Prioritization

10 different techniques for optimizing what to start next

@t_magennis | troy.magennis@focusedobjective.com

All slides and spreadsheets: Bit.ly/SimResources

“The problem with any prioritization decision is [it is a] decision to service one job and delay another.”

Don Reinertsen

Easy Fix

DO EVERYTHING NOW
by employing unlimited trained and talented teams that delivery instantly

Have a doctor of every specialist discipline for every bed in the ER,
and one bed for every citizen in the local area

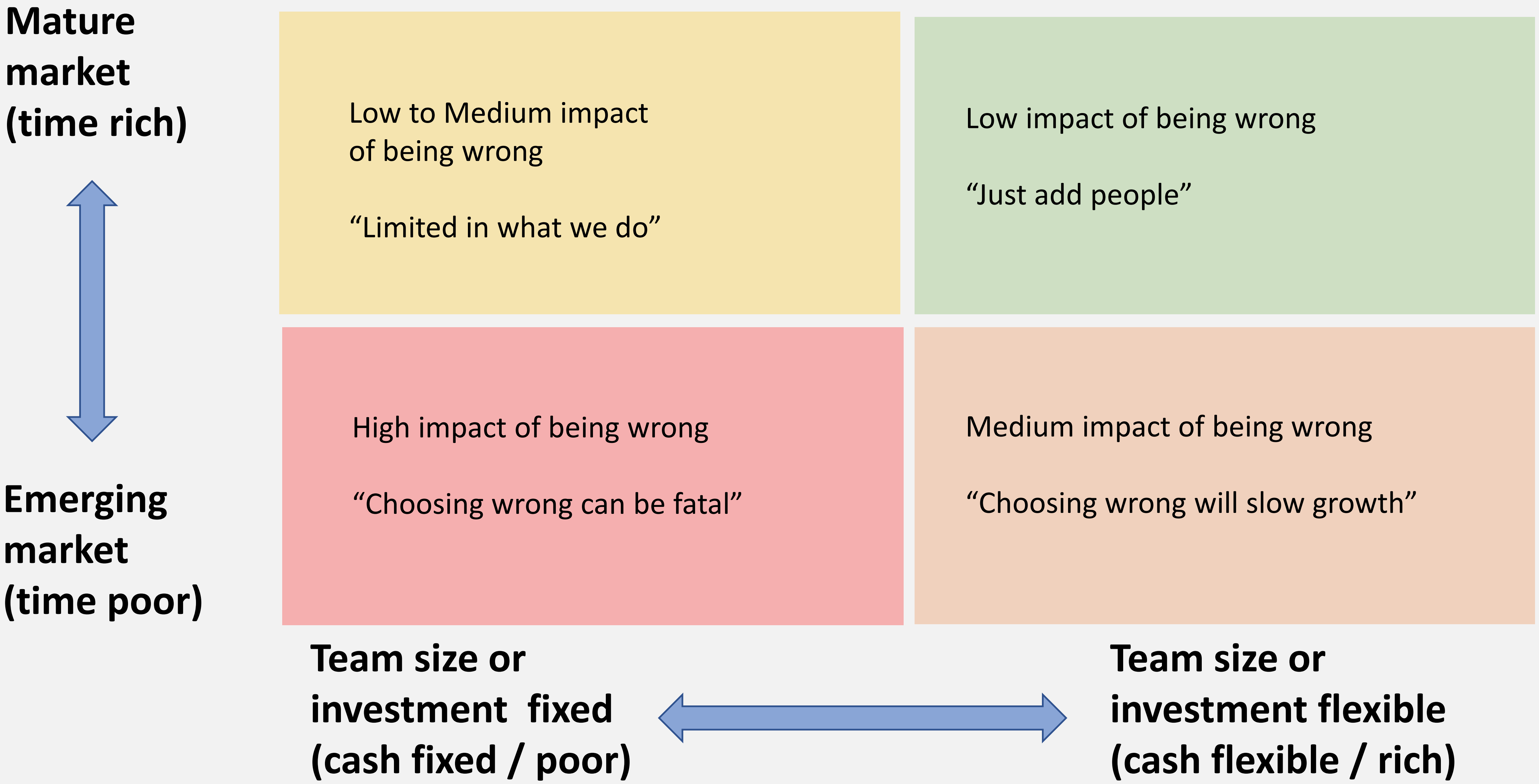
Demand > Available Capacity

The Goal – What to start next

Given a set of things we could do next,
<u>is one more economically advantageous</u>
to start next given **we can't** do them ALL

The Challenge – Doing one thing delays others

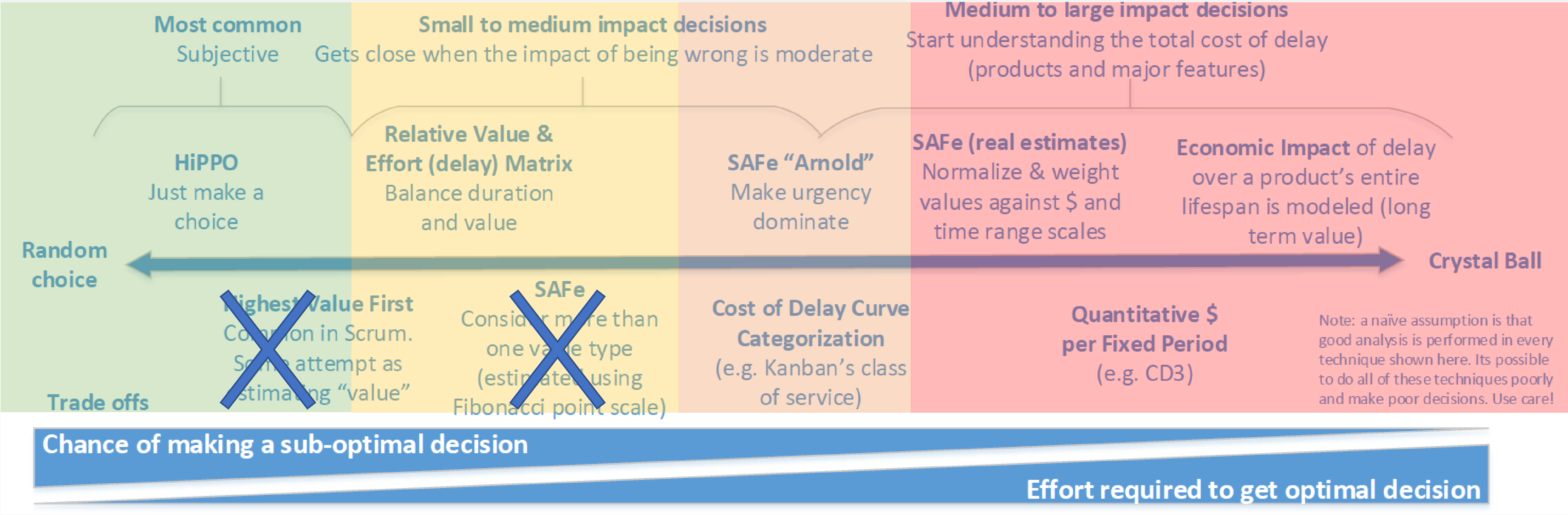
Every item has a different economic impact by being delayed.
The impact will be a product of lost value,
and how long they are delayed.



	Team size or investment fixed (cash fixed / poor)	Team size or investment flexible (cash flexible / rich)
Mature market (time rich)	Low to Medium impact of being wrong “Limited in what we do”	Low impact of being wrong “Just add people”
Emerging market (time poor)	High impact of being wrong “Choosing wrong can be fatal”	Medium impact of being wrong “Choosing wrong will slow growth”

@t_magennis

Choose based on impact of being 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

X Don't use. Better option at similar effort

@t_magennis

What starts on Twitter…

- Martin Burns, Joshua Arnold, Tony Grout, Chris Matts, Don Reinertsen
- Initial argument was about the SAFE cost of delay formula

- What I wanted to learn
  - Document the methods used for prioritization
  - Give advice to clients on why use one method versus another
  - Uncover why flawed methods still occasionally work (context)

- My goal: Single page cheat-sheet

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.

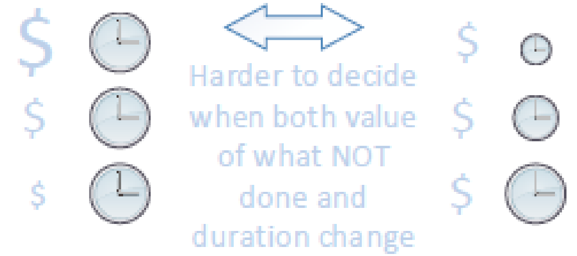
The Challenge – Doing one thing delays others

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

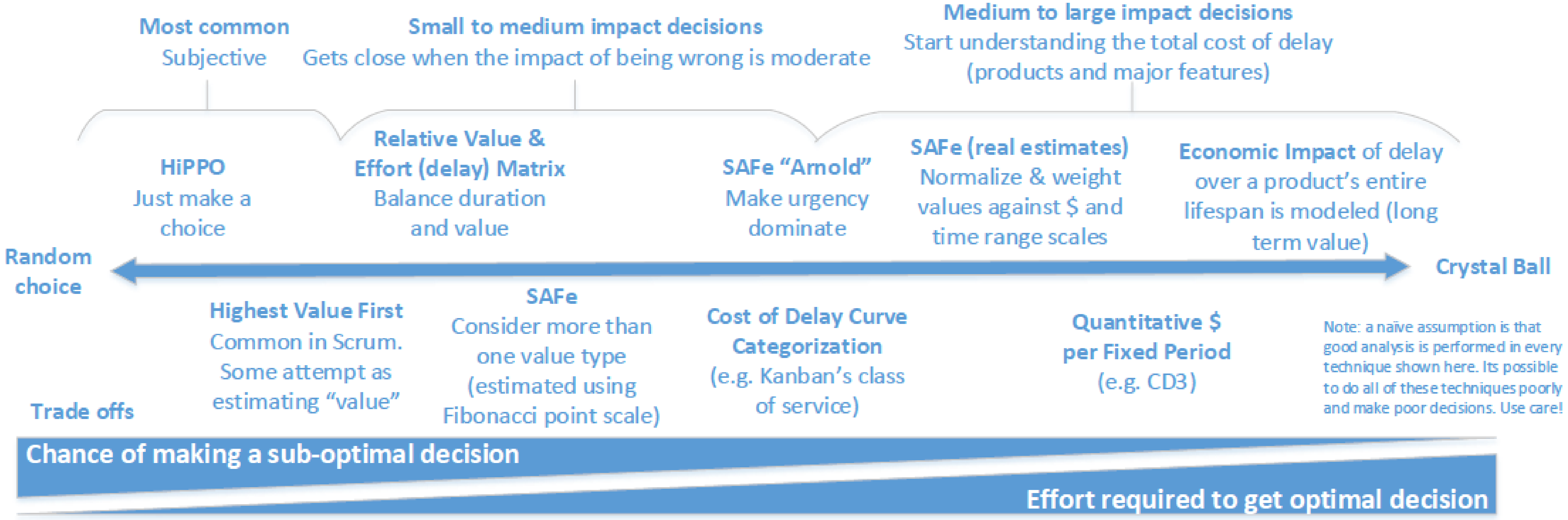
$ = Value lost due to delay 🕒 delay duration


Same duration,
do > lost $ first
Same lost $,
do < duration first
$
$
$
$
$
$
Harder to decide when both value of what NOT done and duration change

There a… d a more opt imal order t… ue t hat is approp… the right.

