Agile Estimating and Planning

Mike Cohn

Some of the questions answered in Cohn’s text

“Everywhere in the agile world I hear the same questions:

- How do I plan for large teams?
- What size iteration should we use?
- How should I report progress to management?
- How do I prioritize stories?
- How do I get the big picture of the project?”

Preface

“His book adds to our knowledge of “how” to do estimating and planning, and it adds to our knowledge 0f “why” certain practices are important.”

Foreword, Part 1

More from the *Foreword – Part 2*

“Agile developers essentially say:

- We will give you a plan based on what we know today
- We will adapt the plan to meet our most critical objective
- We will adapt the project and our plan as both move forward and learn new information
- We expect you to understand what you are asking for…

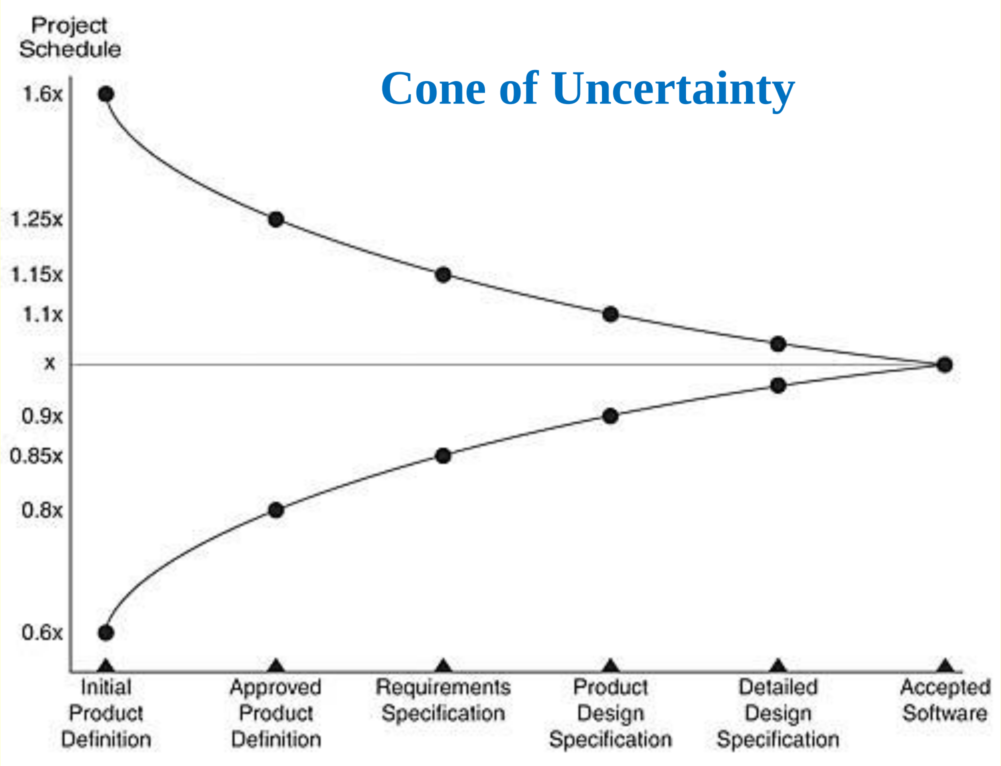
…that flexibility to adapt to changing business conditions **and** absolute conformance to original plans are incompatible objectives.”

Part I

The Problem and the Goal

Chapter 1: The Purpose of Planning

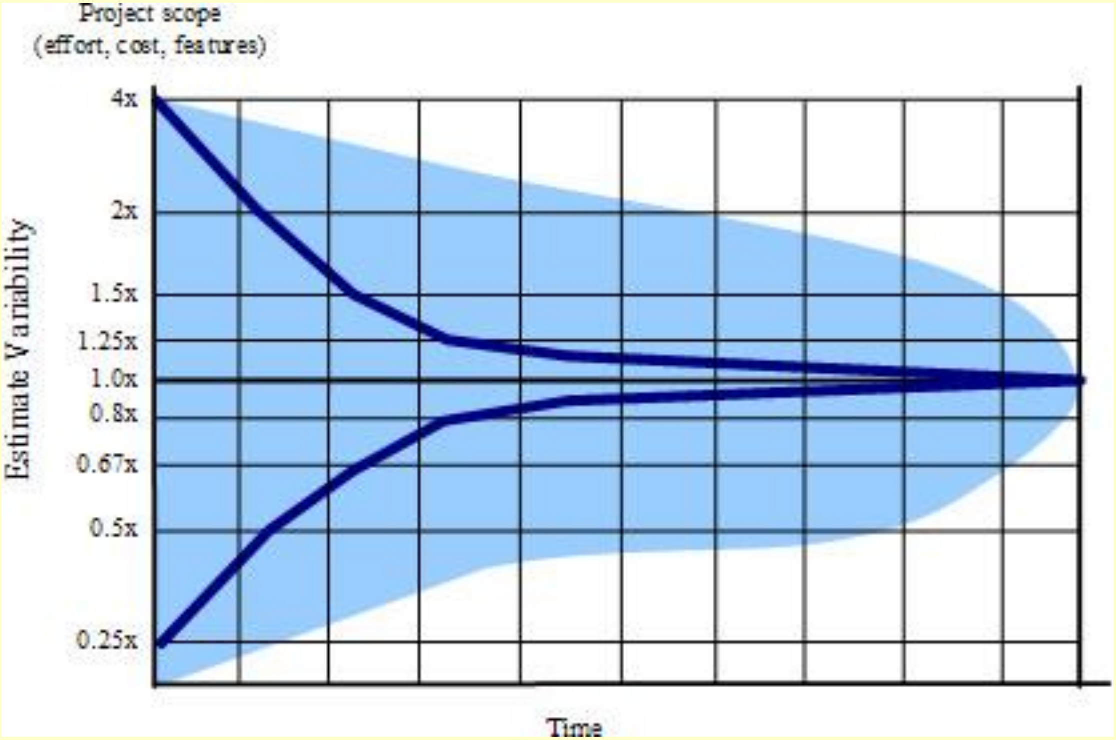
> "Planning is everything, Plans are nothing."
>
> Field Marshal Helmuth Graf von Moltke



