

Agile Software Development

Scrum

`shorturl.at/mdYAq`

COURSE CONTENTS

9/3/2025 8:49 AM

vijaynathani.github.io

Page 2

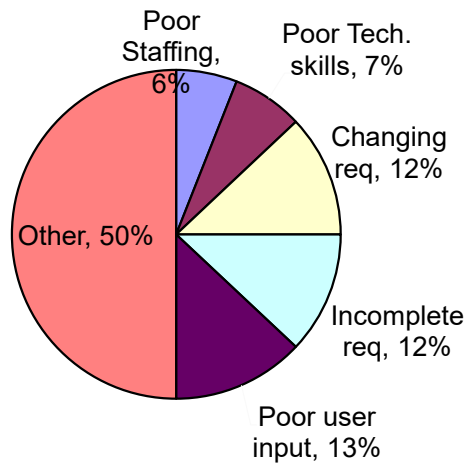
Pre-Training reading material

Problems in Software Development

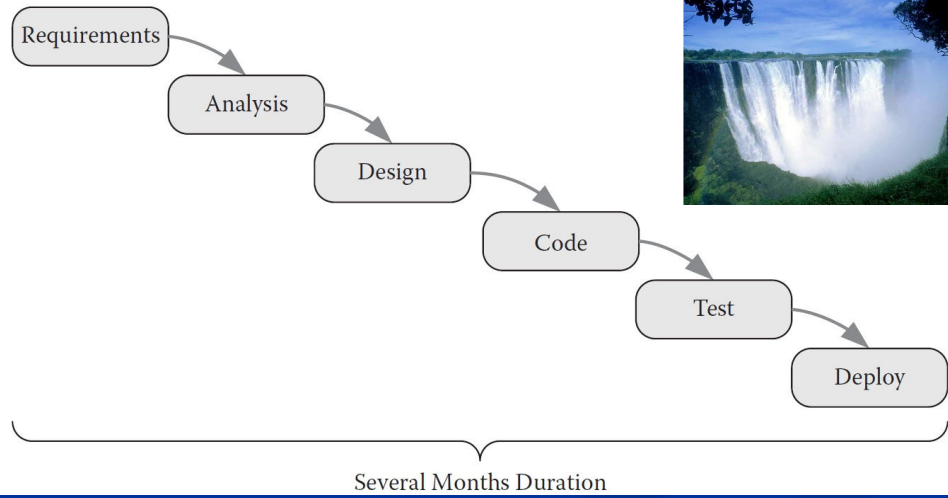
- Why does it take so long?
- Why is cost so high?
- Why can't we find all errors before we give the software to our customers?
- Why do we spend so much time and effort maintain existing programs?
- Why do we continue to have difficulty in measuring the progress as software is being developed and maintained?

Businesses are finding too often that their software systems act as brakes on their competitiveness, rather than as accelerators. Businesses are discovering that while their markets change rapidly, their software systems do not.

