

PROJECT DEVELOPMENT PROCESSES



MASSIMILIANO FASI



Development approaches

A development approach is the means used to create and evolve the product, service, or result during the project life cycle.

-
- A *Guide to the Project Management Body of Knowledge (PMBOK® Guide – Seventh Edition and The Standard for Project Management*, Project Management Institute, Newtown Square, PA, 2021.

Reference: Section 2.3 of PMBOK® Guide.

Development approaches

A development approach is the **means** used to create and evolve the product, service, or result during the **project life cycle**.

-
- A *Guide to the Project Management Body of Knowledge (PMBOK® Guide – Seventh Edition and The Standard for Project Management*, Project Management Institute, Newtown Square, PA, 2021.

Reference: Section 2.3 of PMBOK® Guide.

Development approaches

A development approach is the **means** used to create and evolve the product, service, or result during the **project life cycle**. [...] Three commonly used approaches are predictive, hybrid, and adaptive.

- A *Guide to the Project Management Body of Knowledge (PMBOK® Guide – Seventh Edition and The Standard for Project Management*, Project Management Institute, Newtown Square, PA, 2021.

Reference: Section 2.3 of PMBOK® Guide.

A spectrum of approaches



LESS ITERATIVE
LESS INCREMENTAL

MORE ITERATIVE
MORE INCREMENTAL

- ▶ Clear requirements
- ▶ Clear constraints
- ▶ Well-defined risks
- ▶ Low uncertainty

- ▶ Volatile requirements
- ▶ Unclear constraints
- ▶ High level of risk
- ▶ High uncertainty

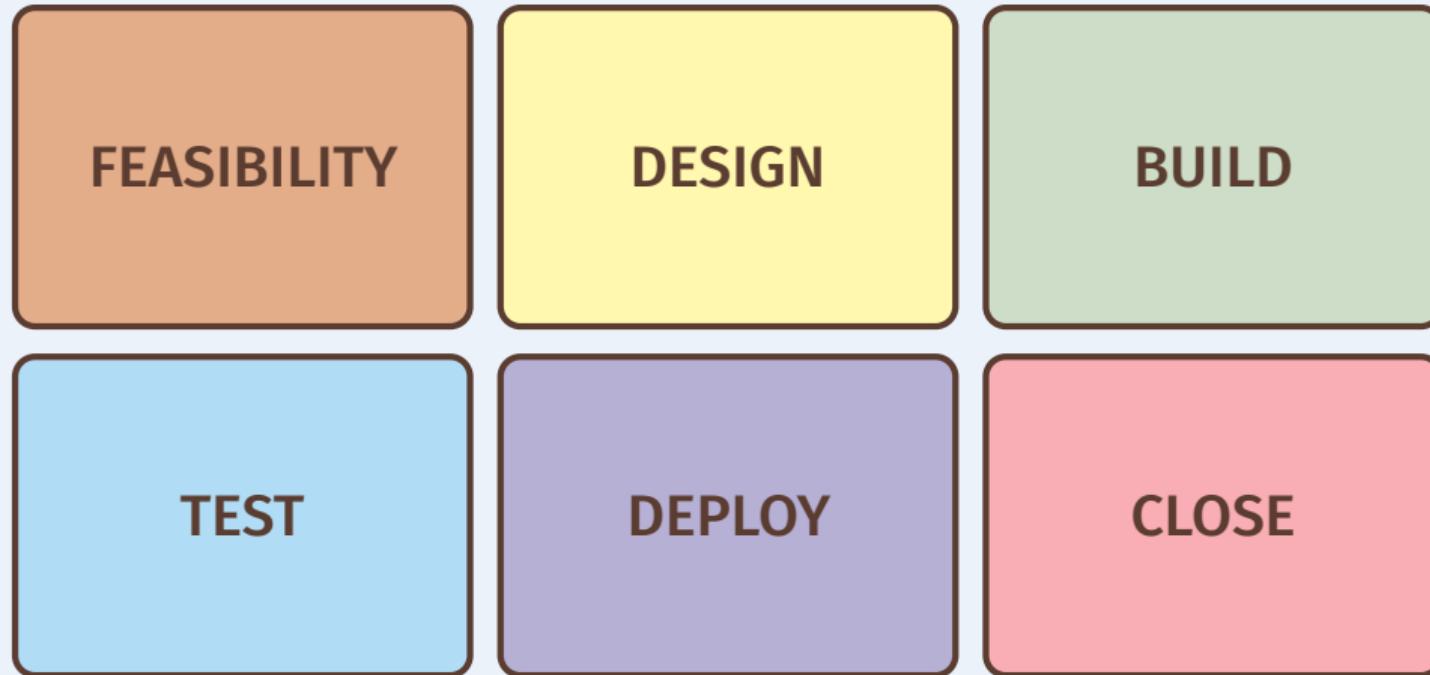
Choice of a development approach

- 1.** project variables (stakeholder, time, and cost)
- 2.** organization (structure, culture, policies, size)
- 3.** outcome (product, service, or result)

