DRAGON AGE
ORIGINS

Improving Programming Estimates

Adriana Lopez
Development Director

Why Are We Here?

Individual Estimate Improvement

A Question

- Have you ever been asked to provide an estimate for a task…
  - only to be proven horribly wrong (>2x)?

- What were some of the consequences you experienced as a result?

What Were the Consequences?

- Lower quality games
- Poor coordination with other groups
- Loss of credibility with your managers or peers

- Oh, and Crunch Time
  - Lots and lots of crunch time

Overview

- The Basics
  - What is an estimate?
  - Why is estimating hard?
- How do we make estimating easier?
- Summary
- Questions?

What is an Estimate?

An Estimate…

- Is a prediction of effort or complexity for a task.
  - A task should result in a concrete deliverable.
- Should reflect inherent uncertainty:
  - "I think this task will take between 3 and 8 days."
- Should be provided by the people doing the work.
  - *We are responsible for improving our ability to estimate our work accurately and precisely*.

An Estimate...

- Has both accuracy and precision.
  - An ***accurate*** estimate is one where the actual value falls within the range of the estimate.
    - Avoid ranges that are too big to be of value.
  - A ***precise*** estimate is one with a narrow range.
    - Avoid misleading precision not supported by data.

An Estimate…

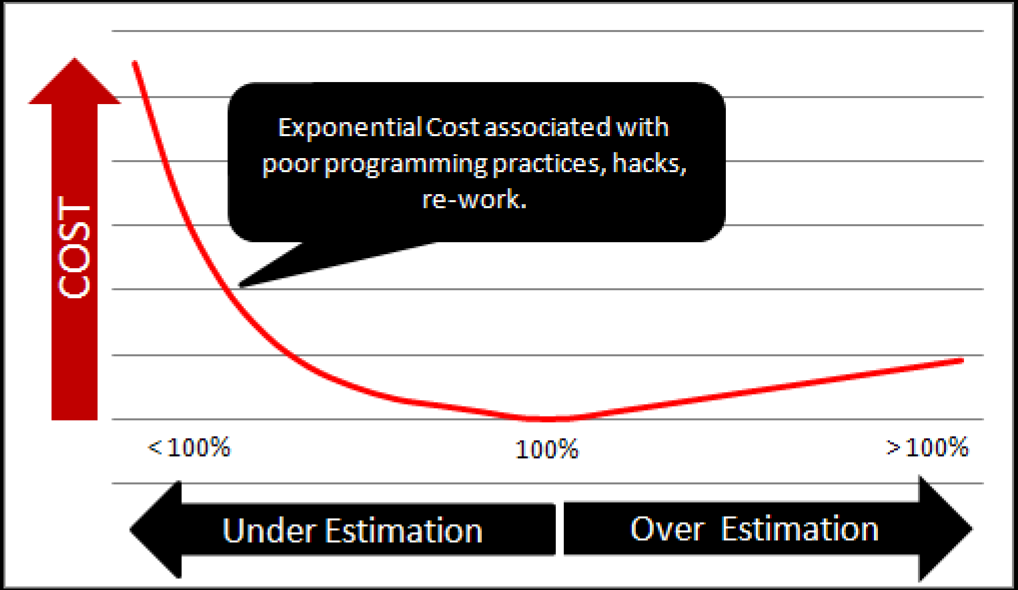
- Allows the project to plan and organize its resources.
  - Accurate and precise estimates allow the project to be completed on time and predictably.

Why is Estimating Hard?

Under- vs. Over-Estimation

- Studies show that programmers typically underestimate the work required to complete a task.