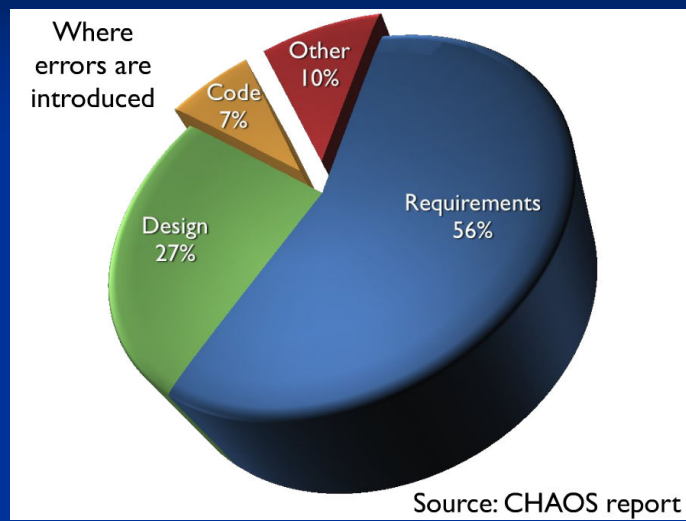
Reasons for Problems



The Waterfall Model

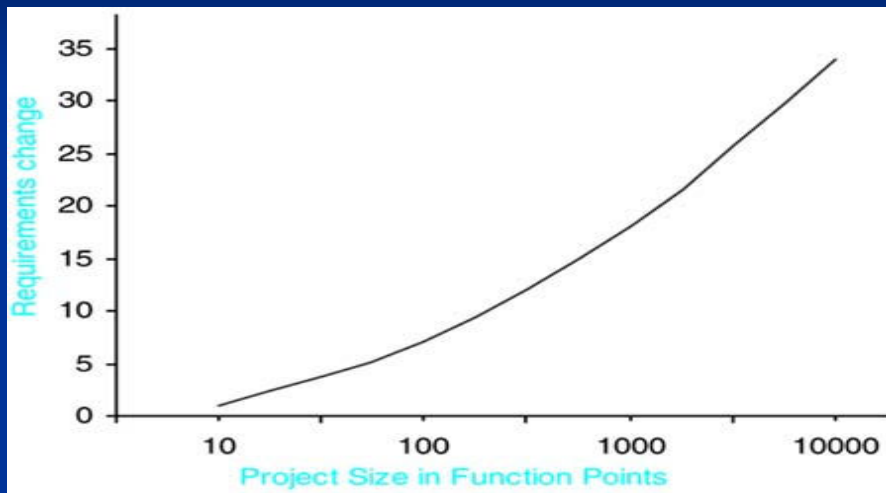


Source of Errors



See: <http://www.infoq.com/presentations/hellesoy-bdd-rspec>

How to handle change?



9/3/2025 8:49 AM

vijaynathani.github.io

Page 7

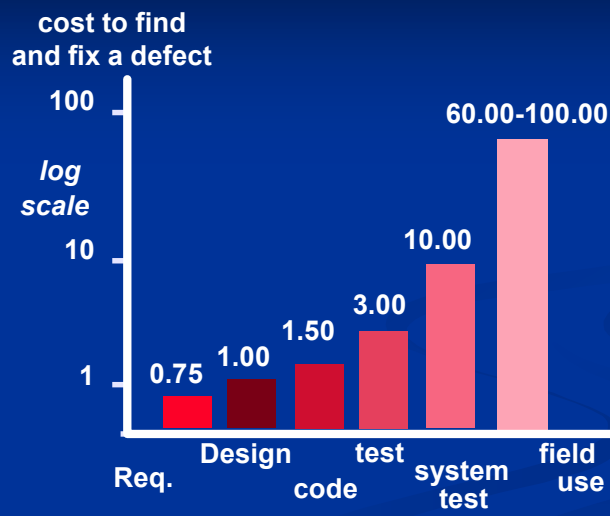
Software development is about change:

- Changing requirements
- Change in technology
- Change in people involved
- Change in competition, customer, government rules

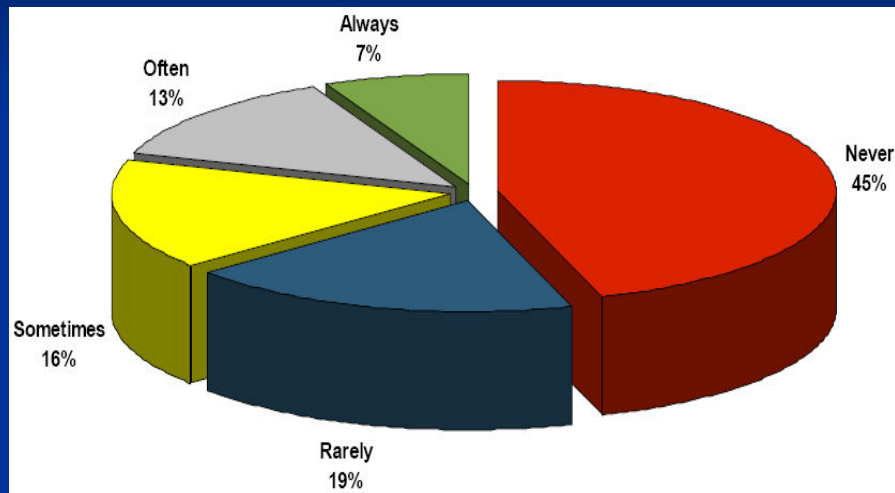
Planned development is like planned economy.