The left… ght, the "value" estimat… ofitability impact due to doing something else first. All techniques are a balance between analysis time and how impactful it might be to be wrong.

http://bit.ly/BetterPrioritization



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)

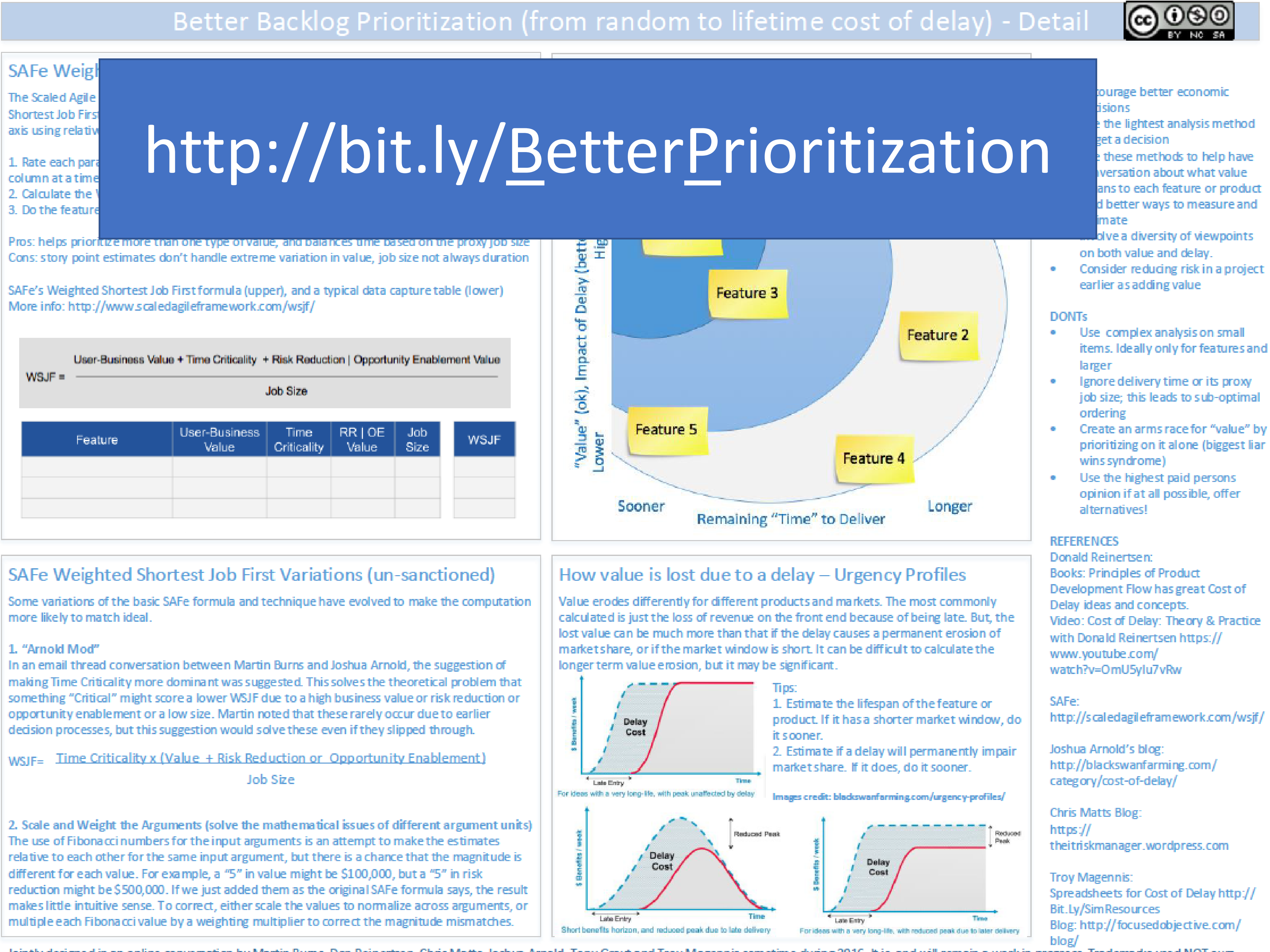
Quantitative $ per Fixed Period
(e.g. CD3)

Note: a naive 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



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

SAFe Weig

The Scaled Agile
Shortest Job Firs
axis using relativ

1. Rate each para
column at a time
2. Calculate the
3. Do the feature

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/

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

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

Hig
"Value" (ok), Impact of Delay (bett
Lower
Feature 3
Feature 2
Feature 5
Feature 4
Sooner
Remaining "Time" to Deliver
Longer

courage better economic
cisions
e the lightest analysis method
get a decision
e these methods to help have
nversation about what value
ans to each feature or product
d better ways to measure and
imate
olve 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!

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.

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

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

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=OmU5ylu7vRw

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/

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 our

http://bit.ly/BetterPrioritization

Lessons…

- There is a continuum and trade off on two axis
  - Effort to analyze (assumed related to time)
  - Ability to uncover a more optimal answer (assumed it can be known)
- Considerations in choice
  - How fixed is investment
  - Uncertainty of development effort (how known and understood)
    - Massive technical or development risk – choose more to the left, live with sub-optimal
    - Minimal technical or development risk – choose more to the right, use impact to choose
  - Impact of being sub-optimal in "value" delivery
    - Inconvenience – choose more to the left
    - Extinction – choose more to the right

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!

Random choice

Trade o

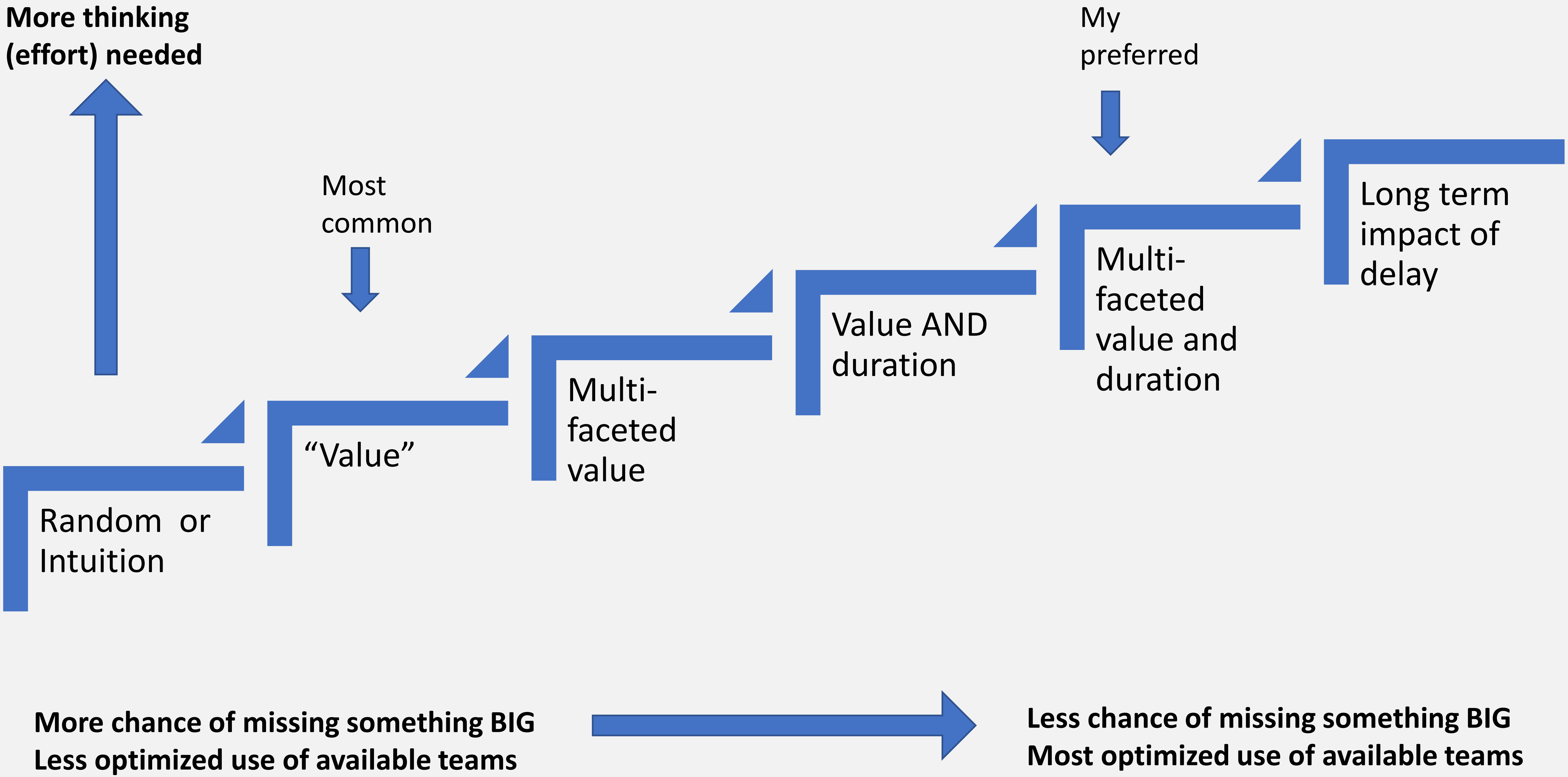
Chanc

lay
e
ng

Crystal Ball

assumption is that
is performed in every
wn here. Its possible
ese techniques poorly
r decisions. Use care!

ecision



More thinking (effort) needed

Random or Intuition

“Value”

Most common

Multi-faceted value

Value AND duration

Multi-faceted value and duration

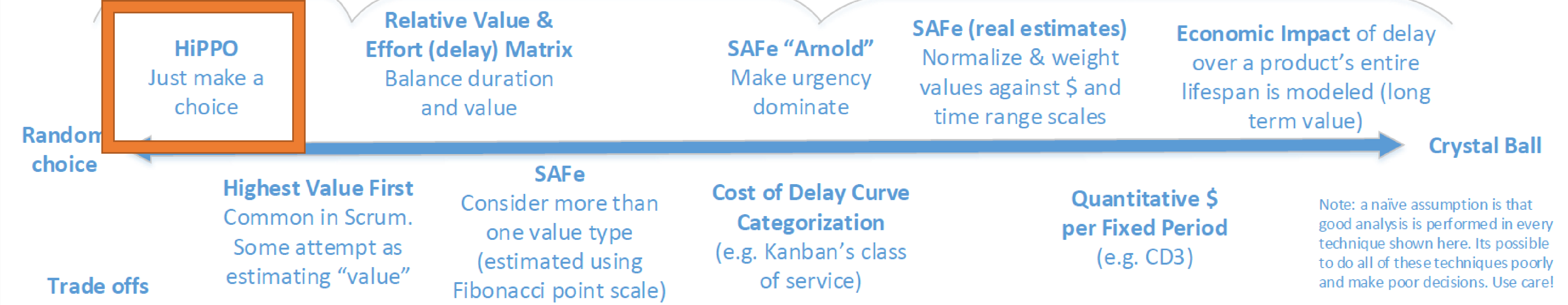
My preferred

Long term impact of delay

More chance of missing something BIG
Less optimized use of available teams

Less chance of missing something BIG
Most optimized use of available teams

@t_magennis



Random choice

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)

Crystal Ball

Trade offs

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!