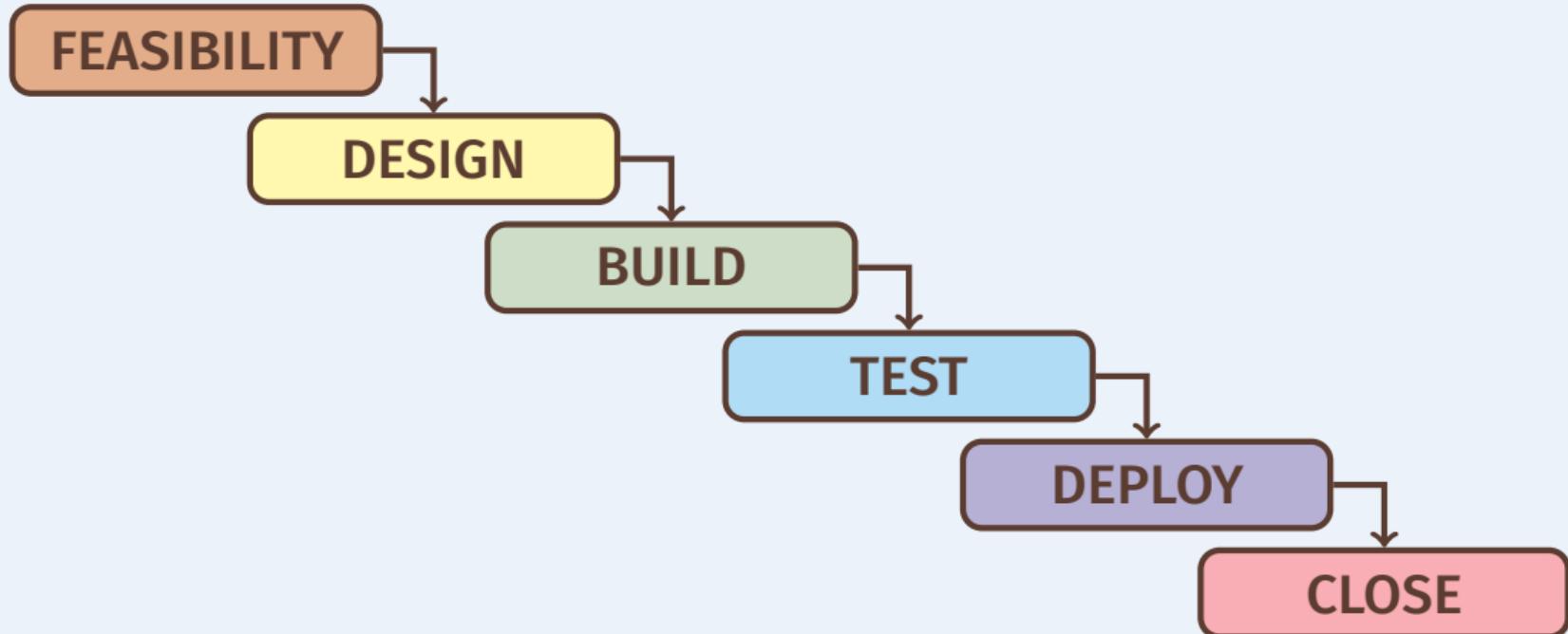
Choice of a development approach

1. project variables (stakeholder, time, and cost)
2. organization (structure, culture, policies, size)
3. outcome (product, service, or result)
 - ▶ degree of innovation
 - ▶ clarity of requirements
 - ▶ stability of scope
 - ▶ inherent risk
 - ▶ safety requirements
 - ▶ relevant regulations

