Unit 6: Lineair relationships (1) Time
Consolidation: C
Title: Program development.
Author: Erik van Eykelen, E11C
Date: dec-8-1988

A programming project has six definable components (the percentages are related to the amount of time each component roughly costs).

1 - Requirements analysis	10 %
2 - Specification	10 %
3 - Design	15 %
4 - Documentation	10 %
5 - Coding	15 %
6 - Testing and debugging	40 %

These six stages must be carried out one after another. Novice programmers often start with writing the code, then they have a hard time debugging the program because its shaky design gives rise to many errors and when the program finally is bug-free the documentation is written. Softwarehouses and software divisions of a big companies usually follow the six rules. "The Six Commandments" have at least two advantages:

1 - The programmer knows exactly what the program must do. It's important to know how a program should behave and what kind of users can be expected. Most of all, the programmer is not free to include all sorts of extras, an inclination of programmers that makes that many programs are never finished.

2 - A project can be carried out by more than one person. A softwarehouse has several systems engineers and programmers: the first group only works on the first four stages and the programmers on the last two. If one person would have to worry about all six stages a smaller throughput of projects would be the result.

To show that the six stages are a chronological sequence, each component is described briefly:

Requirements analysis
At this stage the requirements are defined to come to a solution of a problem. The requirements can be expressed either by the customer or by the systems engineer who knows what the client wants after being filled in about their wishes.

Specification
This stage defines the exact commands used in the program, the way files are stored and retrieved, how the output will be presented etc. The customer now knows how his program will look and act.

Design
The structure of the program is defined during the design stage. Algorithms (to solve problems) and data structures (to store data in an efficient way) are designed. The program is divided into modules so that several programmers can write the coding at the same time. Those modules must interact in a later stage, so very strict rules must be specified.

Documentation

This stage should overlap the specification and design stages. The programmers need technical documentation to write the coding and user documentation is bundled with the software.

Coding

The design is translated into a programming language. The design should be finished completely at this stage. Nothing is more frustrating for a programmer then to rewrite or exclude coding because the design has changed. If the design is well documented and well-considered, programming shouldn't take too long (see the percentages).

Testing and debugging

This stage consists of two components which are not mutual exclusive. If a bug is found during testing and this bug is fixed the testrounds must start all over to check if other parts of the program still behave according the specifications. Testing and debugging go hand in hand until the program is bug free, full proof (it accepts valid input and rejects false input after having warned the user) and fool proof (idiots shouldn't be able to crash the system by sitting on the keyboard).

P: Good

C: Excellent text.
Well-done!