Expected variation of estimates of completion at each phase

The cone narrows as the project progresses… with "progressive refinement".

Cloud that persists to the end of the project



The issue isn’t really that the estimates don’t converge

… the issue is that the project itself doesn’t converge,

… that is, it doesn’t drive out enough variability to support more accurate estimates.

The Cone narrows only as you make decisions that eliminate variability.

What should “we” build?

“The question cannot be answered all at once.”

The answer comes “iteratively and incrementally”

At outset, a specific set of features, resources and schedule is identified.

But:

- At slightly later date more features will be better… or
- At slightly sooner date slightly fewer features will be better

Estimating and Planning

- **Not** about determining single point estimates for deadline or schedule!
- Planning is a process… to determine what should be built.

What to build = f (features, resources, schedule)

- A good planning process
  - Reduces risk
  - Reduces uncertainty
  - Supports better decision making
  - Establishes trust
  - Conveys information

Reducing Risk

- The process of estimating should “expose” uncertainties… things that you nothing about.
- The process should, as a result, identify what you do now know… and therefore what you have no idea of how to estimate…

  and therefore what you need to learn.

Reducing Uncertainty

“Throughout a project, the team is generating new capabilities in the product…

… generating new knowledge about the product, the technologies, and the capabilities of the team.”

“The most critical risk facing most projects is the risk of developing the wrong product.”

Reference to the Standish Group…

Author’s definition of a failed project (one criterion):

“a project on which no one came up with any better ideas than what was on the initial list of requirements.”

Supporting better decision making

What types of decisions?

- Identifying project that are worth doing (benefits versus costs)
- Prioritizing, with high value projects at the top
- Ensuring that appropriate staffing and resources are available
- Making tradeoffs
  - between development time and cost
  - between functionality and effort, cost, and time

Establishing trust

“Frequent reliable delivery of promised features builds trust between the developers… and the customers.

Reliable estimates enable reliable delivery.”

“Customers are reluctant to make these types of trade-off decisions early in a project unless the developers’ estimates have proved trustworthy.”

Benefit of reliable estimates to developers…

- Work is done at a sustainable pace.
- Results are: higher quality code ad fewer bugs…

  Consequently, less time is spent on unpredictable work

  - Work is done at a sustainable pace.
  - Results are: higher quality code ad fewer bugs…

    Consequently, less time is spent on unpredictable work

Conveying information

"A **plan** conveys expectations and describes one possibility of what may come to pass over the course of the project."

Not a guarantee of specific features delivered on a specific date…

Should be understood as a set of ***baseline*** expectations…

Subject to change, as needed.

What makes a good plan?

- Depends on its use value.
- Sufficiently reliable so that it can be used as the basis for making decisions
  - Early on, it might specify a release date in the 3rd quarter with an approximate set of features
  - Later on, the plan will need to be more precise
- A good planning process provides for regular updates throughout the life of the project.

What makes planning Agile?

- Planning is an activity!
- Planning "…balances the effort and investment in planning with the knowledge that it will be revised throughout the life of the project."

  The plan changes

  "… because change means we've learned something or that we've avoided a mistake.

  We may have learned that users want more of this feature or that they want less of that feature or that usability is more important than we'd believed or that programming in this new language takes longer…"
- The Plan is easy to change

  … change a feature, reduce the scope of a feature, add people, etc.

Reality: you cannot totally define a project at the outset!

Agile planning:

- Is focused more on the planning process than on a plan
- Encourages change
- Results in plans that are easily changed
- Is spread throughout the project