Lessons in Estimating

Agile vs. Waterfall

Agile and Waterfall

Jerry Richardson, PMP
Sohail Thaker, PMP
“Prediction is very difficult, especially about the future.”

- Niels Bohr, Danish physicist
Workshop Overview

- Why do we estimate?
- Developing Waterfall estimates
- Developing Agile estimates
- When to use each technique
- Keys to success
Estimating vs. Planning

Plan a Project

- Develop Project Framework
- Determine Resource Requirements
- Prepare a Budget
- Develop a Schedule
- Develop Work Plan
  - Tasks
  - Estimates
- Develop Project Controls
Why Develop Work Plan?

- Funding
- Resources
- Expectations

➢ It’s management’s job to vet our plan!

“Realistic & Defensible”
Why Develop Work Plan?

• Who needs to be involved?

• What will each person do?

• When do certain tasks need to occur?
What Do We Estimate?

- Work Effort
- Elapsed Time
- Cost
Levels of Estimates

Project level estimate

Top Down

Bottom Up

Bottom Up

Bottom Up

Phase level estimates
Bottom Up Estimating Technique

1. Break job into tasks (WBS)
Bottom Up Estimating Technique

1. Break job into tasks (WBS)
2. Develop a task list
<table>
<thead>
<tr>
<th>Task</th>
<th>Deliverable</th>
</tr>
</thead>
<tbody>
<tr>
<td>Confirm Dates</td>
<td>Dates Confirmed</td>
</tr>
<tr>
<td>Make Reservations</td>
<td>Reserv’ns</td>
</tr>
<tr>
<td>Collect Travel Docs</td>
<td>Passport, E-ticket</td>
</tr>
<tr>
<td>…(Etc.)</td>
<td></td>
</tr>
</tbody>
</table>
Bottom Up Estimating Technique

1. Break job into tasks (WBS)
2. Develop a task list
3. For each task, estimate:
   - \( S \) – shortest effort
   - \( L \) – longest effort
   - \( M \) – most likely effort

\[ \text{Note: } M \neq (S + L)/2 \]
## Estimating Spreadsheet

<table>
<thead>
<tr>
<th>Task</th>
<th>Deliverable</th>
<th>Assumptions</th>
<th>S</th>
<th>L</th>
<th>M</th>
</tr>
</thead>
<tbody>
<tr>
<td>Confirm Dates</td>
<td>Dates Confirmed</td>
<td>Boss in office</td>
<td>5</td>
<td>20</td>
<td>5</td>
</tr>
<tr>
<td>Make Reservations</td>
<td>Res’ns</td>
<td>In-house travel agent</td>
<td>15</td>
<td>30</td>
<td>20</td>
</tr>
<tr>
<td>Collect Travel Docs</td>
<td>Passport, E-ticket</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>…(Etc.)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
4. Calculate the best estimate, $E$

$$E = \frac{S + L + 4M}{6}$$
<table>
<thead>
<tr>
<th>Task</th>
<th>Deliverable</th>
<th>Assumptions</th>
<th>S</th>
<th>L</th>
<th>M</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td>Confirm Dates</td>
<td>Dates Confirmed</td>
<td>Boss in office</td>
<td>5</td>
<td>15</td>
<td>5</td>
<td>7</td>
</tr>
<tr>
<td>Make Reservations</td>
<td>Res’ns</td>
<td>In-house travel agent</td>
<td>15</td>
<td>30</td>
<td>20</td>
<td>21</td>
</tr>
<tr>
<td>Collect Travel Docs</td>
<td>Passport, E-ticket</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>…(Etc.)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
4. Calculate the best estimate, \( E \)
   \[
   E = \frac{S + L + 4M}{6}
   \]

5. Add contingency (10-20%)
Contingency Factors

- Project size
- Project duration
- Project organization
- Technical complexity
- Stability of business area
- Project team experience
Levels of Estimates