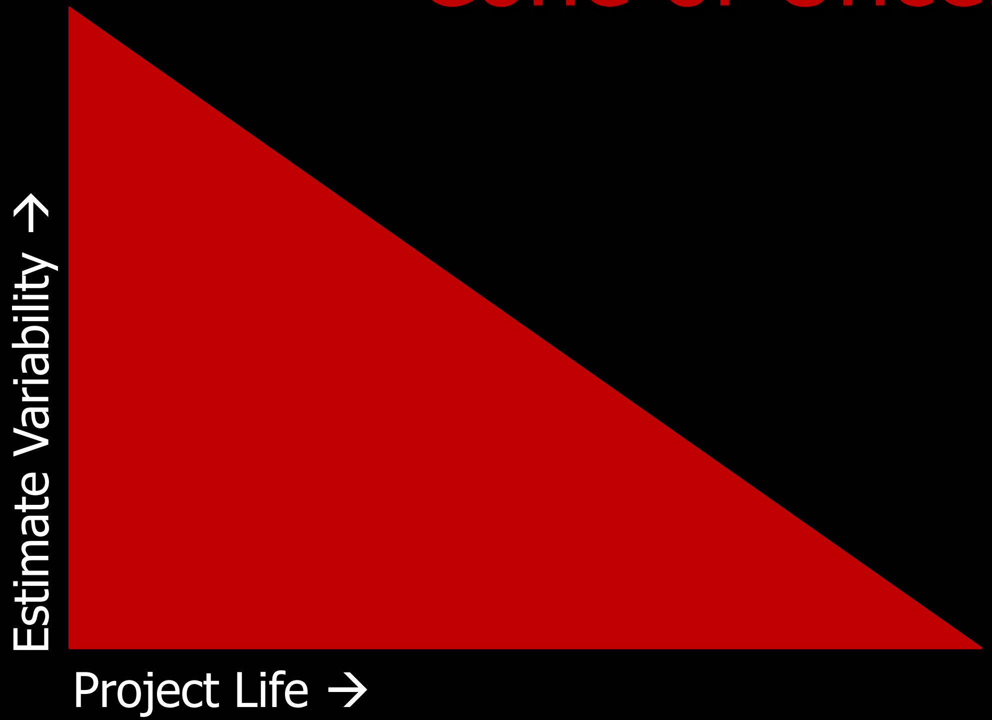
- "If you want a high-quality product OUT OF QA, you need to put a high-quality product INTO QA" - Humphreys



Incomplete Requirements

- It’s too early in the project.
  - Estimating how long “it” will take before anyone knows what “it” is.
- User doesn’t know what they want.
  - I’ll know “it” when I see “it”.
- We haven’t asked the right questions.
  - Remember to get both ***functional*** and ***non-functional*** requirements.
- Differentiate between Incomplete vs. Unstable Requirements

Cone of Uncertainty



Project Management term used to describe the level of uncertainty existing at different stages of a project.

Term popularized by Steve McConnell.

Individual Estimation Practices

- Pressure to Provide Low Estimates
  - Management needs to meet a target.
  - Peer pressure.

- Remember:
  - Lower Estimate != Better Programmer
  - Avoid "Off-the-Cuff" Estimates

Individual Estimation Practices

- Omitted Activities
  - Programming Activities
  - Non-Programming Activities
- Unfounded Optimism
  - A justification to support an estimate that is not based on data.
  - "We're smarter now"
  - "It can't be that bad"

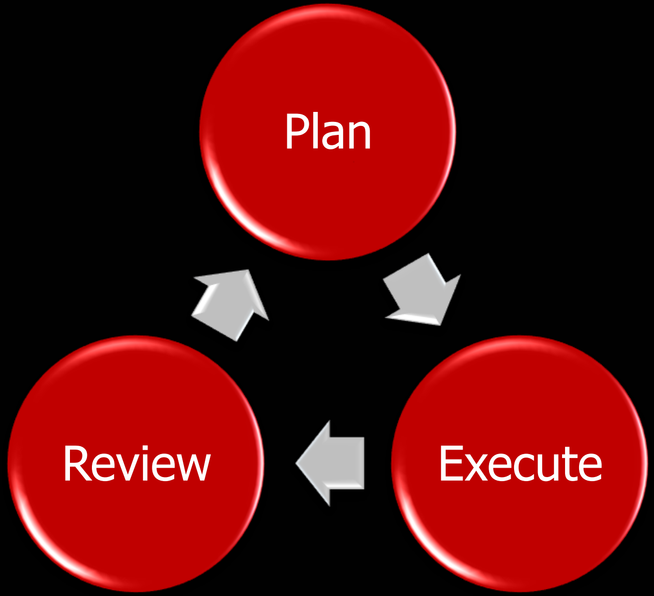
Individual Estimation Practices

- Subjectivity / Bias
  - A desire (conscious or unconscious) to achieve a particular outcome.
- Unfamiliar Problem Domain
  - Lack of experience.
- Lack of Historical Data
  - Base your estimates on past performance.

Project Chaos

- Differentiate between targets and estimates
  - Determine whether you are being asked for a realistic estimate or for a way of meeting a target.
- Do not provide "off-the-cuff" estimates
  - "Off-the-cuff" estimates can become targets / commitments.
- Learn to negotiate the requirements

How do we make it easier?



Planning

What are you getting yourself into?

What are you getting yourself into?

- The project leads are concerned about the total workload required to complete the agreed upon scope.
- Total workload is the sum of individual tasks and is based on how complex / how big **you think** the work is.
- Use estimation techniques to figure that out.