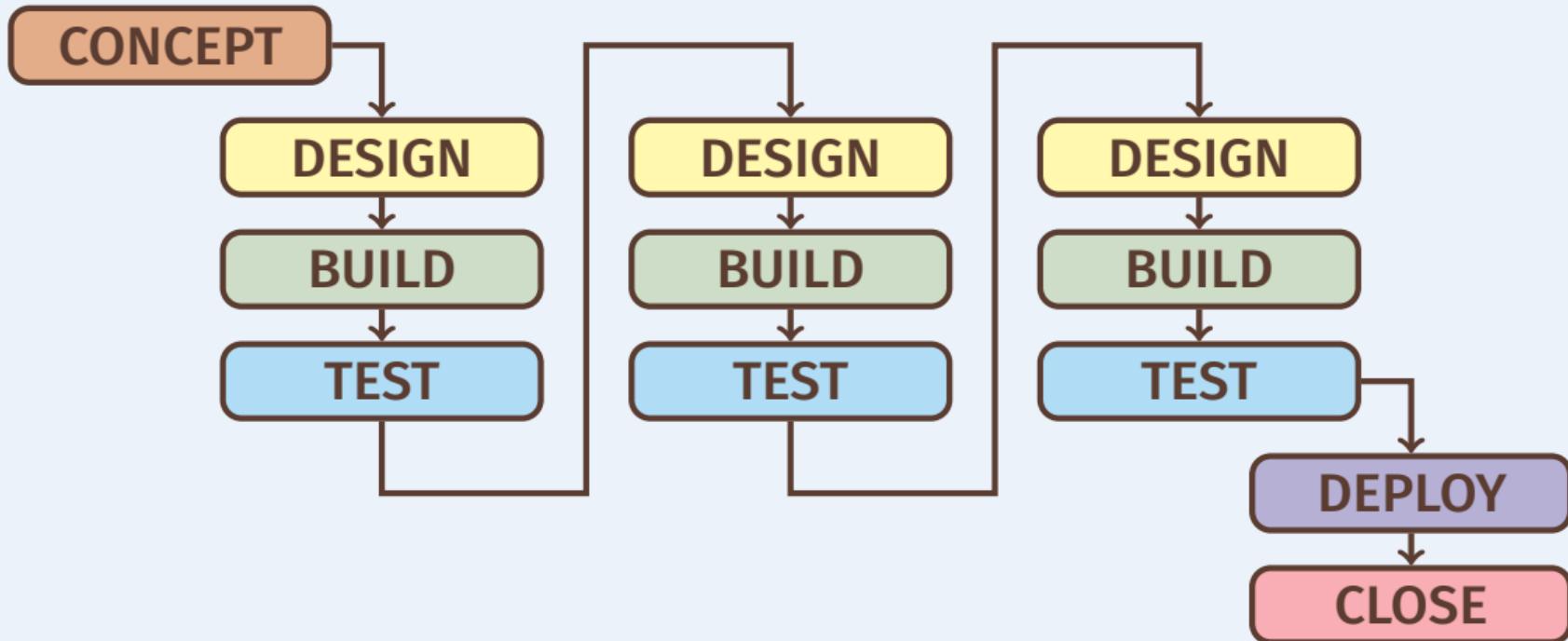
The six phases of a project's life cycle