"Biggest Paycheck Wins"

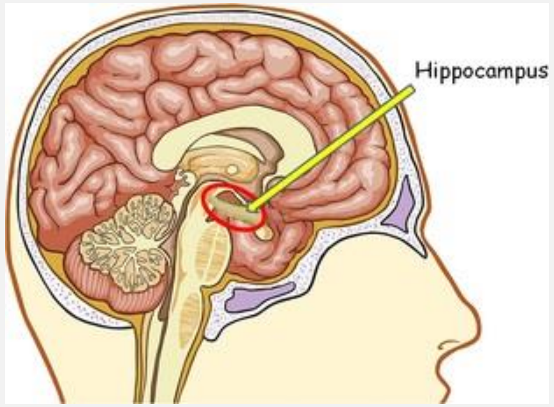
Works:

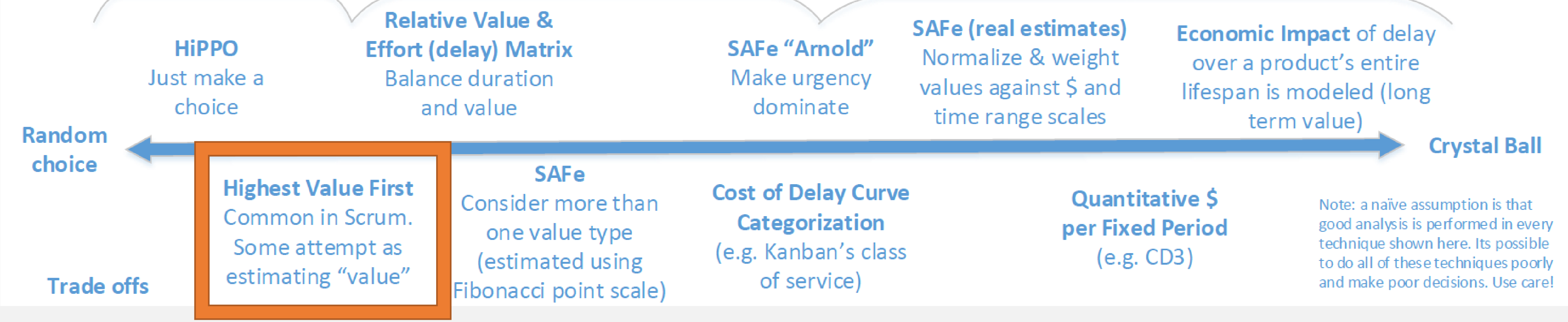
- Lots of compelled options
- Effort is similar or fixed
- Reasonable HiPPO who listens and learns

Fails:

- When HiPPO isn't accessible
- When the HiPPO has alternative motives

NOT EQUAL TO BIGGEST HIPPOCAMPUS





Random choice

Trade offs

HiPPO
Just make a choice

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

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

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

SAFe "Arnold"
Make urgency dominate

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

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

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

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

Crystal Ball

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!

"Biggest Liar Wins"

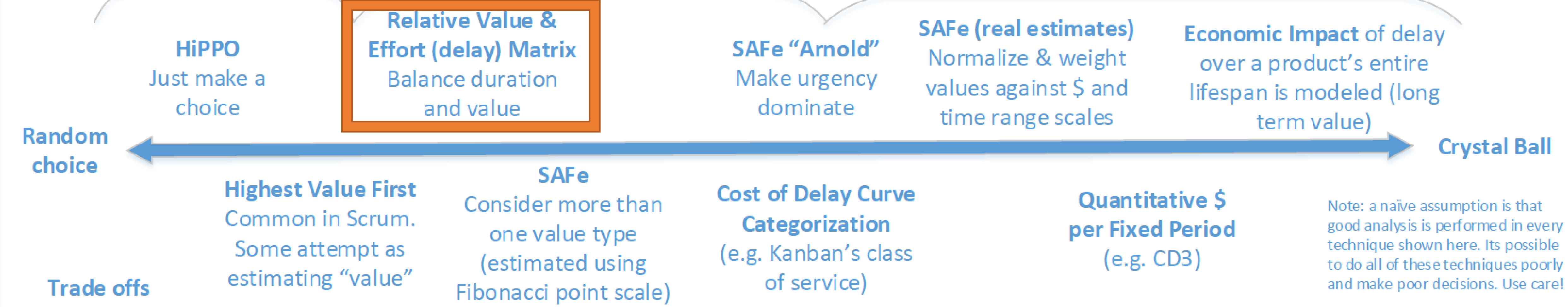
Works:
- Value is computable
- Effort is similar

Fails:
- Value is abstract
- Effort is different

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.

@t_magennis



Random choice

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)

Crystal Ball

Trade offs

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!

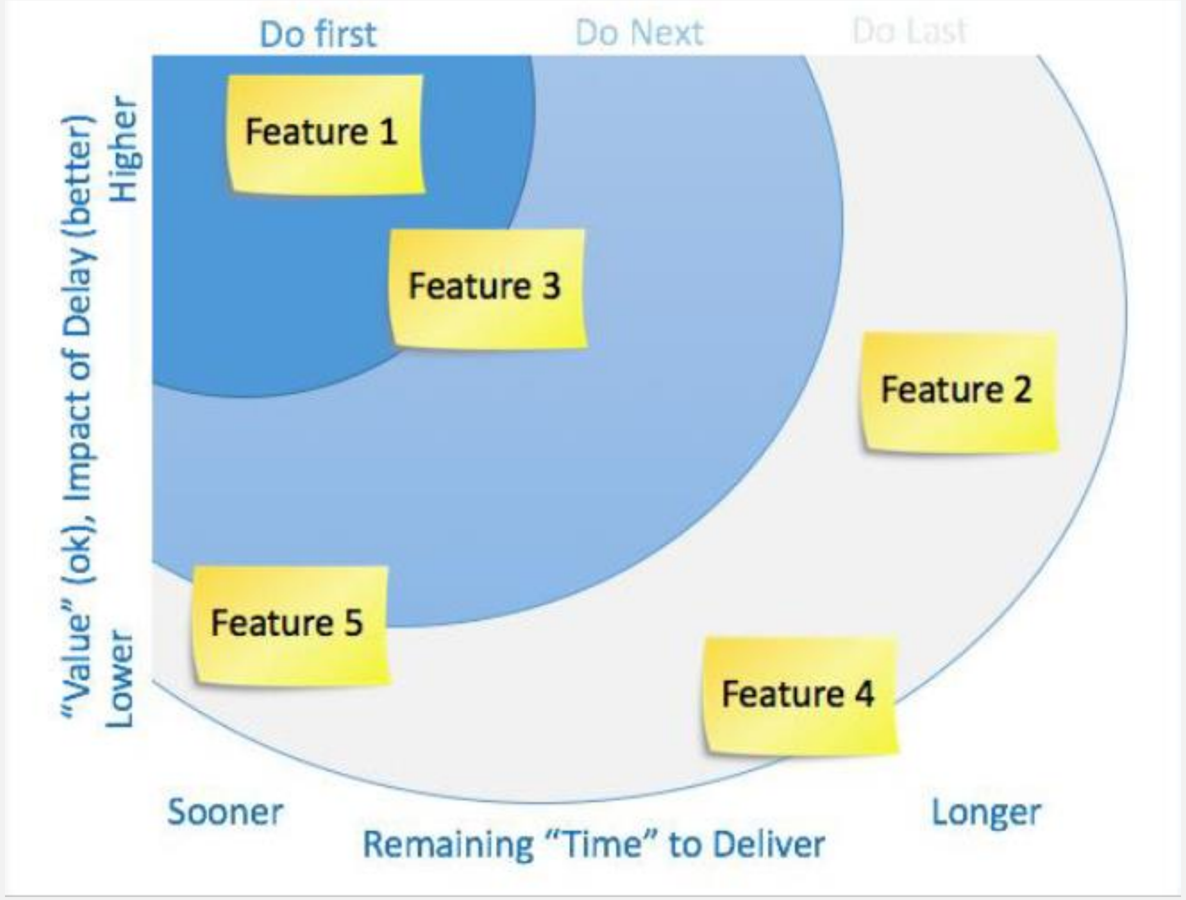
“Now with Added Effort”

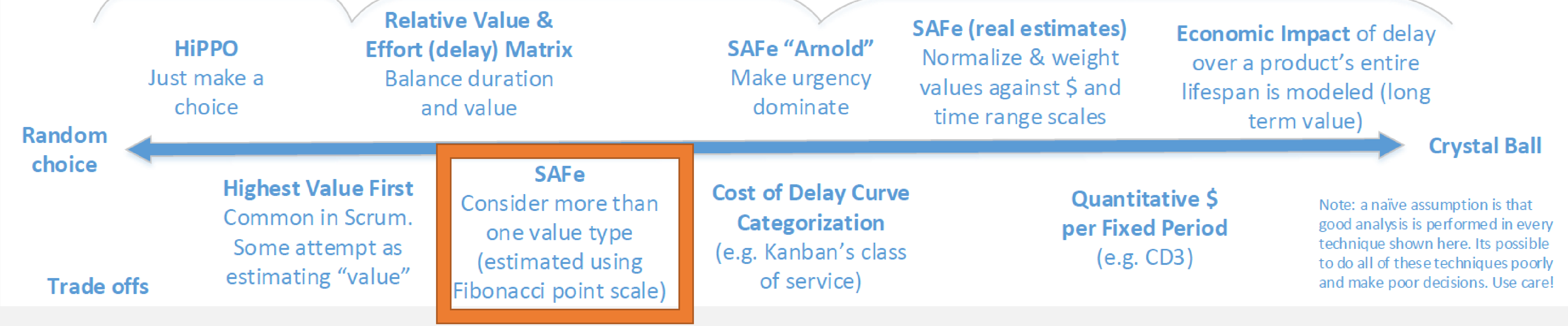
Works:

- Effort is dis-similar

Fails:

- Value and effort scaled poorly
  How much value = how much effort





Random choice

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)

Crystal Ball

Trade offs

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!

"Un - SAFe"

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

Works:

- More than one type of value
- Effort (job size) dis-similar

Fails:

- Mixes dis-similar units
- Mixes dis-similar scales

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

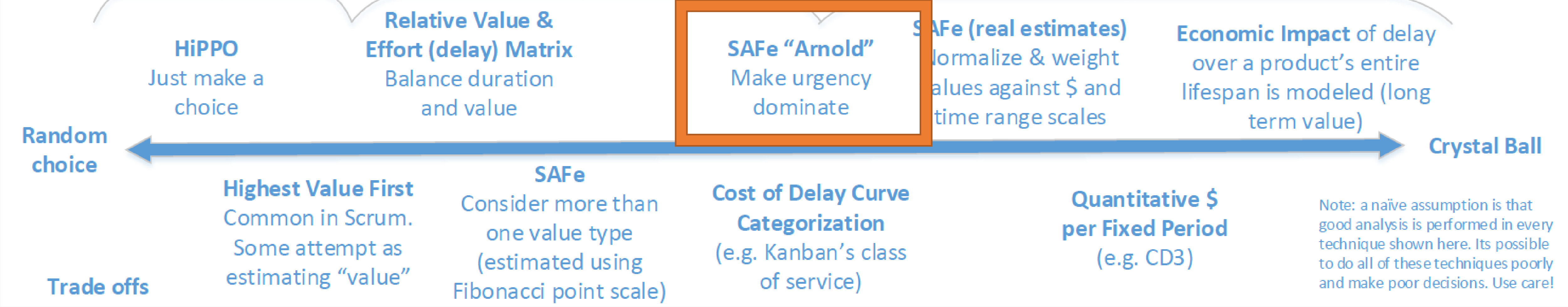
Pros: helps prioritize more than one type of value, and balances time based on the proxy job size
Cons: story point estimates don't handle extreme variation in value, job size not always duration

SAFe Cost of Delay WSJF risks

Martin Burns correctly notes that these issues are handled before items get to this formula, so its important who trains you to use this SAFe formula.
Hope it is Martin.

