Why Size Doesn’t Matter

Daniel S. Vacanti
ActionableAgile™
daniel@actionableagile.com
@danvacanti
“When will it be done?”
“When will it be done?”

Date (number of days)

Elapsed Time
“When will it be done?”

Date (number of days)

Story Points / Velocity?
Stop me if you’ve heard this one before…
“Relative Complexity is the best predictor of how long it takes an item to complete”
<table>
<thead>
<tr>
<th>Stories</th>
<th>Points</th>
<th>InProgress</th>
<th>Ready for QA</th>
<th>QA</th>
<th>Ready for Acceptance</th>
<th>Acceptance</th>
<th>Holding</th>
<th>Total Days</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Closed</td>
<td>157</td>
<td>Total Closed</td>
<td>182</td>
<td>2.82</td>
<td>0.57</td>
<td>1.34</td>
<td>0.53</td>
<td>0.04</td>
</tr>
<tr>
<td>0 Points</td>
<td>9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Half Point</td>
<td>86</td>
<td>Half Point</td>
<td>43</td>
<td>0.95</td>
<td>0.48</td>
<td>0.50</td>
<td>0.47</td>
<td>0.02</td>
</tr>
<tr>
<td>1 Points</td>
<td>25</td>
<td>1 Points</td>
<td>25</td>
<td>2.60</td>
<td>0.56</td>
<td>1.40</td>
<td>0.36</td>
<td>0.04</td>
</tr>
<tr>
<td>2 Points</td>
<td>12</td>
<td>2 Points</td>
<td>24</td>
<td>5.50</td>
<td>0.42</td>
<td>2.00</td>
<td>0.33</td>
<td>0.00</td>
</tr>
<tr>
<td>3 Points</td>
<td>19</td>
<td>3 Points</td>
<td>57</td>
<td>8.00</td>
<td>1.21</td>
<td>4.17</td>
<td>1.32</td>
<td>0.05</td>
</tr>
<tr>
<td>5 Points</td>
<td>5</td>
<td>5 Points</td>
<td>25</td>
<td>9.20</td>
<td>0.40</td>
<td>2.60</td>
<td>0.70</td>
<td>0.00</td>
</tr>
<tr>
<td>8 Points</td>
<td>1</td>
<td>8 Points</td>
<td>8</td>
<td>25.00</td>
<td>n.n.</td>
<td>7.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>12 Points</td>
<td>0</td>
<td>12 Points</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Why do we even bother with Story Points?
So what does matter when it comes to predicting how long it takes an item to complete?
How long does it take you to get to work in the morning?
“It depends...”
Here’s a spot!!
Here's a spot!!
Try an experiment for me
Try the same thing for your process
<table>
<thead>
<tr>
<th>Backlog</th>
<th>Analysis</th>
<th>Develop</th>
<th>Test</th>
<th>Deployed</th>
</tr>
</thead>
</table>

Start Timer

Stop Timer
What we’ve just done is measure the flow metric of Cycle Time.
Cycle Time is the amount of elapsed time it takes for a given work item to complete.
“When will it be done?” for a single item is best answered by the flow metric of Cycle Time
Why Size Doesn’t Matter
Reason #1: Measuring the Wrong Thing
Why does the shape of our Cycle Time look like this?
<table>
<thead>
<tr>
<th>Backlog</th>
<th>Analysis</th>
<th>Develop</th>
<th>Test</th>
<th>Deployed</th>
</tr>
</thead>
</table>

[Image of yellow sticky notes on the left side of the table]
<table>
<thead>
<tr>
<th>Backlog</th>
<th>Design</th>
<th>Develop</th>
<th>Test</th>
<th>Deployed</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Will Item #3 end up…

Exactly Here?

Or Exactly Here?