A predictive life cycle



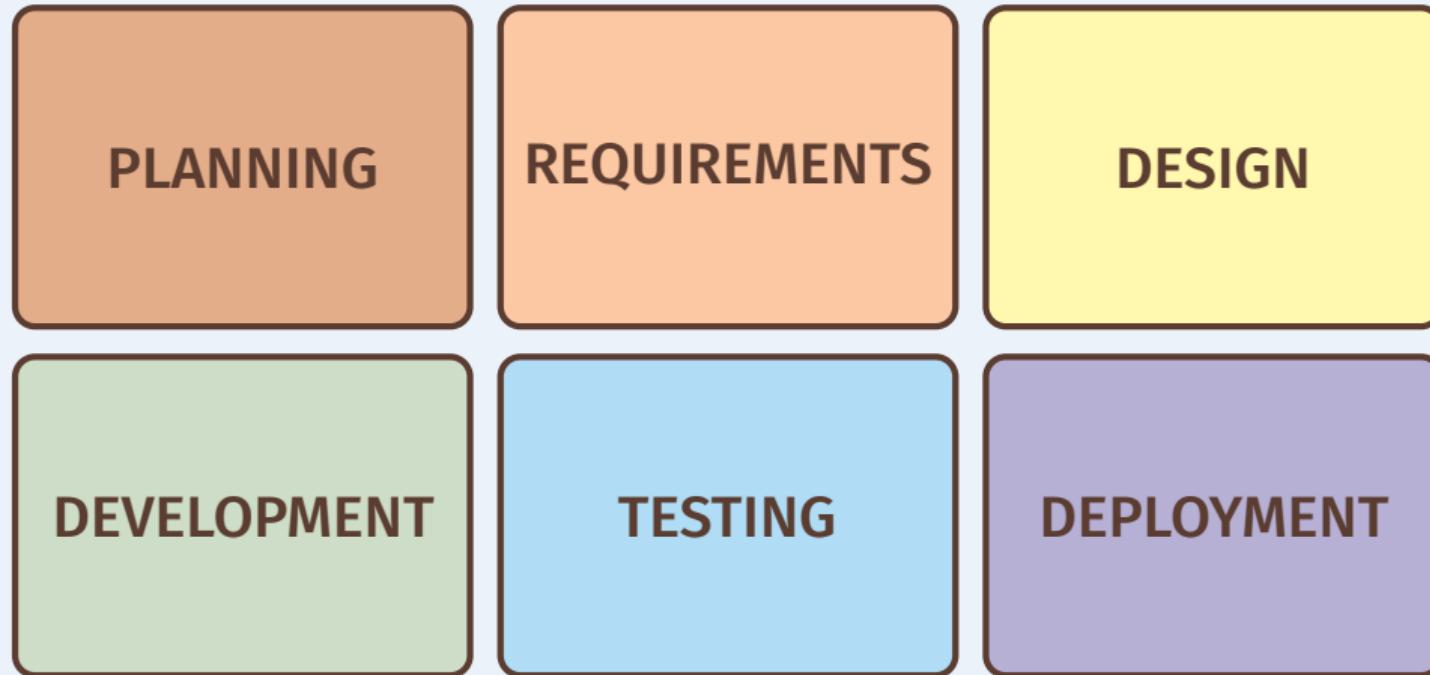
An adaptive life cycle



Can be **iterative** or **incremental**.

THE SOFTWARE ENGINEERING PERSPECTIVE

Software development life cycle (SDLC)



THE WATERFALL APPROACH

MANAGING THE DEVELOPMENT OF LARGE SOFTWARE SYSTEMS

Dr. Winston W. Royce

INTRODUCTION

I am going to describe my personal views about managing large software developments. I have had various assignments during the past nine years, mostly concerned with the development of software packages for spacecraft mission planning, commanding and post-flight analysis. In these assignments I have experienced different degrees of success with respect to arriving at an operational state, on-time, and within costs. I have become prejudiced by my experiences and I am going to relate some of these prejudices in this presentation.

COMPUTER PROGRAM DEVELOPMENT FUNCTIONS

There are two essential steps common to all computer program developments, regardless of size or complexity. There is first an analysis step, followed second by a coding step as depicted in Figure 1. This sort of very simple implementation concept is in fact all that is required if the effort is sufficiently small and if the final product is to be operated by those who built it — as is typically done with computer programs for internal use. It is also the kind of development effort for which most customers are happy to pay, since both steps involve genuinely creative work which directly contributes to the usefulness of the final product. An implementation plan to manufacture larger software systems, and keyed only to these steps, however, is doomed to failure. Many additional development steps are required, none contribute as directly to the final product as

use. It is also the kind of development effort for which most customers are happy to pay, since both steps involve genuinely creative work which directly contributes to the usefulness of the final product. An implementation plan to manufacture larger software systems, and keyed only to these steps, however, is doomed to failure. Many additional development steps are required, none contribute as directly to the final product as analysis and coding, and all drive up the development costs. Customer personnel typically would rather not pay for them, and development personnel would rather not implement them. The prime function of management is to sell these concepts to both groups and then enforce compliance on the part of development personnel.

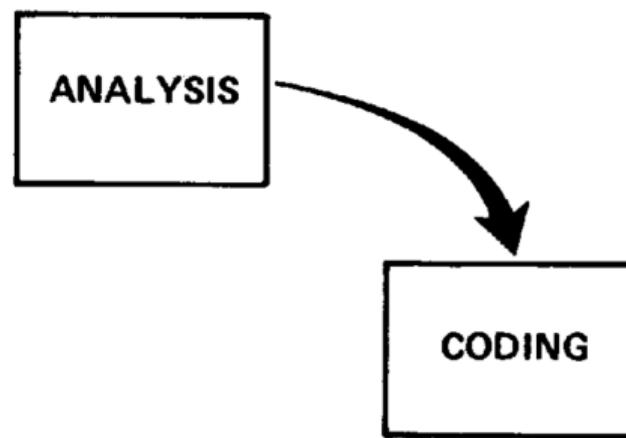


Figure 1. Implementation steps to deliver a small computer program for internal operations.