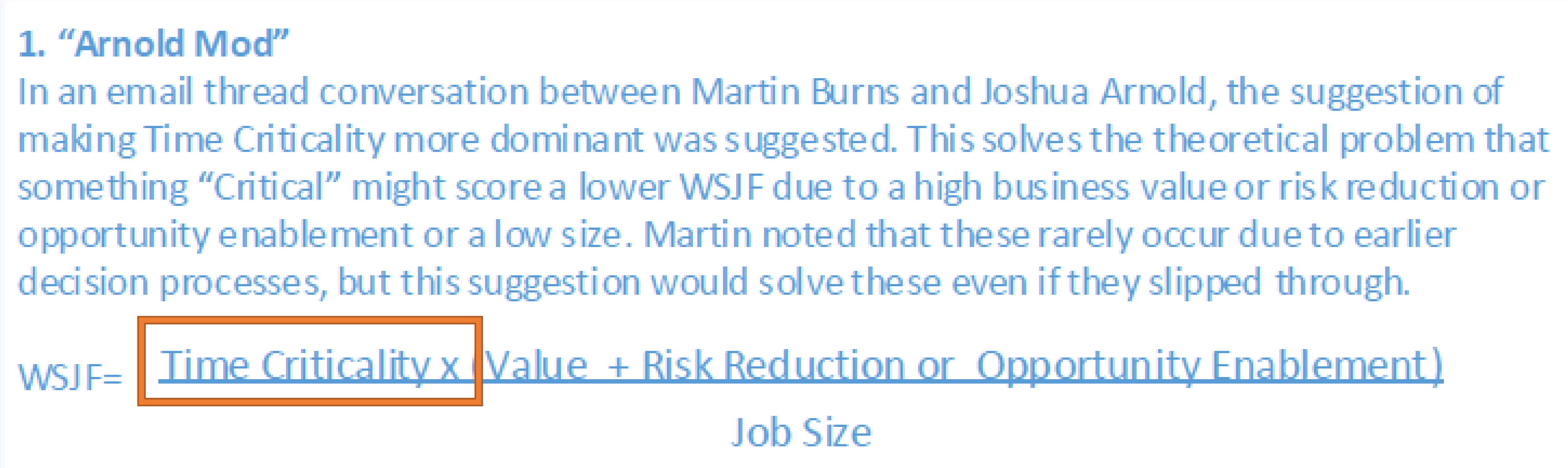
I suspect some trainers don't!

- Mixes units
  - Time criticality vs User Business Value?
- Un-scaled inputs
  - A "5" in time criticality = "?" in user business value?
  - What if one input is in the millions and one is in the thousands range
- Risk Reduction OR Opportunity Enablement
  - Boolean operator in a mathematical formula?
  - It would seem an arbitrary crutch because the formula was getting too big…


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)
Trade offs

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!

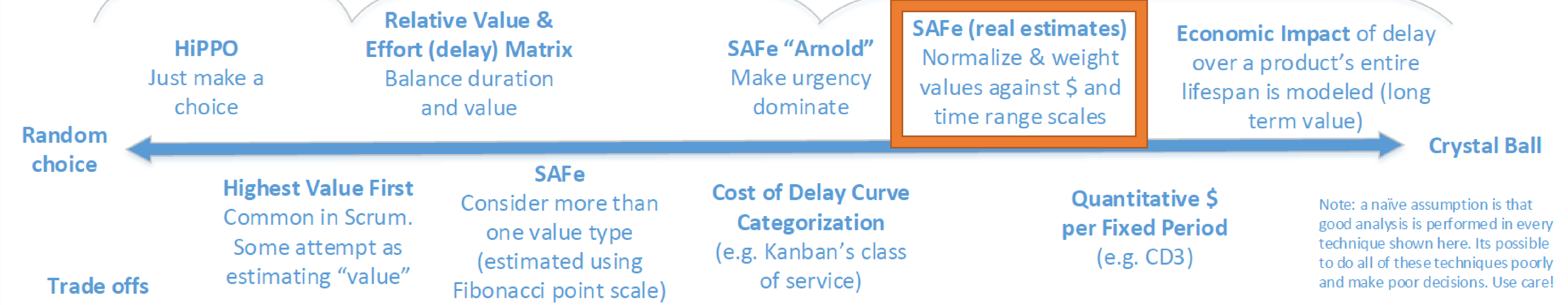
"SAFeR"



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.

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



Random choice

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)

Crystal Ball

Trade offs

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

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

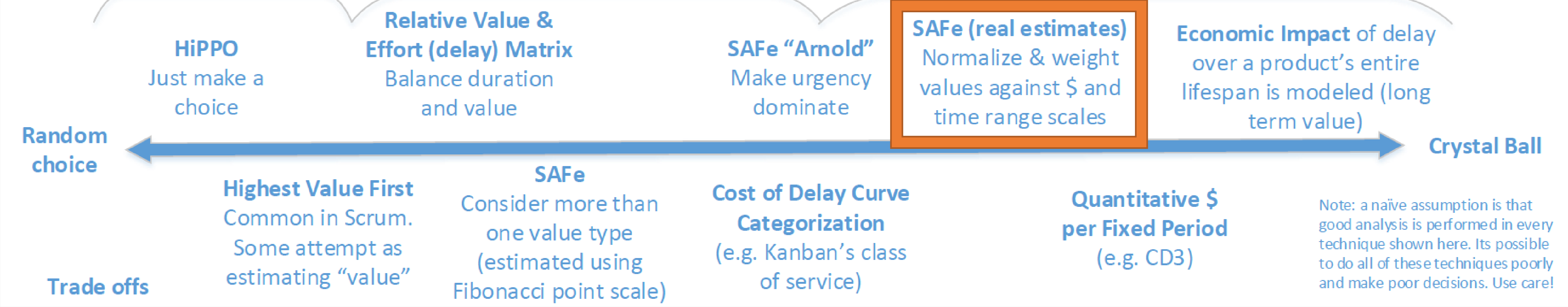
Cost of Delay Curve Categorization
(e.g. Kanban's class of service)

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!

"SAFeST"

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.



Random choice

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)

Crystal Ball

Trade offs

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!

"SAFeST"

1 2 3 5 8

User business value $1000 … $100,000

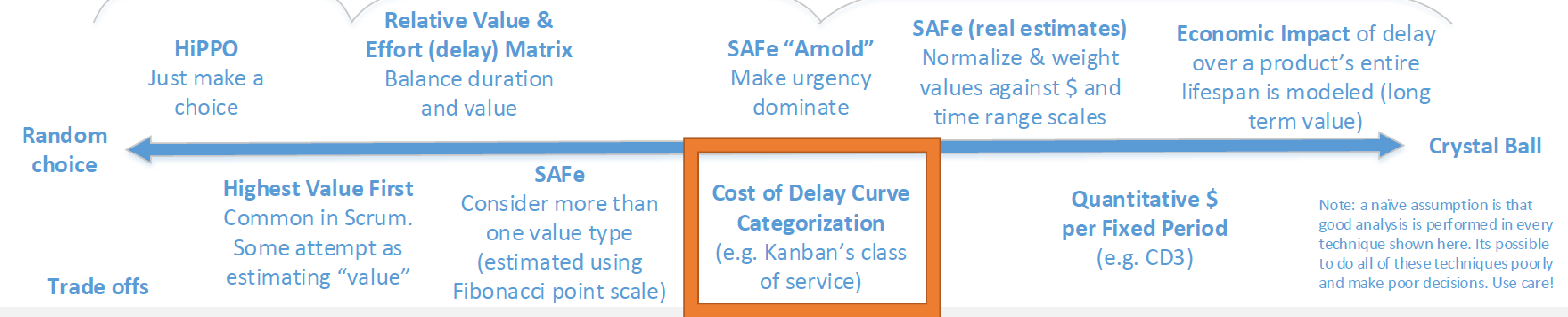
Risk reduction / Opportunity Enablement $1000 … $500,000

5x difference!
This might matter!

Time criticality

Hmmmmm…..
"How much $ value would make us choose a "2" versus a "1"
"How much $ value would make us choose a "3" versus a "2"
Etc.

@t_magennis



Random choice

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)

Crystal Ball

Trade offs

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!

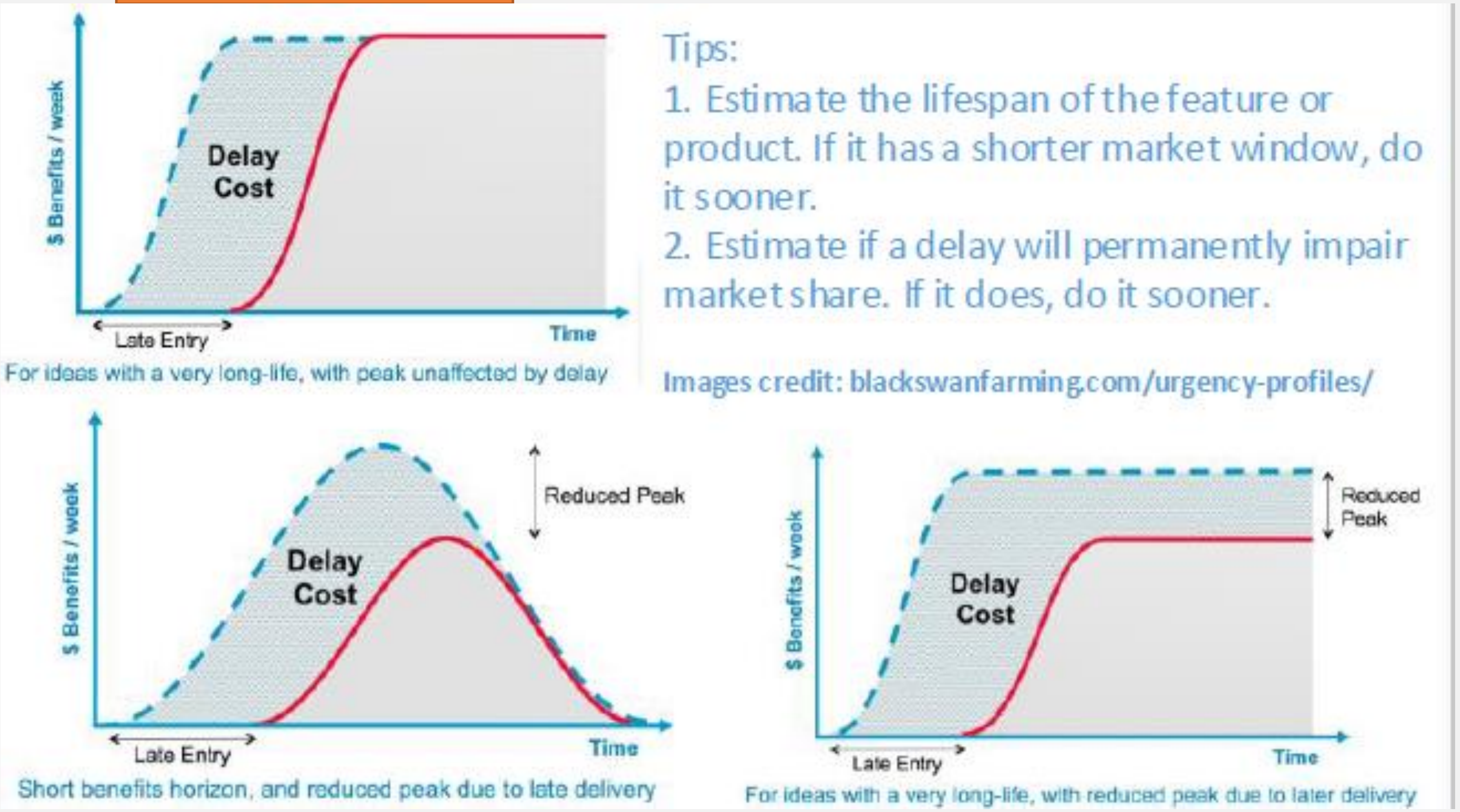
Urgency Profiles

Works:

