

# Process Flows

Alex Sverdlov  
alex@theparticle.com

## 1 Introduction

There are essentially two extreme methodologies for building software: linear and iterative.

### 1.1 Linear Process Flow

When using linear process flow, we sub-divide software construction into phases. Each phrase represents a big-step that must be complete before next phrase starts.

The first phase is often requirements gathering, followed by analysis and/or design, followed by implementation, integration, deployment and maintenance.

Each phase proceeds after the last, with no official backtracking to re-examine previous decisions/phases.

The advantages of linear flow is that development does not start until requirements are solidified, and design is complete. Often the users sign-off on the requirements document before implementation begins.

The disadvantage is that requirements and/or design often do change after development starts. Development cannot start until requirements are solidified, which is often inefficient.

### 1.2 Iterative Process Flow

We can imagine the other extreme—where we start all-phases at mostly-the-same-time. We get a bit of requirements, implement whatever we can, run the partial implementation by the users to get more requirements and/or clarification, and then repeat the cycle—refining it with every iteration.

The other approaches are just variations of this scheme—with user feedback happening in months vs weeks, etc., or with business users being on the development team (collaboration vs contracts)—to enable quick turn-around on development direction.