Better Backlog Prioritization (from random to lifetime cost of delay)

The Goal – what to start next

Given a set of things we could do next, is one more economically advantageous to start first.

The Challenge – Doing one thing delays others

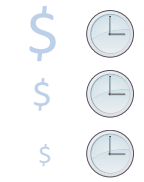
Every item has a different economic impact by being delayed. The impact will be a balance of lost value, and how long they are delayed.

"The problem with any prioritization decision is [it is a] decision to service one job and delay another." Don Reinertsen

The basic concept – balance $ & time

$ = Value lost due to delay 🕒 delay duration

Same duration, do > lost $ first



Harder to decide when both value of what NOT done and duration change

Same lost $, do < duration first

There are many ways that a more economically optimal work order can be derived. These rank from no attempt to find a more optimal order to computing the impact of delay on profitability over the products useful lifespan. The goal is to use the technique that is appropriate for the level of impact if wrong. For major decision with huge financial impact, use the techniques more to the right.

The leftmost techniques focus on quickly determining an items value and prefer to do those first. As we move to the right, the "value" estimate starts to consider more factors. Techniques on the right start to estimate the value based on total market profitability impact due to doing something else first. All techniques are a balance between analysis time and how impactful it might be to be wrong.

Most common
Subjective

Small to medium impact decisions
Gets close when the impact of being wrong is moderate

Medium to large impact decisions
Start understanding the total cost of delay (products and major features)

HiPPO
Just make a choice

Relative Value & Effort (delay) Matrix
Balance duration and value

SAFe "Arnold"
Make urgency dominate

SAFe (real estimates)
Normalize & weight values against $ and time range scales

Economic Impact of delay over a product's entire lifespan is modeled (long term value)

Random choice ⟷ **Crystal Ball**

Highest Value First
Common in Scrum. Some attempt as estimating "value"

SAFe
Consider more than one value type (estimated using Fibonacci point scale)

Cost of Delay Curve Categorization
(e.g. Kanban's class of service)

Quantitative $ per Fixed Period
(e.g. CD3)

Note: a naïve assumption is that good analysis is performed in every technique shown here. Its possible to do all of these techniques poorly and make poor decisions. Use care!

Trade offs

Chance of making a sub-optimal decision

Effort required to get optimal decision

Highest Value First: Common in Scrum

Scrum proposes starting the highest customer value work first. Some teams use a qualitative low, medium and high. Some attempt to estimate it in dollars. This is better than random ordering, but often leads to "Biggest liar wins." It also doesn't consider how long each item will take, meaning more value might be delivered in a number of smaller items that sum to greater value.

SAFe and SAFe "Arnold Mod"

SAFe (Scaled Agile Franework) uses a weighted shortest job first balancing technique. It uses subjective measures and approximates optimal starting order using the following formula (highest first):

$$\frac{\text{Value + Criticality + Risk Reduction or Opportunity Enablement}}{\text{Job Size}}$$

Joshua Arnold offers the following modification to make time criticality more dominant:

$$\frac{\text{Criticality x (Value + Risk Reduction or Opportunity Enablement)}}{\text{Job Size}}$$

Economic Models and WSJF

Donald Reinertsen in his book "Principles of Product Development Flow" offers a variety of scheduling techniques. The most popular is Weighted Shortest Job First where optimal starting order is calculated using delay impact in dollars and size. Optimal order (highest to lowest) is calculated using the formula:

$$\frac{\text{Cost of delay}}{\text{Duration of delay}}$$

Reinertsen suggests it's prudent to consider the total market impact of a delay, not just the immediate lost value.

Jointly designed in an online conversation by Martin Burns, Don Reinertsen, Chris Matts, Joshua Arnold, Tony Grout and Troy Magennis sometime during 2016. It is, and will remain a work in progress. Trademarks used NOT ours.

Better Backlog Prioritization (from random to lifetime cost of delay) - Detail

SAFe Weighted Shortest Job First

The Scaled Agile Framework proposes an ordering system based on Don Reinertsen's Weighted Shortest Job First (WSJF) principles. Proposed features are assessed on multiple value and size axis using relative Fibonacci story point estimates. The process is described as -

1. Rate each parameter against the other features using the scale: 1,2,3,5,8,13,20. Do one column at a time, and calibrate the lowest value to be a "1" - each column MUST have one "1"
2. Calculate the WSJF value for each column using the formula shown below
3. Do the feature that has the HIGHEST WSJF value first if possible

Pros: helps prioritize more than one type of value, and balances time based on the proxy job size
Cons: story point estimates don't handle extreme variation in value, job size not always duration