- Discusses when the impact felt
- Shows long-term impact of delays, starts the conversation

Fails:

- Doesn't quantify values, still qualitative at its roots



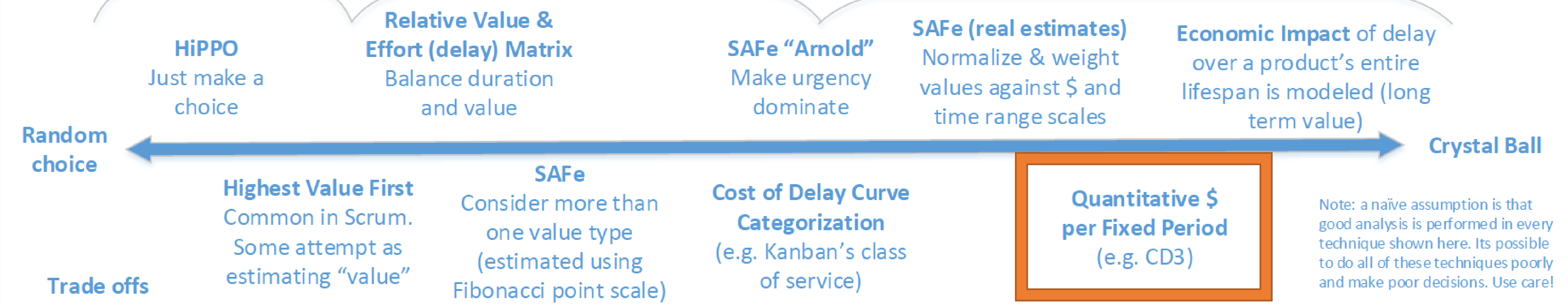
For ideas with a very long-life, with peak unaffected by delay

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



Random choice

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)

Crystal Ball

Trade offs

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!

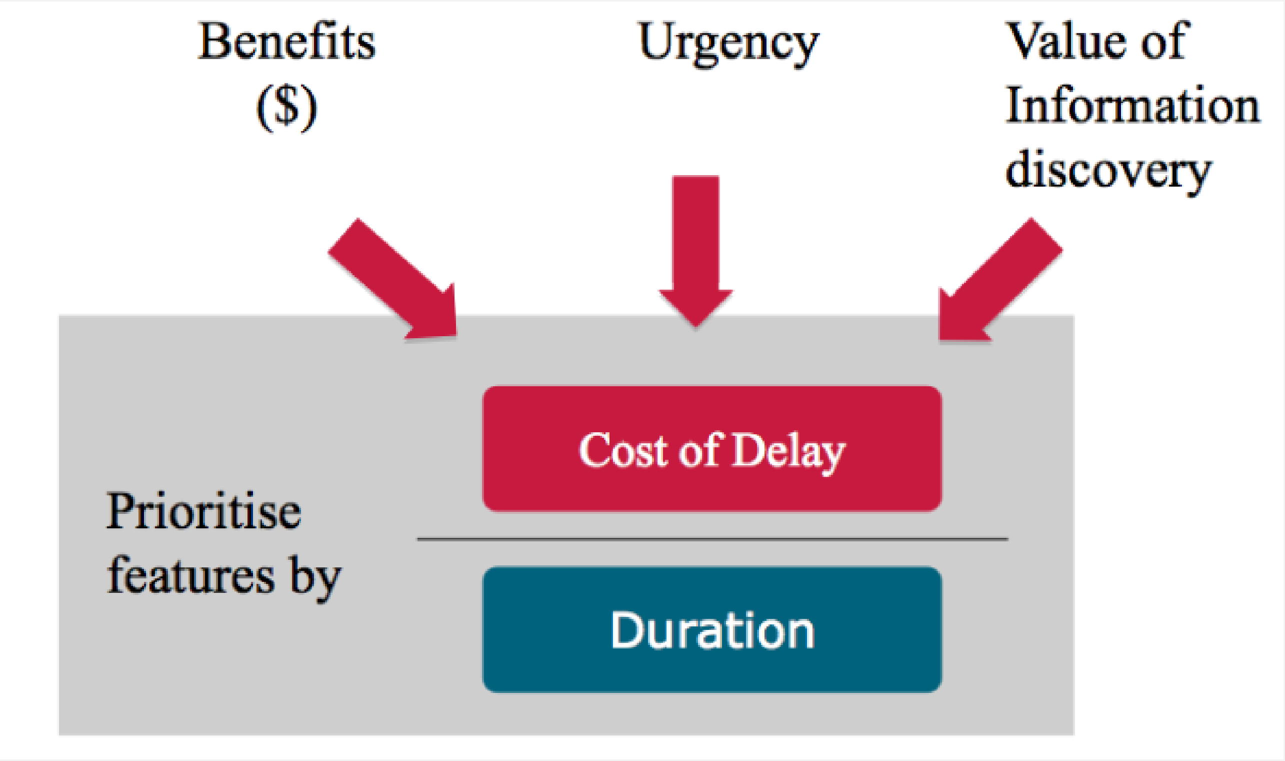
"CD3" – Price tag of Time

Works:

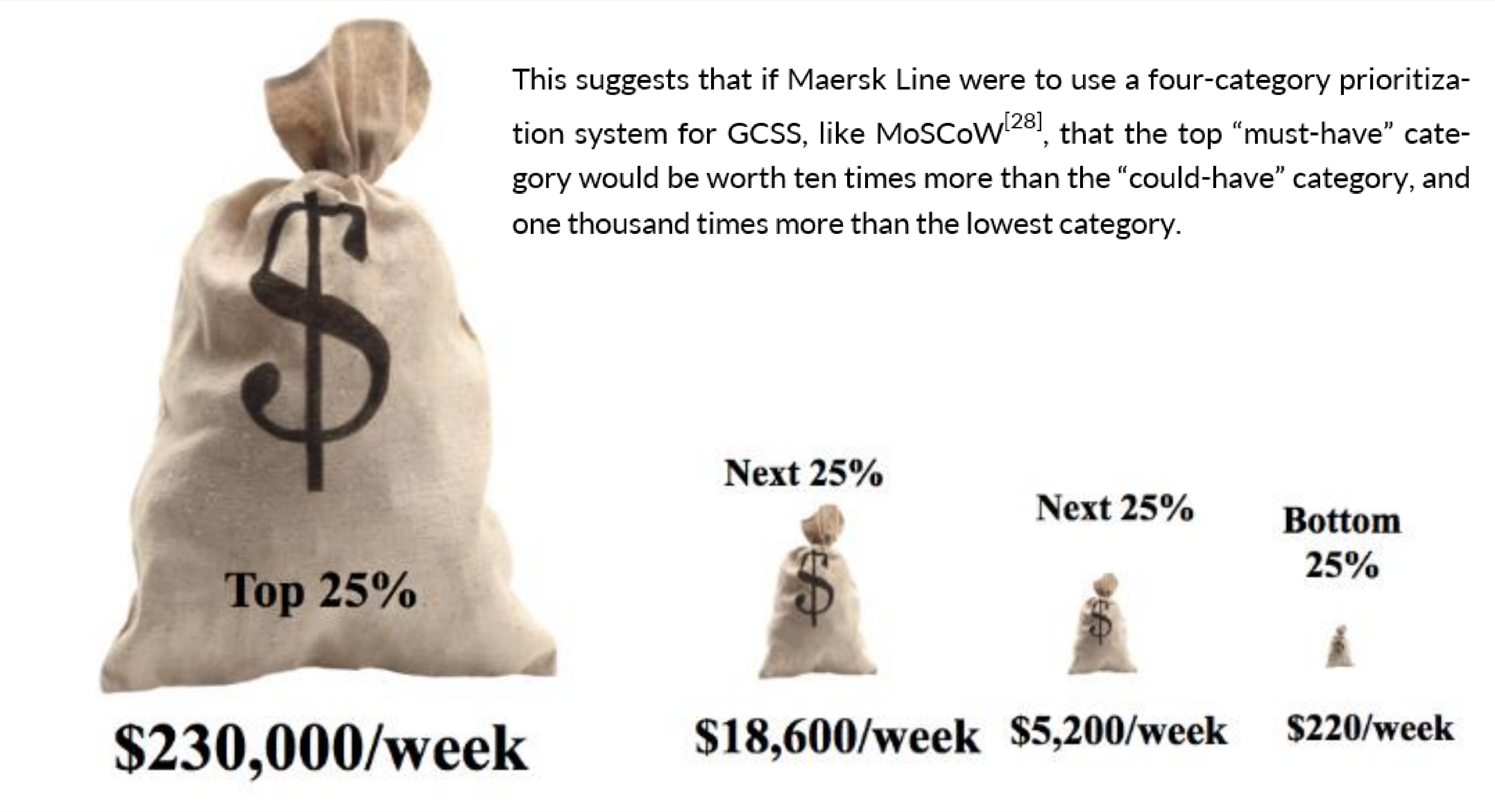
- Handles value AND URGENCY
- Identifies the top priorities fast

Fails:

- Value model requires "thinking"
  - Increase / protect revenue
  - Reduce / Avoid cost