As per 2014 CHAOS report from Standish group. On Small projects, 24% features are due to change requests in end product; 35% for medium projects and 39% for large projects.

Cost of Change (COC) rises Exponentially

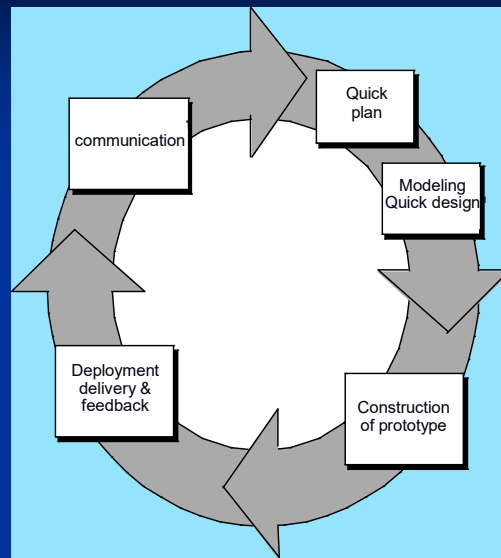


YAGNI violated

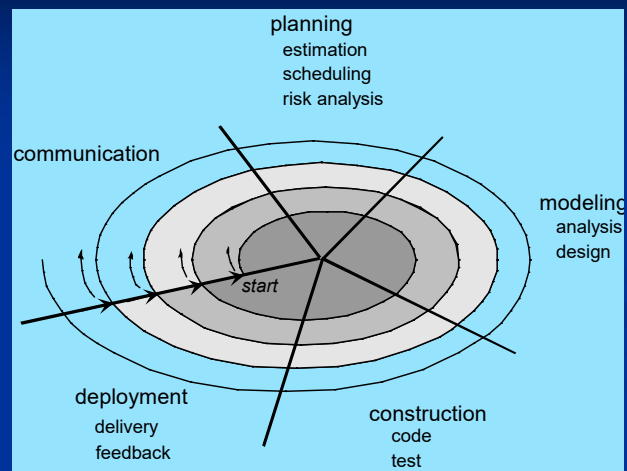


Standish group study reported at XP2002 by Jim Johnson.