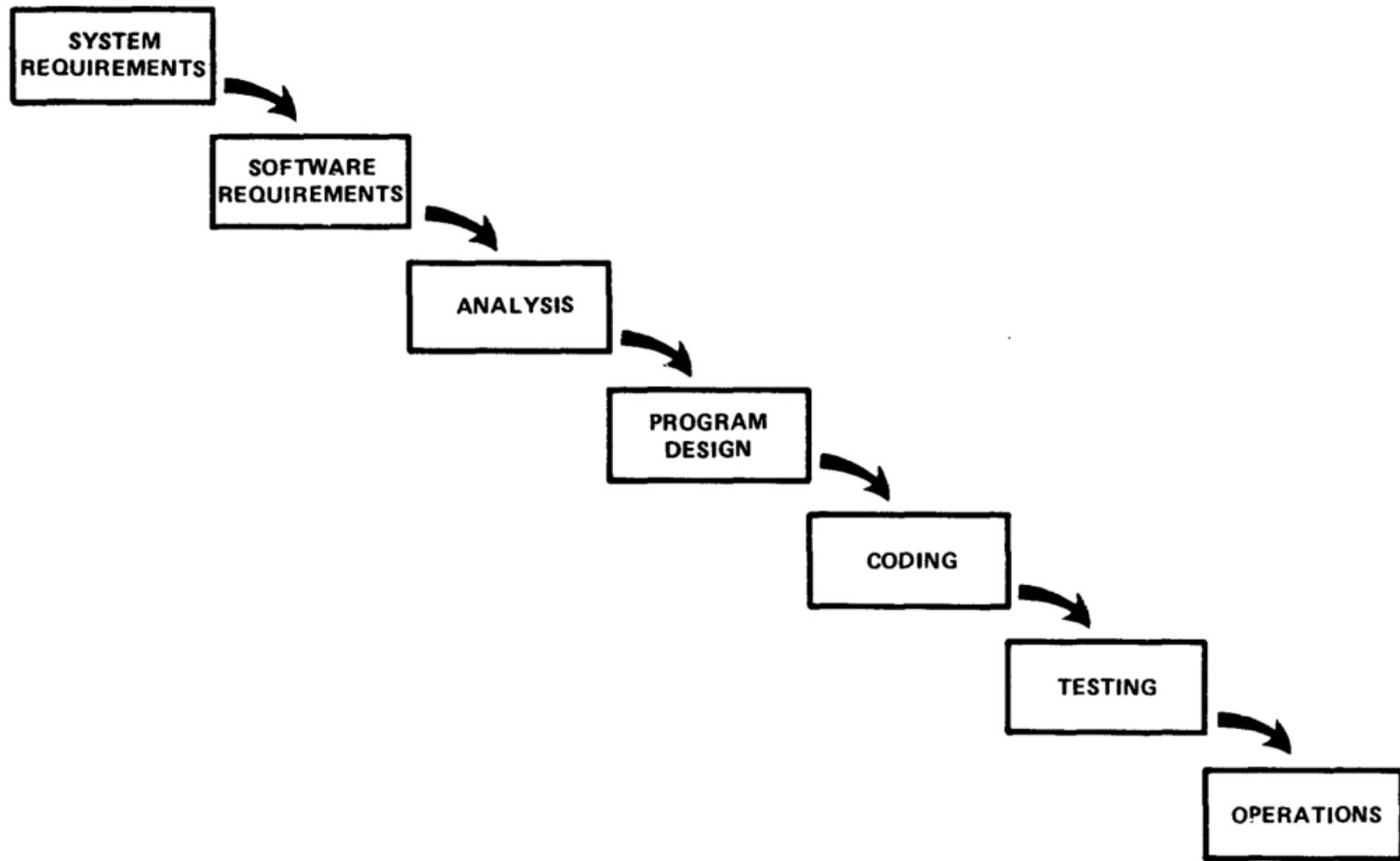


Figure 2. Implementation steps to develop a large computer program for delivery to a customer.

