

The Waterfall Lifecycle Model and Prototyping

Dror Feitelson

Basic Seminar on Software Engineering
Hebrew University
2009

Royce 1970

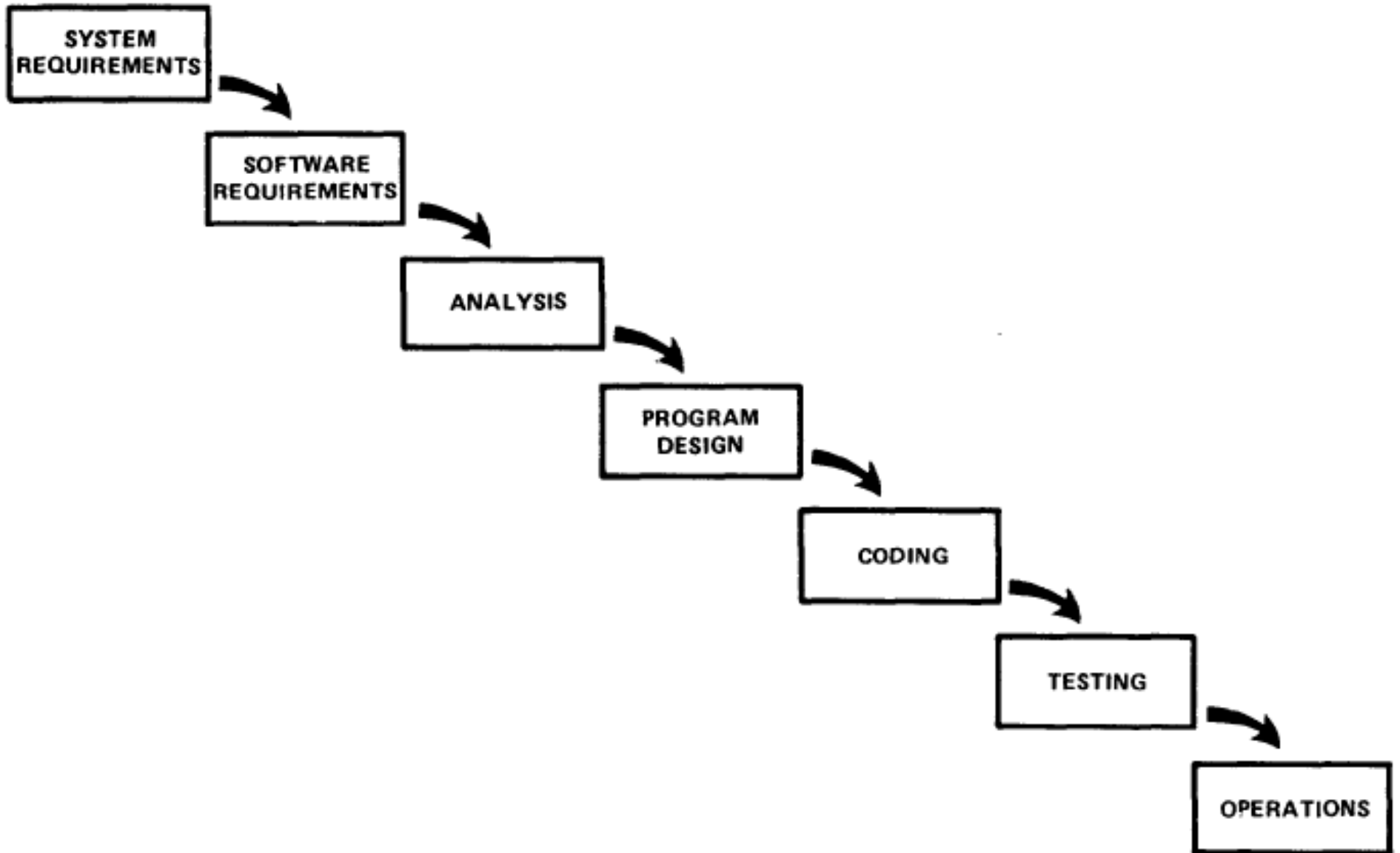
Dr. Winston W. Royce, “Managing the development of large software systems”.

Proc. IEEE WESCON, Aug 1970.

Reprinted 9th *Intl. Conf. Softw. Eng.*, 1987.

- Universally cited as the reference for the waterfall model
 - But, the word “waterfall” is not mentioned
 - And the model looks more like a cascade
- Moreover, the paper is actually against the waterfall model

The Basic Waterfall Model



Problems

- Doing everything in a single sequence is unrealistic
 - A better model involves iteration between successive steps
 - However, testing comes too late and may uncover problems in the initial design
 - The solution: do it twice
- (Same advice as Fred Brooks in *The Mythical Man-Month*, but referring to a full-scale system)

Additional Emphases

- Need to plan and control the testing
- Need to involve the client in key points
- Create multiple documents (requirements, specification, design, test plan, manual) and keep them up to date
 - “Write an overview document that is understandable, informative, and current. Each and every worker must have an elemental understanding of the system.”
 - “If the documentation is in serious default my first recommendation is simple: replace project management.”

The Frustration

This paper is very insightful and foreshadows several modern ideas.

So why is the waterfall model still being used?
(Or is it?)

Schneider 1996

Kurt Schneider, “Prototypes as assets, not toys”.
18th Intl. Conf. Software Engineering, pp. 522-531,
Mar 1996

- A lot of information is generated as part of prototyping
- Much of it is in the developer's head
- It has to be extracted and recorded to be useful

Types of Prototypes

- Presentation – emphasis on user interface, to show clients/users, get feedback, and understand requirements
- Proper prototype – temporary executable system used to gain experience and identify problems
- Breadboard – study alternatives and foster creativity
- Pilot – kernel used in real application domain that evolves into full system

Main Goal

- Capture experience gained by developer of prototype
 - Rationale behind design choices (cannot be extracted from code)
- Developer may express this in a tangled associative manner
 - Need tool support to record this
 - Need to summarize more clearly

Exercise:

The ministry of interior wants to create a computerized system for filing forms.

There are 31 different forms that need to be supported, with various fields.

Need to be able to save partially filled forms, return to form started earlier, print a completed form, and submit it electronically.

What would you implement in a prototype?