Exactly Here?
We can’t think deterministically. We need to think probabilistically.
What does it mean to think probabilistically?
Thinking probabilistically means acknowledging there is more than one possible future outcome.
Think of Cycle Time as a shape and not a number.
When it comes to predictability, focus on how to affect the shape of your Cycle Time distribution
What factors affect shape?
Average Cycle Time = \frac{Average Work in Progress}{Average Throughput}

Cycle Time = how long it takes one item to go through the process

Work in Progress = how many items are in the process at any time

Throughput = how many items are produced per unit of time
Why Size Doesn’t Matter
Reason #2: Too Much WIP
What else?
Let’s try another experiment...
What if we could control WIP and make all items the same “size”?
Simulation Setup

• 50 Items in the backlog
• All work items need exactly 10 days of active work in each stage
• No other blocking events or added items
2
Analysis

Active | Done

2
Dev

Active | Done

1
Test
At the end of day 1

- Analysis
  - Active: 1
  - Done: 2

- Development
  - Active: 2
  - Done: 1
At the end of day 11
At the end of day 20

How do we choose which item to pull?
Baseline Case

Items are worked through the process in a strict FIFO queuing manner.
At the beginning of day 21

2
Analysis
Active
Done

2
Dev
Active

1
Test
Done

3

4

2

1
After 50 simulations, what do you think our expected Cycle Time will be at the 85th percentile?

(If you want to put a number on Cycle Time you have to give an associated confidence)
FIFO Queuing (no expedites)

85% 50 days
Some Expedite are Introduced

Baseline extended to use random queuing instead of strict FIFO queuing.
Choose one at random.
What will this do to Cycle Time? To variability? To predictability?
Random queuing (no expedites)

85% 60 days
Some Expedites are Introduced

Baseline case extended so that there is always one (and only one) expedite on the board. Strict FIFO queuing is reintroduced.

Expedites can violate WIP Limits and are always pulled in preference to other items.
Always choose the expedite
What do you think our expected Cycle Time will be at the 85th percentile for the non-escalated items?
FIFO queuing with always one expedite on board

85% 65 days
Some Expedites are Introduced

Baseline case extended so that there is always one (and only one) expedite on the board. Random queuing is reintroduced.

What will this do to standard class Cycle Time? To variability? To predictability?
Random queuing with always one expedite on board

85% 100 days
<table>
<thead>
<tr>
<th>Queuing Method</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Strict FIFO (no expedites)</td>
<td>50 days</td>
</tr>
<tr>
<td>Random Queuing</td>
<td>60 days</td>
</tr>
<tr>
<td>Strict FIFO always</td>
<td>65 days</td>
</tr>
<tr>
<td>one expedite:</td>
<td></td>
</tr>
<tr>
<td>Random Queuing always</td>
<td>100 days</td>
</tr>
</tbody>
</table>
Strict FIFO (no expedites)

Strict FIFO always one expedite

Random Queuing (no expedites)

Random Queuing always one expedite
Let’s look at these in reverse order as that is how most teams start out
Random queuing with always one expedite on board

Max Width: 130 days
Max Height: 7 items
FIFO queuing with always one expedite on board

Max Width: 40 days
Max Height: 27 items
FIFO Queuing (no expedites)

Max Width: 20 days
Max Height: 47 items
Why Size Doesn’t Matter
Reason #3: Poor Pull Policies
Real World Example
No WIP Limits, No Pull Policies, Doing Estimation

Max Width: 185 days
Max Height: 30 items
Limiting WIP, Pull Policies, No Estimation

Max Width: 90 days
Max Height: 62 items
To Sum Up
The biggest reasons why traditional sizing doesn’t matter:

- Measuring the wrong thing
- Too much WIP
- Poor Pull Policies

These items have a much bigger impact on elapsed time than estimation of size.
The good news is that these two things are directly under our control.
Take Control of the stuff you can control.

Go back to your team and Measure the right thing, Control WIP, Implement Sane Pull Policies, and…
Stop Estimating!
For next time...

What does a 19th century Yorkshire cotton industrialist have to do with the Manhattan Project?
Actionable Agile Metrics for Predictability
An Introduction

Daniel S. Vacanti

https://leanpub.com/actionableagilemetrics
All charts created by:

Actionable Agile
Take Control
Daniel S. Vacanti
http://www.actionableagile.com
daniel@actionableagile.com
@danzvacanti