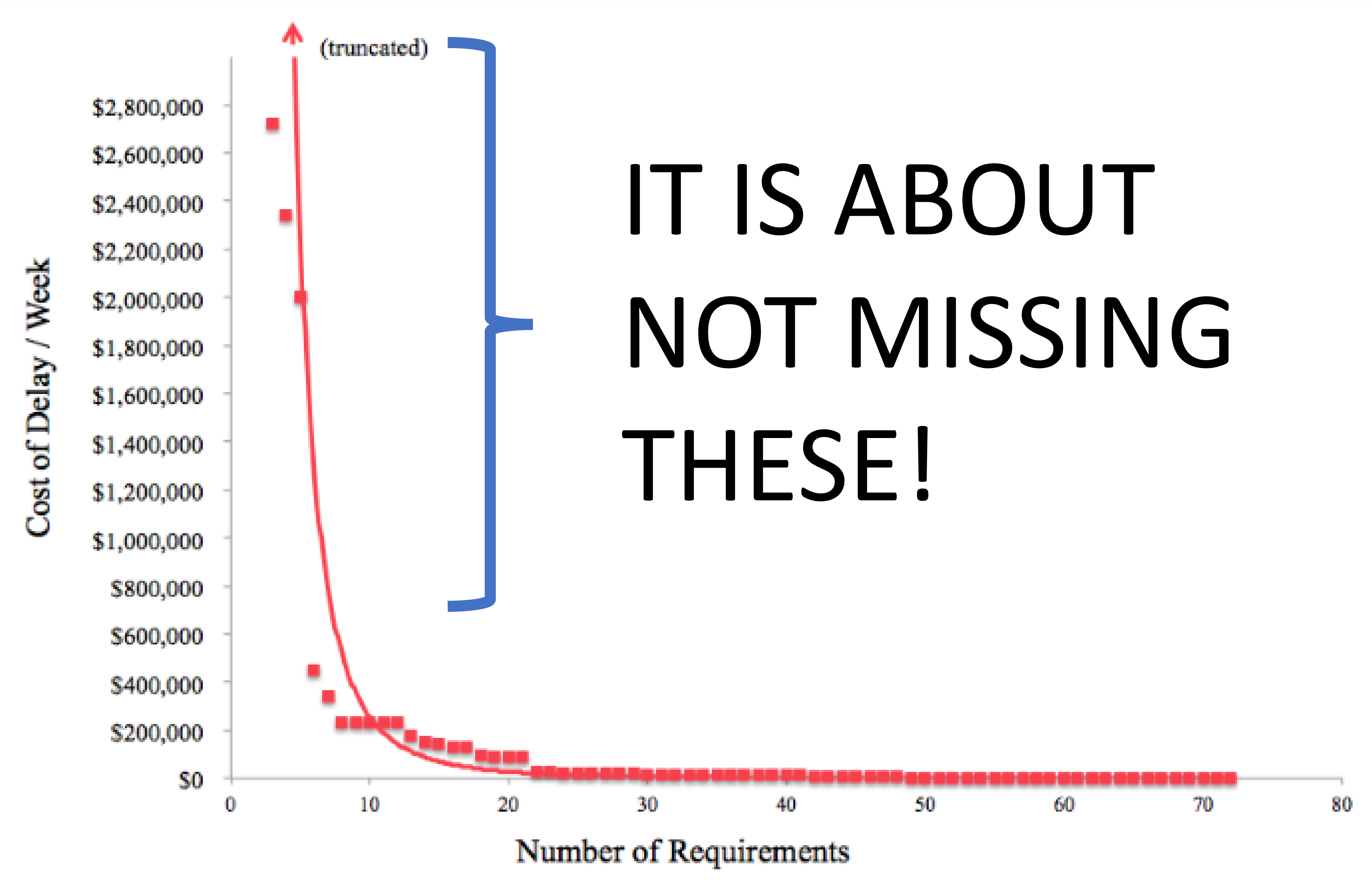


Full paper at http://blackswanfarming.com



This suggests that if Maersk Line were to use a four-category prioritization system for GCSS, like MoSCoW[28], that the top "must-have" category would be worth ten times more than the "could-have" category, and one thousand times more than the lowest category.

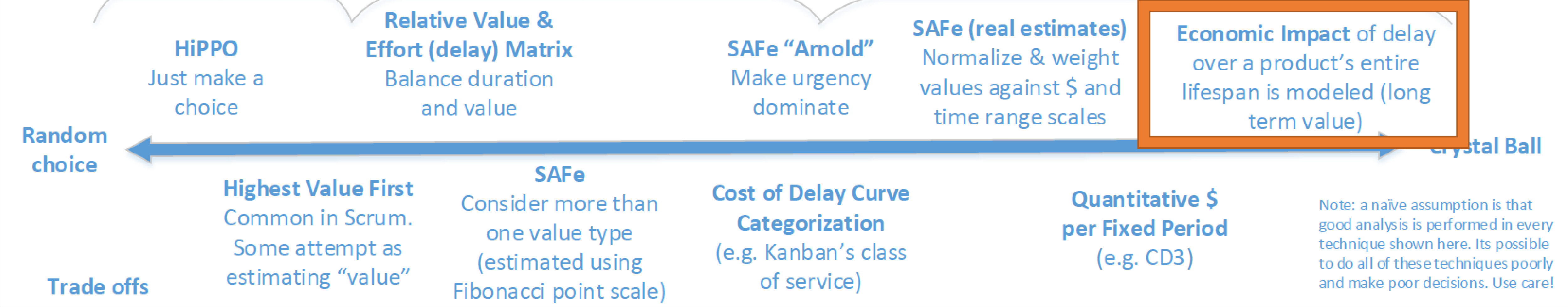
Full paper at http://blackswanfarming.com



IT IS ABOUT NOT MISSING THESE!

@t_magennis

Full paper at http://blackswanfarming.com



Random choice

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)

Crystal Ball

Trade offs

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!

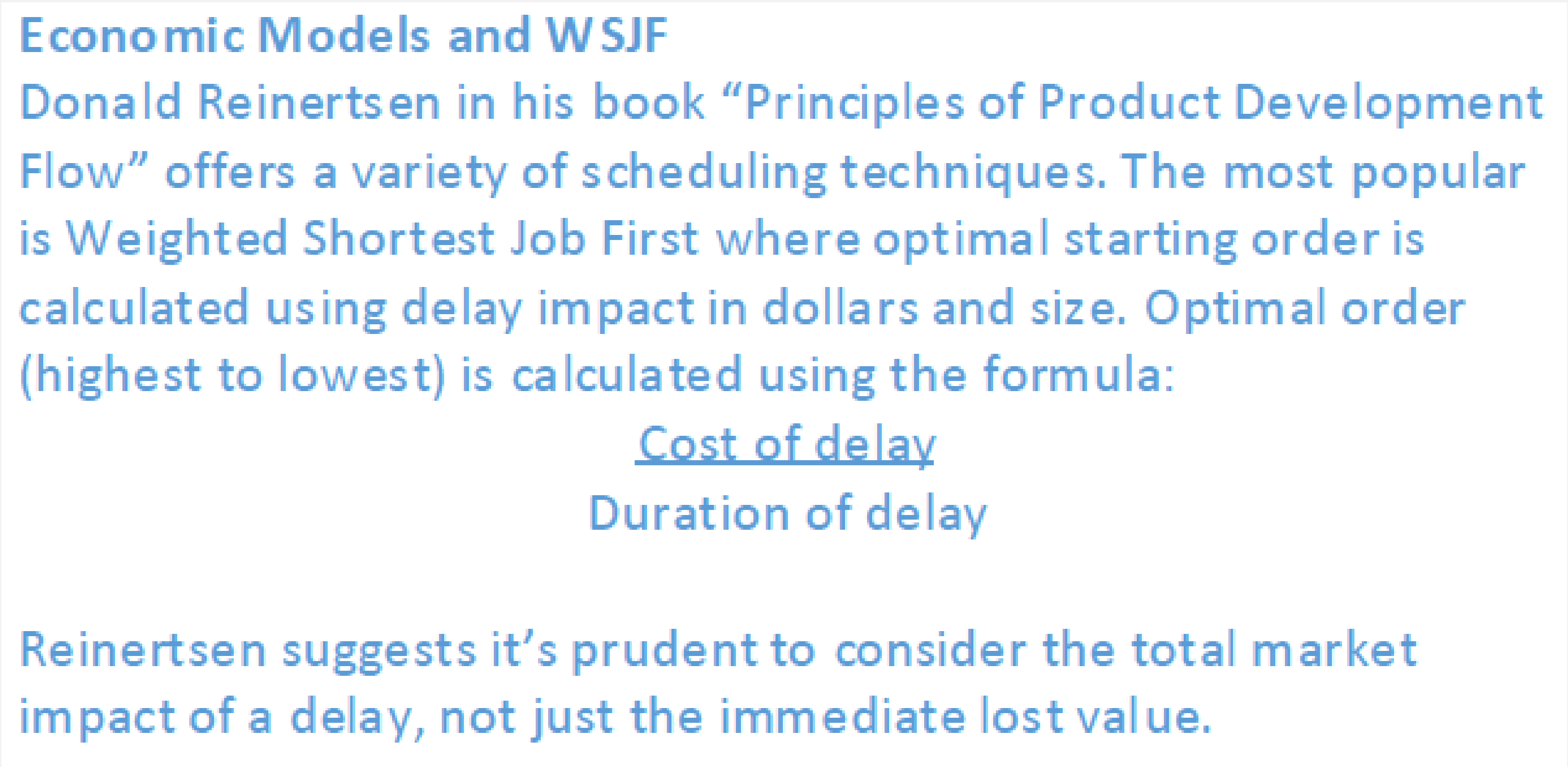
"Cost of Delay"

Works:

- Misses nothing
- A versus B choices

Fails:

- Never getting adopted because it's too hard!



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.

Common Problem 1: Dependencies

- Definition: Something can’t be started until ….
- Options that are enabled by something else
    - The enabler should carry the delay cost of those that it enables
- Dependencies restrict start order options

Common Problem 2: Value Certainty

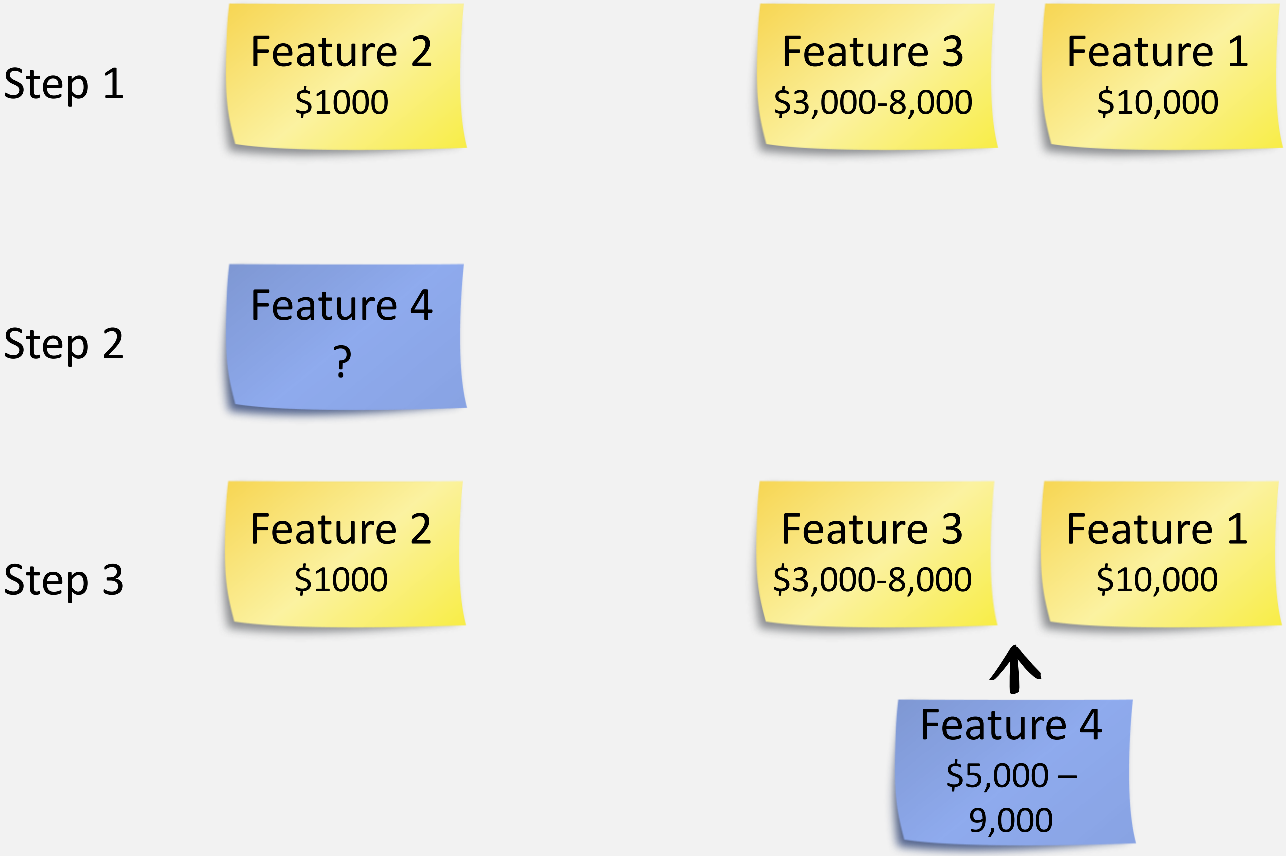
- Definition: Some valuable things are very uncertain….
- Balanced portfolio of risk tolerance
  - 20% - exploratory: 50% of these can make a loss but you survive
  - 50% - good bets, customer needs: moderate win/loss certainty
  - 30% - certainties: enough of these to pay most of the bills

- Failure to choose enough high risk work increases risk of extinction
- Aligned to longer term strategy vs short term needs

Common problem 3: Value is hard to define

- You need an agreed “Value” definition; it changes
- Quick alternatives –
  - Compare to average:
    - Last year revenue gain / number of features delivered last year
  - Reference class forecasting

Reference Class Value Forecasting



Step 1

Feature 2
$1000

Feature 3
$3,000-8,000

Feature 1
$10,000

Step 2

Feature 4
?

