" - *autor (1997)*
- System planning does not guarantee correctness or uniqueness of a particular solution

Goal of Software System Planning

- To obtain the smoothest possible transition from “what is” to “how is” a solution obtained
- Offer tools (graphic) and guidelines to facilitate problem comprehension and its transition to “how” to plan its construction
- Offer some sort of criteria by which to gauge the quality of a specific solution

How Does System Planning Help?

- Better s/w system understandability
- Improved system reliability
- Better s/w system flexibility
- Longer system durability (effective use)
- Smoother s/w development process
- Better end-user efficiency

Effort Associated with System Planning

- More thought (re. the s/w system)
- More discipline (both thought & actions)
- Training in the use of specific tools
- Training in the use of specific techniques

The situation till very recently

- Little or no requirements specification
- Little or no system specification
- Only cosmetic system architecture design
- System planning was more a “forced” activity rather than recognised need
- Difficult system component integration
- Primitive (usually empirical) testing
- Large maintenance costs (~75%)

How Did We Get Here?

- Historical evolution (good programmers are not necessarily good designers)
- The nature of software has radically and rapidly changed (and is constantly changing) to reflect changes in our society
- Plain laziness and human (sometimes) rebellious nature

Know your enemy (meaning task)

- Clear your mind of pre-conception as to which model your system is to follow
- Never try to force a design on to a model
- Leave “how” to as later on as possible
- “What” should lead to “how” not vice-versa

Major thrusts of System Planning

- System simplification
- Natural Hierarchical system structure
- Graphic tools (methods)
- Design elaboration (overall & detailed)
- Design solution evaluation

System Simplification

- Divide & conquer (Julius Caesar)
  - Identify "isolatable" tasks within a larger one
  - Tackle smaller tasks at different levels of design
  - Design the right relationship between tasks
  - Design system structure using tasks as blocks
- Black-Box concept (also Mr. J. Caesar)
  - Clearly defined I/O
  - Clearly defined function
  - Internals may be ignored (at that given point)

Hierarchical System Organisation

- Very old concept (even older than J. C.)
- Permeates our whole existence
- Tantamount to nature itself

e.g.

- The Universe: *Galaxies, star-clusters, solar systems, planets, land masses, continents,… sub-atomic particles, and who knows what else!*
- *Company structures*
- *Government establishments*
- *Society, etc.*

Graphic Tools

- A picture is (can be) worth a thousand words - this is a physiological *fact* that has to do with brain evolution
- Examples of graphic tools for design and specification include
  - Structure Charts (SCs)
  - Data Flow Diagrams (DFDs)
  - Structure Diagrams (SDs)
  - (Good old) Flow Charts (FCs)

Summary (Session 2)

- Analysis of solutions;
- Specification of systems;
- Examples of Diagrammatic specification models;
- The notion of a system.

Ernest Cachia
Department of Computer Science