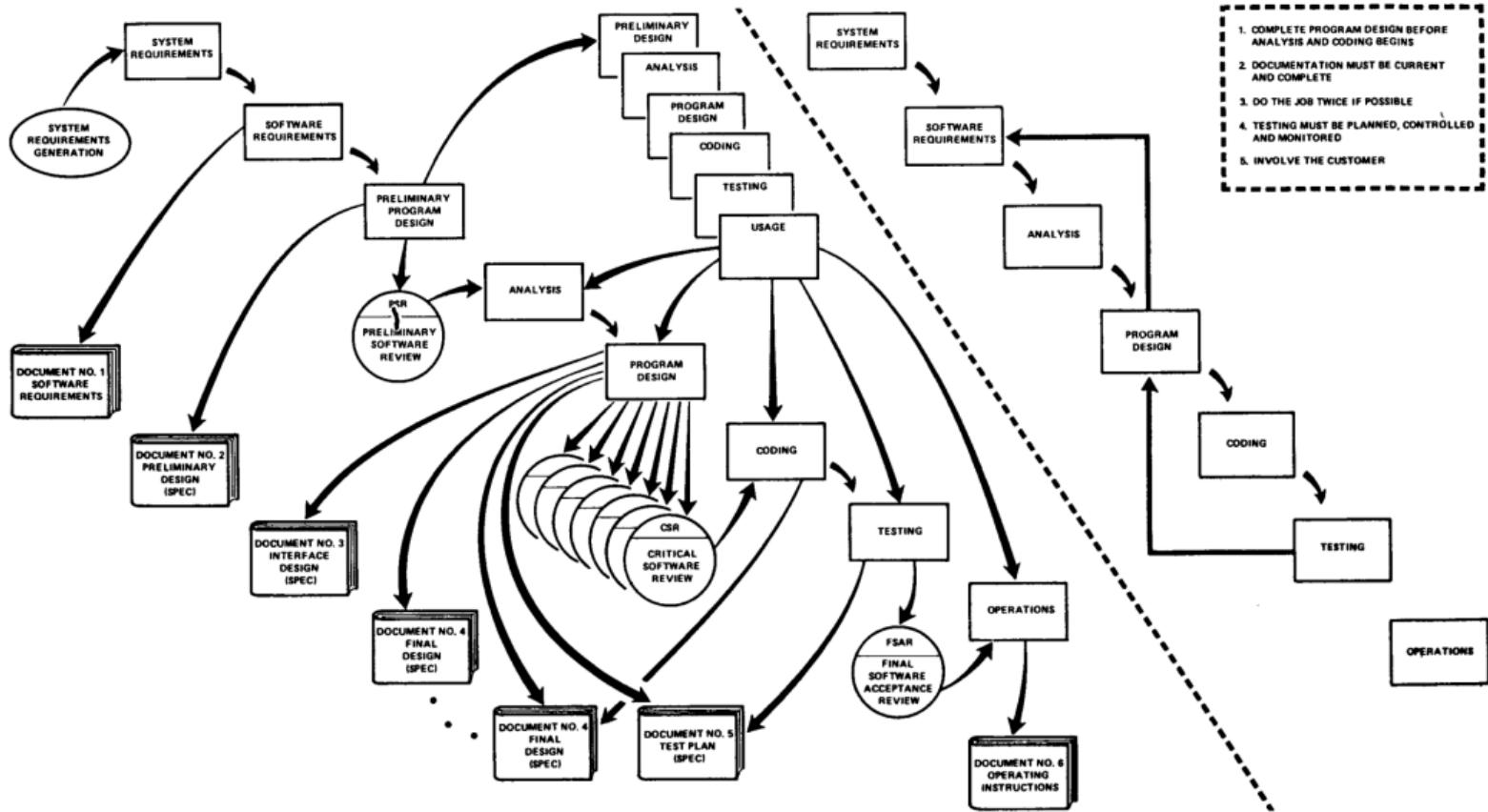


Figure 10. Summary

THE AGILE METHODOLOGY



We are uncovering better ways of developing software by doing it and helping others do it.

Through this work we have come to value:

Individuals and interactions over processes and tools

Working software over comprehensive documentation

Customer collaboration over contract negotiation

Responding to change over following a plan

That is, while there is value in the items on the right, we value the items on the left more.

Kent Beck

Mike Beedle

Arie van Bennekum

Alistair Cockburn

Ward Cunningham

Martin Fowler

James Grenning

Jim Highsmith

Andrew Hunt

Ron Jeffries

Jon Kern

Brian Marick

Robert C. Martin

Steve Mellor

Ken Schwaber

Jeff Sutherland

Dave Thomas

Summary

1. What are the most common project development approaches?
2. When is a predictive approach most suitable?
3. What is the most popular predictive approach?
4. What is the most popular adaptive approach?