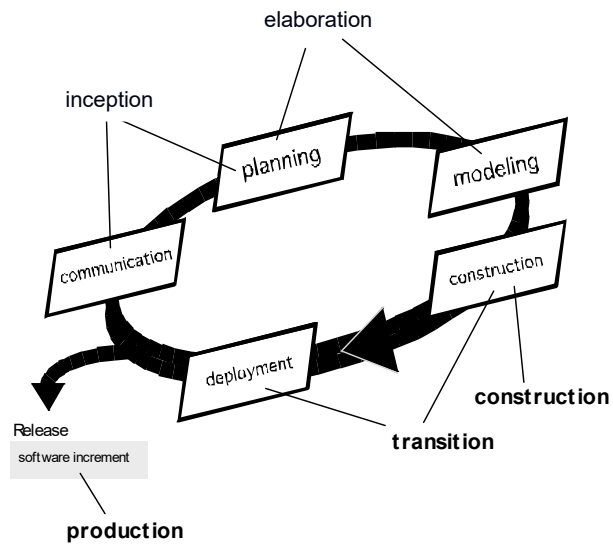
Evolutionary Models: Prototyping



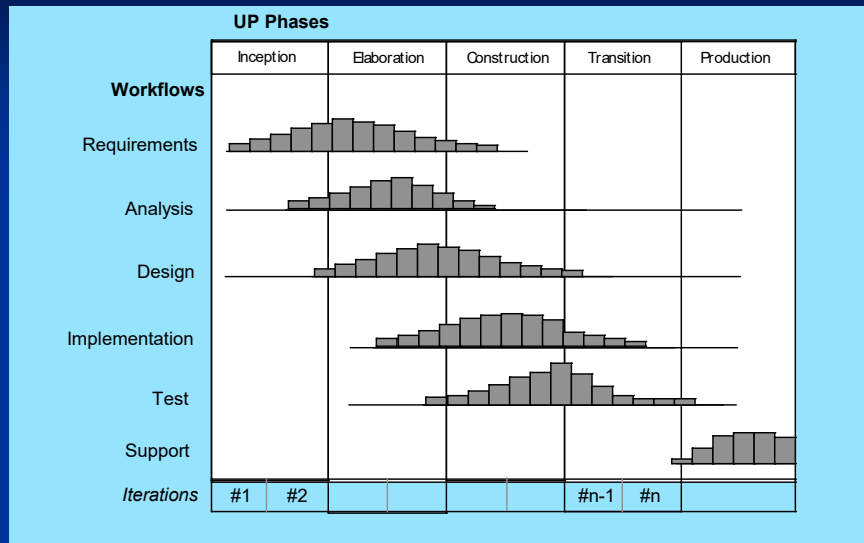
Evolutionary Models: The Spiral



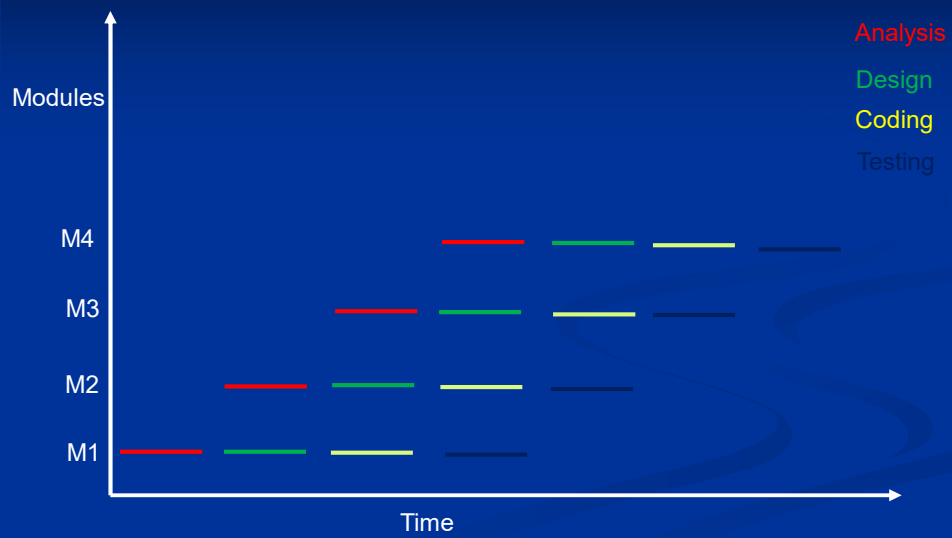
The Unified Process (UP)



UP Phases



Unified Process



Any other Process?

- Code & Fix
 - Cowboy Programming
- What is your process?
 - What is your definition of a Process?

Definition of Process: What you do under pressure is your process, everything else is window dressing or political correctness.

All methods are someone else's idea about what you should do when you develop software. It may be useful, from time to time, to borrow from those ideas and integrate them into your own style. It is essential, however, to transcend any method, even your own idiosyncratic method, and "just do it."

Software development is like riding a surfboard—there is no process that will assure a successful ride, nor is there any process that will assure that you will interact propitiously with the other surfers sharing the same wave. Published processes, like published methods, provide observational data from which you can learn and thereby improve your innate abilities—just as observation of master surfers enables you to improve yourself.

=====

Methodology is a social construction. Your "methodology" is everything you regularly do to get your software out. It includes who you hire, what you hire them for, how they work together, what they produce, and how they share. It is the combined job descriptions, procedures, and conventions of everyone on your team. It is the product of your particular ecosystem and is therefore a unique construction of your organization.

All organizations have a methodology: It is simply how they do business. Even the proverbial trio in a garage has a way of working away of trading information, of separating work, of putting it back together all founded on assumed values and cultural norms. The way of working includes what people choose to spend their

time on, how they choose to communicate, and how decision-making power is distributed.