Step 3

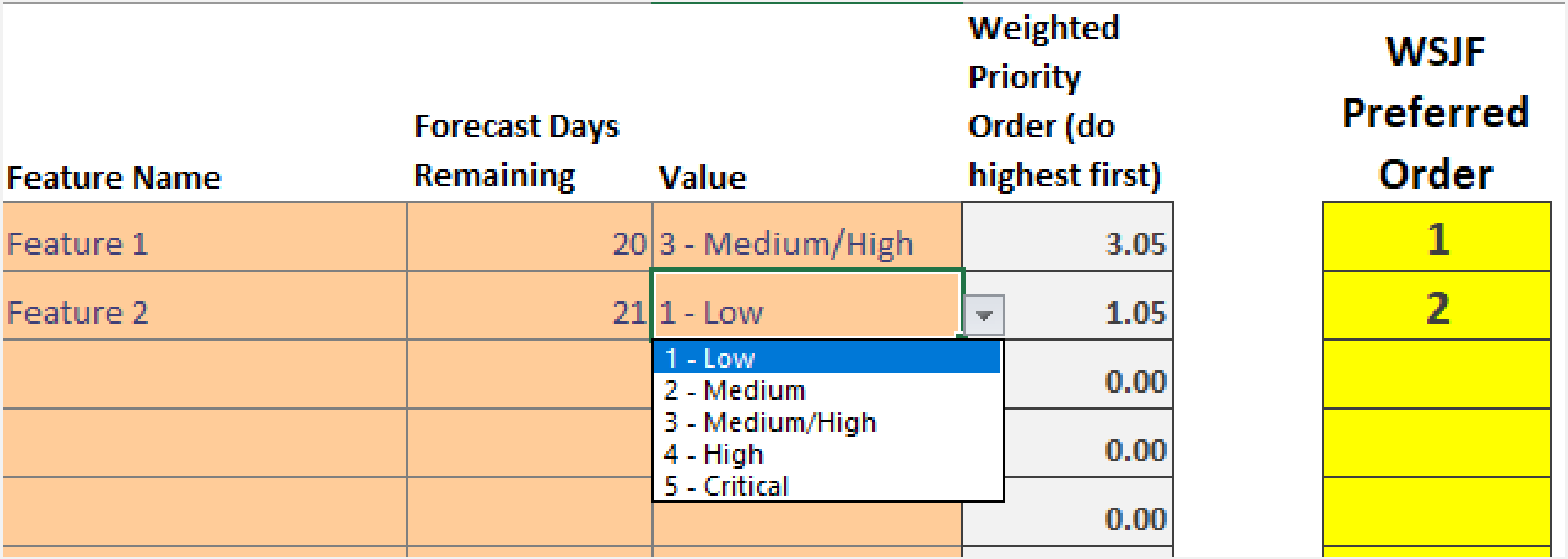
Feature 2
$1000

Feature 3
$3,000-8,000

Feature 1
$10,000

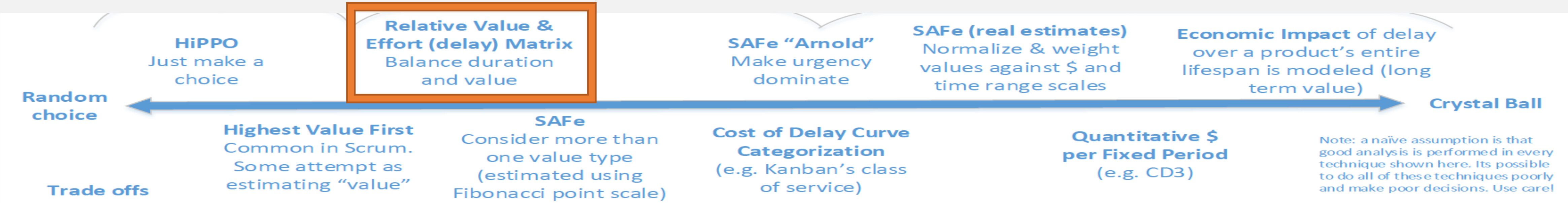
Feature 4
$5,000 –
9,000

Demo – COD spreadsheet (light)



Feature Name	Forecast Days Remaining	Value	Weighted Priority Order (do highest first)	WSJF Preferred Order
Feature 1	20	3 - Medium/High	3.05	1
Feature 2	21	1 - Low	1.05	2
			0.00	
			0.00	
			0.00	

1 - Low
2 - Medium
3 - Medium/High
4 - High
5 - Critical



Random choice

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)

Crystal Ball

Trade offs

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

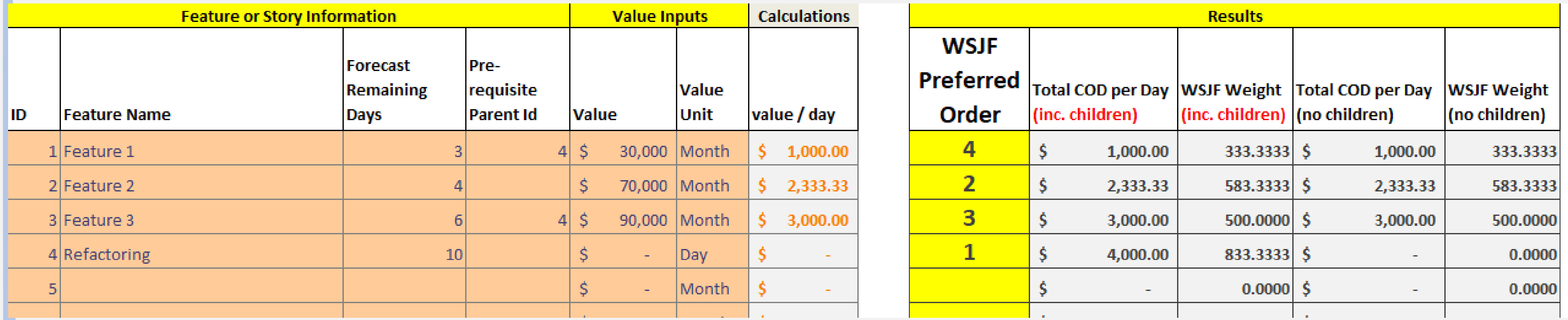
SAFe Consider more than one value type (estimated using Fibonacci point scale)

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

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

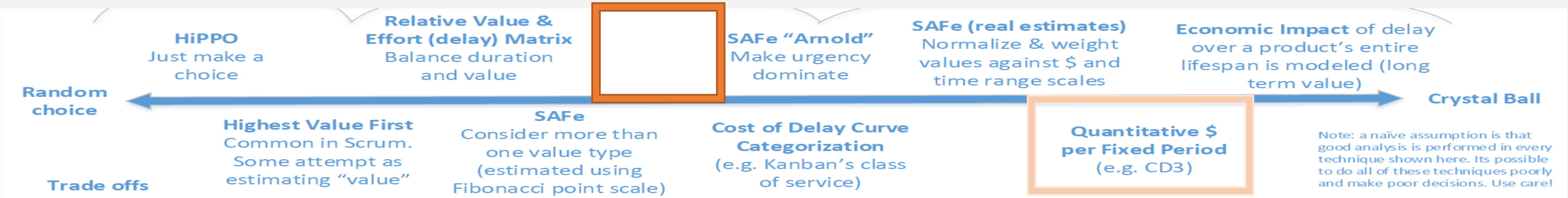
Note: a naïve assumption is that good analysis is performed in every technique shown here. Its possible to do all of these techniques poorly and make poor decisions. Use care!

Demo – COD spreadsheet (medium)



Feature or Story Information				Value Inputs		Calculations
ID	Feature Name	Forecast Remaining Days	Pre-requisite Parent Id	Value	Value Unit	value / day
1	Feature 1	3	4	$ 30,000	Month	$ 1,000.00
2	Feature 2	4		$ 70,000	Month	$ 2,333.33
3	Feature 3	6	4	$ 90,000	Month	$ 3,000.00
4	Refactoring	10		$ -	Day	$ -
5				$ -	Month	$ -

Results				
WSJF Preferred Order	Total COD per Day (inc. children)	WSJF Weight (inc. children)	Total COD per Day (no children)	WSJF Weight (no children)
4	$ 1,000.00	333.3333	$ 1,000.00	333.3333
2	$ 2,333.33	583.3333	$ 2,333.33	583.3333
3	$ 3,000.00	500.0000	$ 3,000.00	500.0000
1	$ 4,000.00	833.3333	$ -	0.0000
	$ -	0.0000	$ -	0.0000



Random choice

Crystal Ball

Trade offs

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)

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

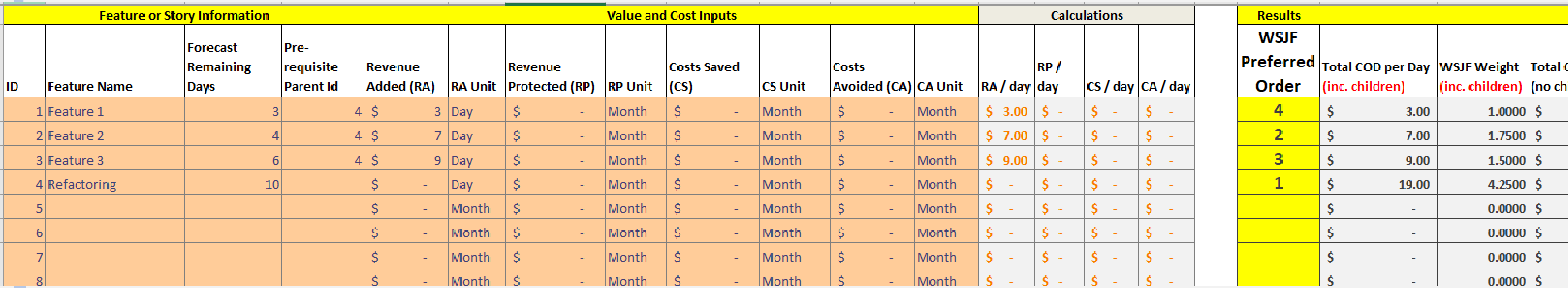
SAFe Consider more than one value type (estimated using Fibonacci point scale)

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

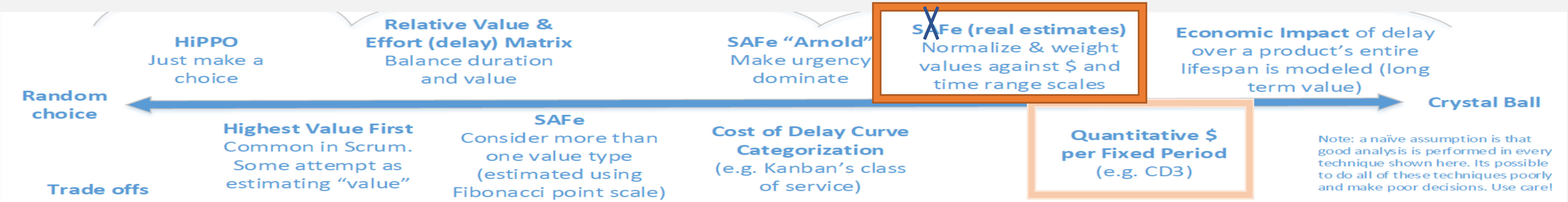
Quantitative $ per Fixed Period (e.g. CD3)

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!