Project level estimate

Top Down

Bottom Up

Phase level estimates

Copyright © 2011 Ethier All rights reserved.
Top Down Estimating

- All Top Down estimates are a range
- Techniques
  - Equation
    - % of effort for each phase
  - Comparison
    - similarity to another company project
  - Analogy
    - similarity to a database of projects
## Top Down Estimating

**Equation Approach – “Ball Park” Example**

<table>
<thead>
<tr>
<th>Stage</th>
<th>% of Total Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initiate</td>
<td>5 - 15%</td>
</tr>
<tr>
<td>Analyze</td>
<td>15 - 30%</td>
</tr>
<tr>
<td>Design</td>
<td>40 - 70%</td>
</tr>
<tr>
<td>Construct</td>
<td>10 - 25%</td>
</tr>
<tr>
<td>Deploy</td>
<td></td>
</tr>
</tbody>
</table>
Top Down Estimating

Equation Approach – “Ball Park” Example

Initiate 2 days
Analyze (est’d) 8 days

5% of total project = 10 days
  total project = 200 days

15% of total project = 10 days
  total project = 67 days

“Ball Park” Estimate = 70 – 200 days

Copyright © 2011 Ethier  All rights reserved.
Keys to Success

• Start with a “ball park” and compare to your bottom up estimate
• Use your team
• Ask for input
• Estimate next phase as part of current phase
• Document assumptions
• Verify spreadsheets
What is Agile?

- Agile Manifesto
- Many versions
- Focused on software development
- Incorporates many best practices
- Focus on “features”
- Business needs to be very involved
- Iterative delivery of features
Agile Iterations

WATERFALL

- Analyze
- Design
- Construct
- Deploy

AGILE

[Diagram showing iterative cycles]
• Developing Agile Estimates

• What is Velocity and how do you use it for Planning & Refining Estimates?
Law of Diminishing Returns

- The more effort we put into something, the better result. Right?

- Law of Diminishing Returns – additional estimation effort yields very little value beyond a certain point.
Estimating: Accuracy vs. Effort

Accuracy vs. Effort Graph

Accuracy

Effort
Agile Assumptions

- Estimates are shared
- Core team is dedicated
- Complete a feature/user story in one iteration
- Final product details are not completely known
- Business support
Ideal Days vs. Story Points
Point Estimating Scales

- Fibonacci Sequence
  - 1, 2, 3, 5, 8
  - (+ 13, 20, 40, 100 for epics, themes)
- T-Shirt Sizing
  - Small, Med, Large, X-Large
- 1, 2, 4, 8
Points Example

- Feature 1 - 8 points (M)
- Feature 2 - 8 points (M)
- Feature 3 - 3 points (S)
- Feature 4 - 13 points (L)
- Feature 5 - 3 points (S)
- Feature 6 - 40 points (XL)

... 

- ETC - NN points

| TOTAL | = 500 points for project |
Validating Story Point Estimates

• Time Estimates
  – Pick some sample user stories
  – Break down to task level
  – Estimate hours
  – Calculate how long it will take, in real hours, to develop a small, a medium, etc.

• Boundary Samples
Agile Estimating: Results

• Master Story List with points for each story
• Apply the real estimates to get an overall estimate of the size of the project
• Build in some contingency
• Relatively accurate estimates for next few iterations
• Minimized time estimating
What is Velocity?

- Velocity determination
- Self-correcting
Watch for Velocity Trends

• Example:
  – Estimated 500 points for a project
  – Team initially estimates they can complete 50 points per iteration
  – Initial estimate = 500/50 = 10 iterations to complete project
  – If first iterations average at 75 points, then team was too cautious and 500/75 = 7 iterations
  – If first iterations average at 40 points, then team was too optimistic and 500/75 = 13 iterations
Track Progress Using Points

• Total Points Completed, Points Remaining
• Percent of Points completed
• Number of iterations to Complete
• Number of weeks/months to complete
• Impact of changing resources to velocity
• PM can add in factors once trends established
Using Velocity to Plan