Technique 1: Establish a baseline

- Complexity Estimates:
  - XP (Experience Points)
  - Complexity Points
  - Headaches
  - Beers
- Time Based Estimates:
  - Effort Hours
  - Ideal days

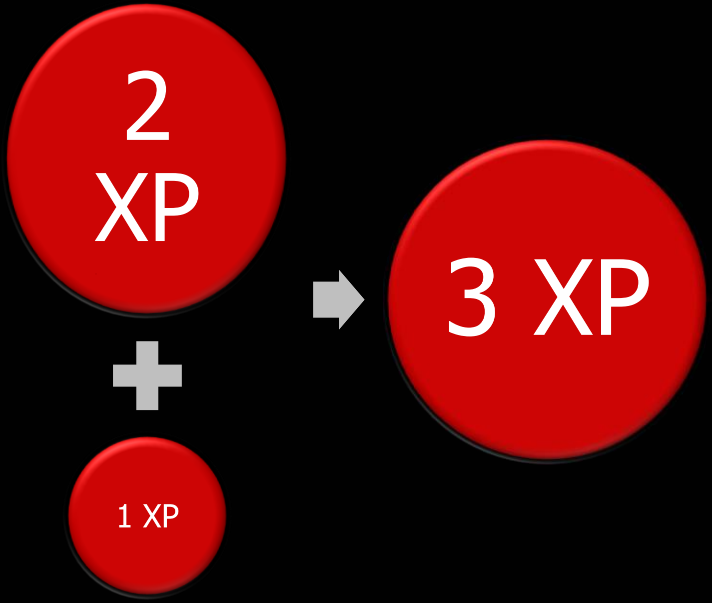
- Allow ranges: 1,2,3,5,8,13,20…
- Actual work is measured in real time (e.g. hours).

Technique 2: Collaboration

- Lowers personal bias.
- Without collaboration:
  - The estimate will depend upon who is / when asked.
  - Do not assume that the most reliable estimates come from the people with the most powerful vocal chords.

Technique 3: Triangulation

- Compare New Work with Known Past Work.
- Requires Historical Data.
- Be Consistent !



Technique 4: Decomposition

- Breakdown large tasks / user stories into smaller tasks / user stories that:
  - You're more familiar with
  - Are no bigger than a few days of work

A – 2 XP

B – 3 XP

C – 1 XP

Planning

- Use these techniques to arrive at an estimate that minimizes the impact of:
  - Omitted Activities
  - Unfounded Optimism
  - Subjectivity
  - Bias
  - Unfamiliar Problem Domain

Planning

- Remember to record your estimate!
  - To have data from previous tasks, you need to start writing your estimates down.

- Use it as a tool to help **you** monitor and improve **your** own performance during execution.

Execution

80% of the time is spent doing 20% of the work

Record Actual Time Spent

- The most effective way to deal with many of the Individual Estimation problems.
- Depends upon having a clear and unambiguous task.
- A clear task allows you to differentiate between defects and changes.

Where is your time going?

- **Exercise:** Try tracking your time for your next user-story or Task. Observe where you spend your time.

Review

How did I do?

How did I do?

- Ask questions:
  - How fast are you working?
  - How close was your estimate to the actual result?
  - Did I forget to anticipate anything that would be useful to know when estimating my next task?

Analyze the Results

- Useful Calculations
  - Velocity
    - How fast are you working?
  - Magnitude of Relative Error (MRE)
    - How close was your estimate to the actual result?

Example – Part 3a, Velocity

- Velocity is Work / Unit Time
  - Points: 7 XP / 3 calendar days = 2.3 XP / day
  - Effort: 9.5 hrs / 3 calendar days = 3.2 hrs / day

Date	Task	Original Estimate	Total calendar days	Total Velocity
4/7/2008	A	3 XP	5	
4/8/2008	B	2 XP	2	
4/9/2008	C	5 XP	8	0.66 XP /day

Example – Part 3b, MRE

- MRE is ABS (Actual – Expected) / Actual
  - ABS (9.5 – 7) / 9.5 = 0.26

Task	Best	Expected	Worst	Actual	MRE
A				1	
B				5	
C				10	
D	5	7	10	9.5	***26%***

Update Future Estimates

- Don’t update the original estimate when the task is complete just to get a better result!

- It is to be expected that estimates change over time.
  - Requirements are refined or change altogether as the project moves forward.

- Remember the “Cone of Uncertainty”

Summary

- It is our INDIVIDUAL responsibility to improve the accuracy and precision of our estimates.
  - Watch for Bad Individual Estimation Practices
  - Use Collaboration Techniques
  - Collect and Analyze Data
  - Use Analysis to Improve Your Future Estimates

Summary

- Find a method that works for you.
  - (Make sure it will still work with your project).

¿Questions?

"It is very difficult to make a vigorous, plausible, and job-risking defense of an estimate that is derived by no quantitative method, supported by little data, and is certified chiefly by the hunches of the managers."

Fred Brooks

References

- 10x Software Engineering course offered by Construx Software (http://www.construx.com)
- Software Estimation by Steve McConnell
- Mountain Goat Software (http://www.mountaingoatsoftware.com/)