Incremental & Iterative Development

Figures are from: “Agile and Iterative Development: A Manager's Guide”, Craig Larman Addison Wesley, 2004
Incremental & Iterative Development

• **Iterative** development: an approach to building software (or anything) in which the overall lifecycle is composed of several iterations in sequence. Each iteration is a self contained “mini-project” of development activities.

• **Incremental**: the act of growing the system incrementally with new features, iteration by iteration.

• In other words… **incremental & iterative development**.
Iterative & Incremental Development

Feedback from iteration N leads to refinement and adaptation of the requirements and design in iteration N+1.

Build for some requirements → feedback → Build for some requirements → feedback → Build for some requirements

a 3-week iteration

The system grows incrementally.

RELEASE TO CUSTOMERS
Product Delivery

• **Incremental Delivery**: the practice of repeatedly delivering a system into the marketplace in a series of expanding capabilities. (future prediction)

• **Evolutionary Delivery**: a refinement of incremental delivery in which a vigorous attempt to capture feedback regarding the installed product and use it to guide the next delivery. (feedback)

• In practice, a marriage of some future prediction and feedback is obvious and common, and the two terms are often used interchangeably.
Disciplines Across Iterations

A 3-week iteration (for example). A mini-project that includes work in most disciplines, ending in a stable executable.

Note that although an iteration includes work in most disciplines, the relative effort and emphasis change over time.
Timeboxing

In timeboxing, the variable of time in each iteration is fixed.

A 3-week timeboxed iteration. The end date may not slip.

A 2-week timeboxed iteration. The end date may not slip.

Time, scope, resources, and quality are four common variables that we can play with on a project.

Timeboxing removes time from the options, during an iteration.
Evolutionary & Iterative Requirements

Imagine this will ultimately be a 20-iteration project.

In evolutionary iterative development, the requirements evolve over a set of the early iterations, through a series of requirements workshops (for example). Perhaps after four iterations and workshops, 90% of the requirements are defined and refined. Nevertheless, only 10% of the software is built.
Cone of Uncertainty

many estimates and schedules are prematurely defined at this time

but this is the realistic period for reliability
Multiple Phase Contracts

- 3 requirements workshops
- Software development to build a core architecture and obtain information for future estimation.
- 10% of the final software
- Most high-level requirements
- Realistically, only 75% of the detailed requirements, or less

Phase 1. Fixed Time, Fixed Price. e.g., 8 weeks, $200,000

Phase 2. Fixed Price
Incremental Delivery with Iterations

Jan - June
Incremental Delivery 1
- 4 weeks: iteration 1
- 2 weeks: iteration 2
...
- iteration 10
  deliver

July - Dec
Incremental Delivery 2
- 3 weeks: iteration 1
- 3 weeks: iteration 2
...
- iteration 7
  deliver