• Reduce scope
  – deselect some specific stories (features)

• Increase capacity (size) of team
  – add to team velocity

• Reduce cost by reducing resources
  – slow down velocity
Agile: Pros & Cons

PROS
• Fail Fast
• Working deliverables provided frequently
• Don’t need fully detailed requirements to start
• Tight feedback loop allows for requirement changes throughout the project

CONS
• No detailed design phase makes it difficult to estimate the cost and duration of the whole project
• Best applied to software development
**Waterfall: Pros & Cons**

**PROS**
- Detailed design phase gives better understanding to estimate the duration of the whole project
- Detailed design done by the experienced developers allow for more inexperienced developers (or outsourcing of development)
- Enforced discipline by clear phasing with defined start and end points (milestones)

**CONS**
- Big overhead caused by detailed early-phase planning, as plans tend to change
- Limited working deliverables until the very end of the project
Choosing an Approach

<table>
<thead>
<tr>
<th>ONLY WATER -FALL</th>
<th>DECIDE ON BEST APPROACH FOR SITUATION</th>
<th>ONLY AGILE</th>
</tr>
</thead>
</table>

Copyright © 2011 Ethier  All rights reserved.
How Do We Choose?

- Buy vs. Build (Waterfall, Agile, Blended)

  - Business Availability
  - Organization Maturity
  - Technical Complexity
  - Understood vs. New
  - Market Maturity
  - Competitive Advantage
  - Logical Feature Sets
## Selection Criteria

<table>
<thead>
<tr>
<th>Business Availability</th>
<th>Buy</th>
<th>Build Waterfall</th>
<th>Build Agile</th>
<th>Don’t Do It</th>
</tr>
</thead>
<tbody>
<tr>
<td>Very Low</td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Low / Medium</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Medium / High</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Business Maturity</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Start-Up Organization</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mature Organization</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
</tbody>
</table>
## Selection Criteria

<table>
<thead>
<tr>
<th>Technical Complexity</th>
<th>Buy</th>
<th>Build Waterfall</th>
<th>Build Agile</th>
<th>Don’t Do It</th>
</tr>
</thead>
<tbody>
<tr>
<td>Low</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Medium</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>High</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Understood vs. New</th>
<th>Buy</th>
<th>Build Waterfall</th>
<th>Build Agile</th>
<th>Don’t Do It</th>
</tr>
</thead>
<tbody>
<tr>
<td>New Territory</td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Well Understood</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Selection Criteria</td>
<td>Buy</td>
<td>Build Waterfall</td>
<td>Build Agile</td>
<td>Don’t Do It</td>
</tr>
<tr>
<td>------------------------------------</td>
<td>-----</td>
<td>-----------------</td>
<td>-------------</td>
<td>-------------</td>
</tr>
<tr>
<td><strong>Market Maturity</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>No packages available</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>Some packages available</td>
<td></td>
<td>✔</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>Many packages available</td>
<td>✔</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Competitive Advantage</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Not required</td>
<td>✔</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Required</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Copyright © 2011 Ethier  All rights reserved.
## Selection Criteria

<table>
<thead>
<tr>
<th>Logical Feature Sets</th>
<th>Buy</th>
<th>Build Waterfall</th>
<th>Build Agile</th>
<th>Don’t Do It</th>
</tr>
</thead>
<tbody>
<tr>
<td>Small discrete groups</td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Big – Need All</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Look at all the aspects before selecting the appropriate approach
Both techniques have pros & cons

Agile **and** Waterfall – know when to use each technique

“Buy” when you can!
Some PM Truths Don’t Change…

• The more ridiculous the deadline the more money will be wasted trying to meet it.

• The first 90% of a project takes 90% of the time the last 10% takes the other 90%.
Thanks!

jrichardson
@ethier.ca

sthaker
@ethier.ca

www.ethier.ca