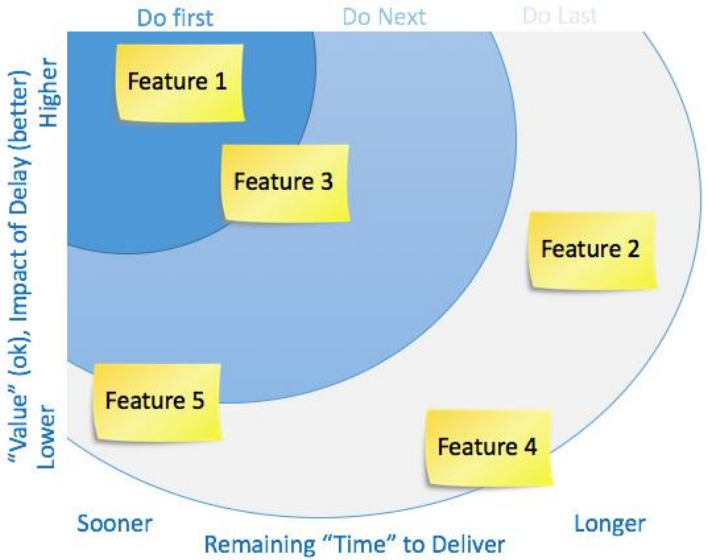
SAFe's Weighted Shortest Job First formula (upper), and a typical data capture table (lower)
More info: http://www.scaledagileframework.com/wsjf/

$$WSJF = \frac{\text{User-Business Value + Time Criticality + Risk Reduction | Opportunity Enablement Value}}{\text{Job Size}}$$

Feature	User-Business Value	Time Criticality	RR | OE Value	Job Size	WSJF

Matrix Techniques – quick filtering

Have the teams place a feature on a matrix of delay time and impact of delay (start with the value you use today, strive for more complete cost of delay). Do higher impact, shortest delivery time items first. Erik Willeke uses a variation where ONLY product owners can move items up or down (indicating higher or lower impact) and the development team left and right (shorter or longer to deliver).



SAFe Weighted Shortest Job First Variations (un-sanctioned)

Some variations of the basic SAFe formula and technique have evolved to make the computation more likely to match ideal.

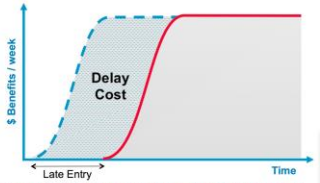
1. "Arnold Mod"
In an email thread conversation between Martin Burns and Joshua Arnold, the suggestion of making Time Criticality more dominant was suggested. This solves the theoretical problem that something "Critical" might score a lower WSJF due to a high business value or risk reduction or opportunity enablement or a low size. Martin noted that these rarely occur due to earlier decision processes, but this suggestion would solve these even if they slipped through.

$$WSJF = \frac{\text{Time Criticality x (Value + Risk Reduction or Opportunity Enablement)}}{\text{Job Size}}$$

2. Scale and Weight the Arguments (solve the mathematical issues of different argument units)
The use of Fibonacci numbers for the input arguments is an attempt to make the estimates relative to each other for the same input argument, but there is a chance that the magnitude is different for each value. For example, a "5" in value might be $100,000, but a "5" in risk reduction might be $500,000. If we just added them as the original SAFe formula says, the result makes little intuitive sense. To correct, either scale the values to normalize across arguments, or multiple each Fibonacci value by a weighting multiplier to correct the magnitude mismatches.

How value is lost due to a delay – Urgency Profiles

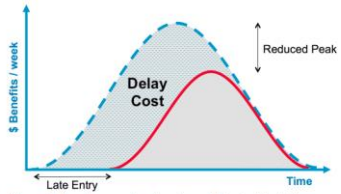
Value erodes differently for different products and markets. The most commonly calculated is just the loss of revenue on the front end because of being late. But, the lost value can be much more than that if the delay causes a permanent erosion of market share, or if the market window is short. It can be difficult to calculate the longer term value erosion, but it may be significant.



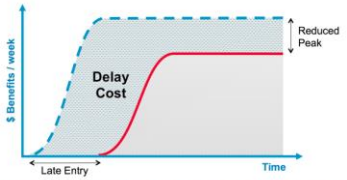
For ideas with a very long-life, with peak unaffected by delay

Tips:
1. Estimate the lifespan of the feature or product. If it has a shorter market window, do it sooner.
2. Estimate if a delay will permanently impair market share. If it does, do it sooner.

Images credit: blackswanfarming.com/urgency-profiles/



Short benefits horizon, and reduced peak due to late delivery



For ideas with a very long-life, with reduced peak due to later delivery

DOs
- Encourage better economic decisions
- Use the lightest analysis method to get a decision
- Use these methods to help have conversation about what value means to each feature or product
- Find better ways to measure and estimate
- involve a diversity of viewpoints on both value and delay.
- Consider reducing risk in a project earlier as adding value

DONTs
- Use complex analysis on small items. Ideally only for features and larger
- Ignore delivery time or its proxy job size; this leads to sub-optimal ordering
- Create an arms race for "value" by prioritizing on it alone (biggest liar wins syndrome)
- Use the highest paid persons opinion if at all possible, offer alternatives!

REFERENCES

Donald Reinertsen:
Books: Principles of Product Development Flow has great Cost of Delay ideas and concepts.
Video: Cost of Delay: Theory & Practice with Donald Reinertsen https://www.youtube.com/watch?v=OmU5yIu7vRw

SAFe:
http://scaledagileframework.com/wsjf/

Joshua Arnold's blog:
http://blackswanfarming.com/category/cost-of-delay/

Chris Matts Blog:
https://theitriskmanager.wordpress.com

Troy Magennis:
Spreadsheets for Cost of Delay http://Bit.Ly/SimResources
Blog: http://focusedobjective.com/blog/