Demo – COD spreadsheet (X-Large)



Feature or Story Information				Value and Cost Inputs								Calculations				Results			
ID	Feature Name	Forecast Remaining Days	Pre-requisite Parent Id	Revenue Added (RA)	RA Unit	Revenue Protected (RP)	RP Unit	Costs Saved (CS)	CS Unit	Costs Avoided (CA)	CA Unit	RA / day	RP / day	CS / day	CA / day	WSJF Preferred Order	Total COD per Day (inc. children)	WSJF Weight (inc. children)	Total C (no ch
1	Feature 1	3	4	$ 3	Day	$ -	Month	$ -	Month	$ -	Month	$ 3.00	$ -	$ -	$ -	4	$ 3.00	1.0000	$
2	Feature 2	4	4	$ 7	Day	$ -	Month	$ -	Month	$ -	Month	$ 7.00	$ -	$ -	$ -	2	$ 7.00	1.7500	$
3	Feature 3	6	4	$ 9	Day	$ -	Month	$ -	Month	$ -	Month	$ 9.00	$ -	$ -	$ -	3	$ 9.00	1.5000	$
4	Refactoring	10		$ -	Day	$ -	Month	$ -	Month	$ -	Month	$ -	$ -	$ -	$ -	1	$ 19.00	4.2500	$
5				$ -	Month	$ -	Month	$ -	Month	$ -	Month	$ -	$ -	$ -	$ -		$ -	0.0000	$
6				$ -	Month	$ -	Month	$ -	Month	$ -	Month	$ -	$ -	$ -	$ -		$ -	0.0000	$
7				$ -	Month	$ -	Month	$ -	Month	$ -	Month	$ -	$ -	$ -	$ -		$ -	0.0000	$
8				$ -	Month	$ -	Month	$ -	Month	$ -	Month	$ -	$ -	$ -	$ -		$ -	0.0000	$



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

Trade offs

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

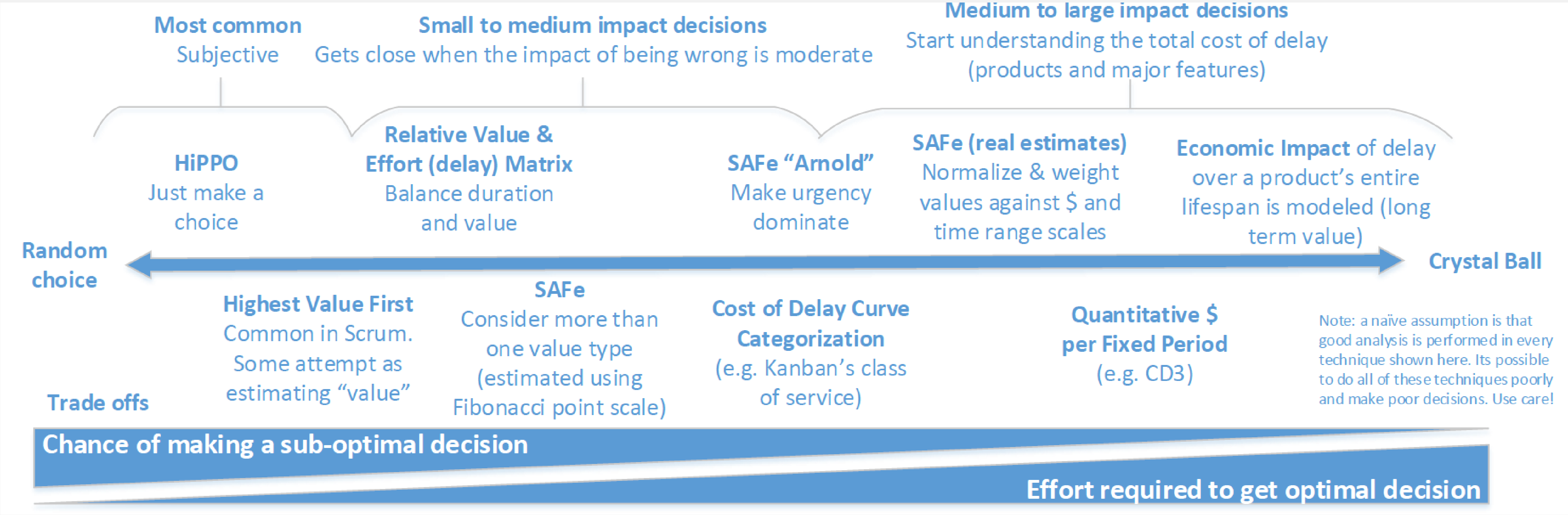
SAFe
Consider more than one value type (estimated using Fibonacci point scale)

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!

How to choose…



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)

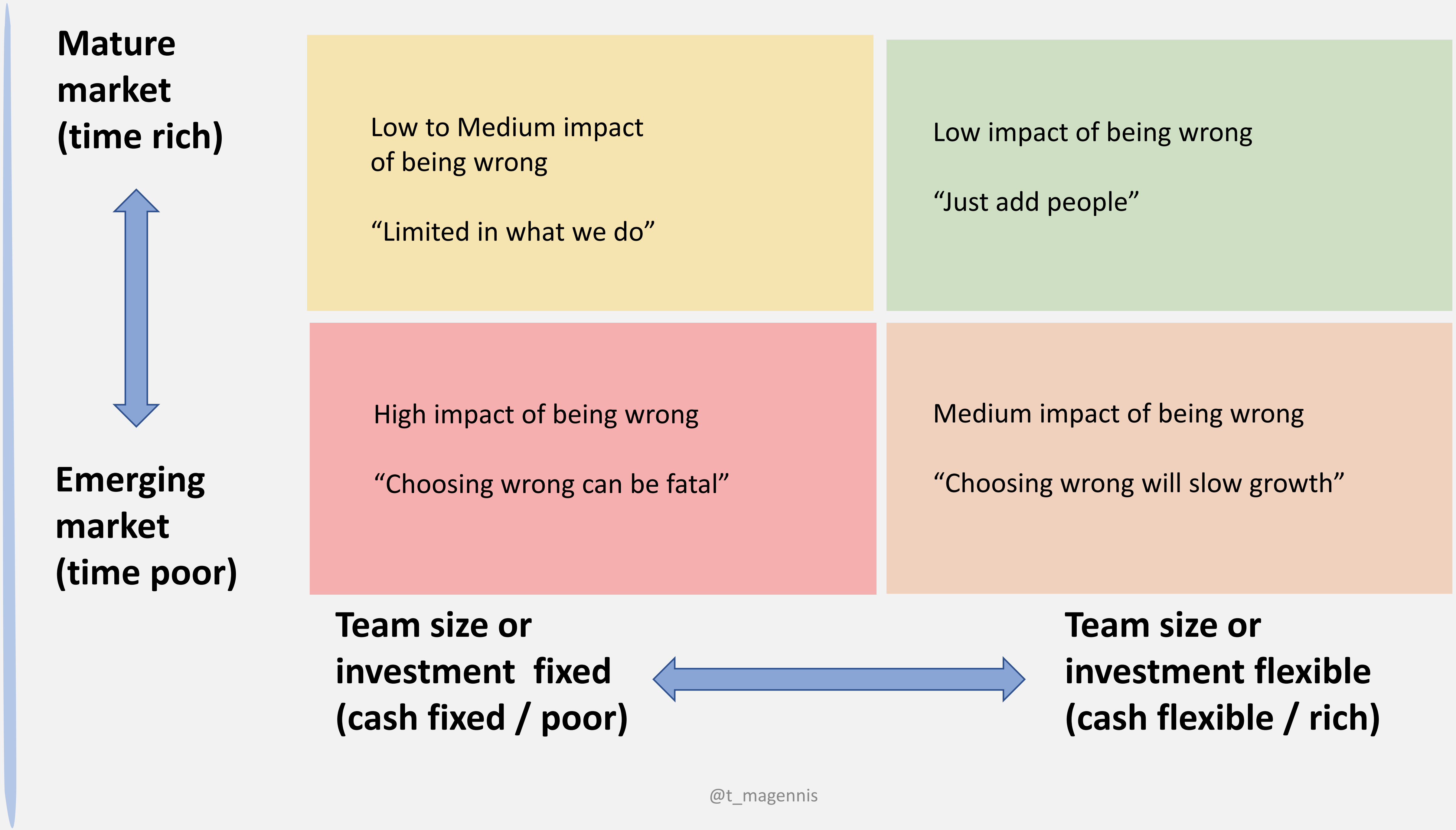
Quantitative $ per Fixed Period
(e.g. CD3)

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

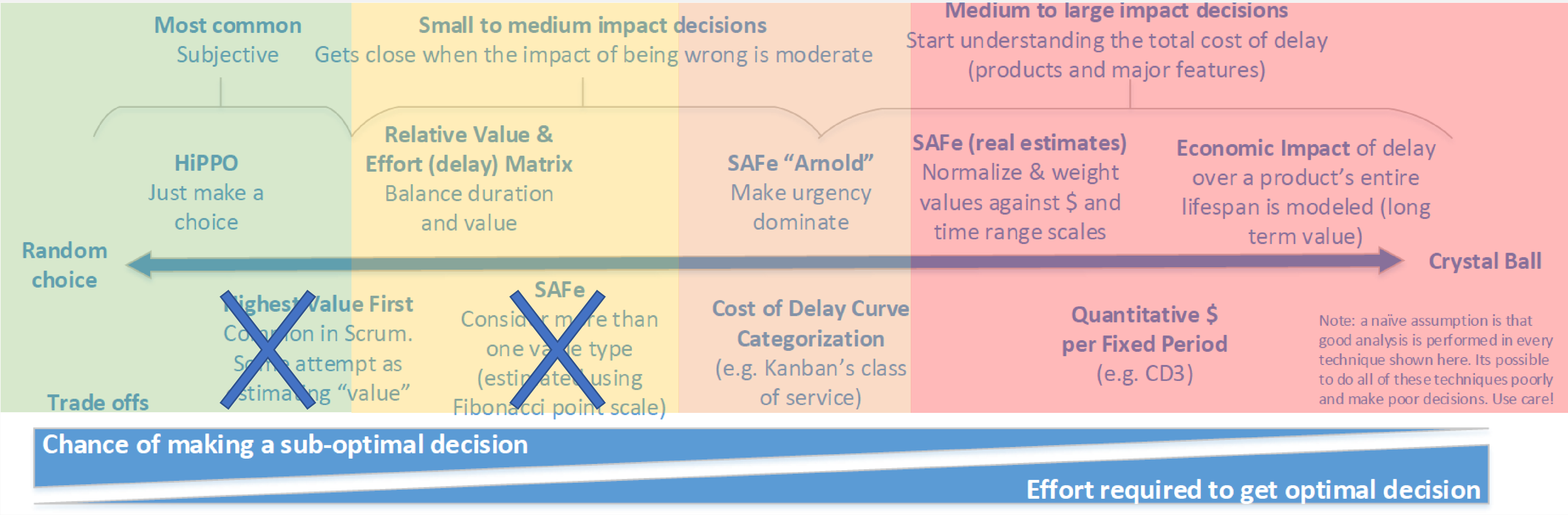
Effort required to get optimal decision



Mature market (time rich)	Low to Medium impact of being wrong “Limited in what we do”	Low impact of being wrong “Just add people”
Emerging market (time poor)	High impact of being wrong “Choosing wrong can be fatal”	Medium impact of being wrong “Choosing wrong will slow growth”
	Team size or investment fixed (cash fixed / poor)	**Team size or investment flexible (cash flexible / rich)**

@t_magennis

Choose based on impact of being 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!

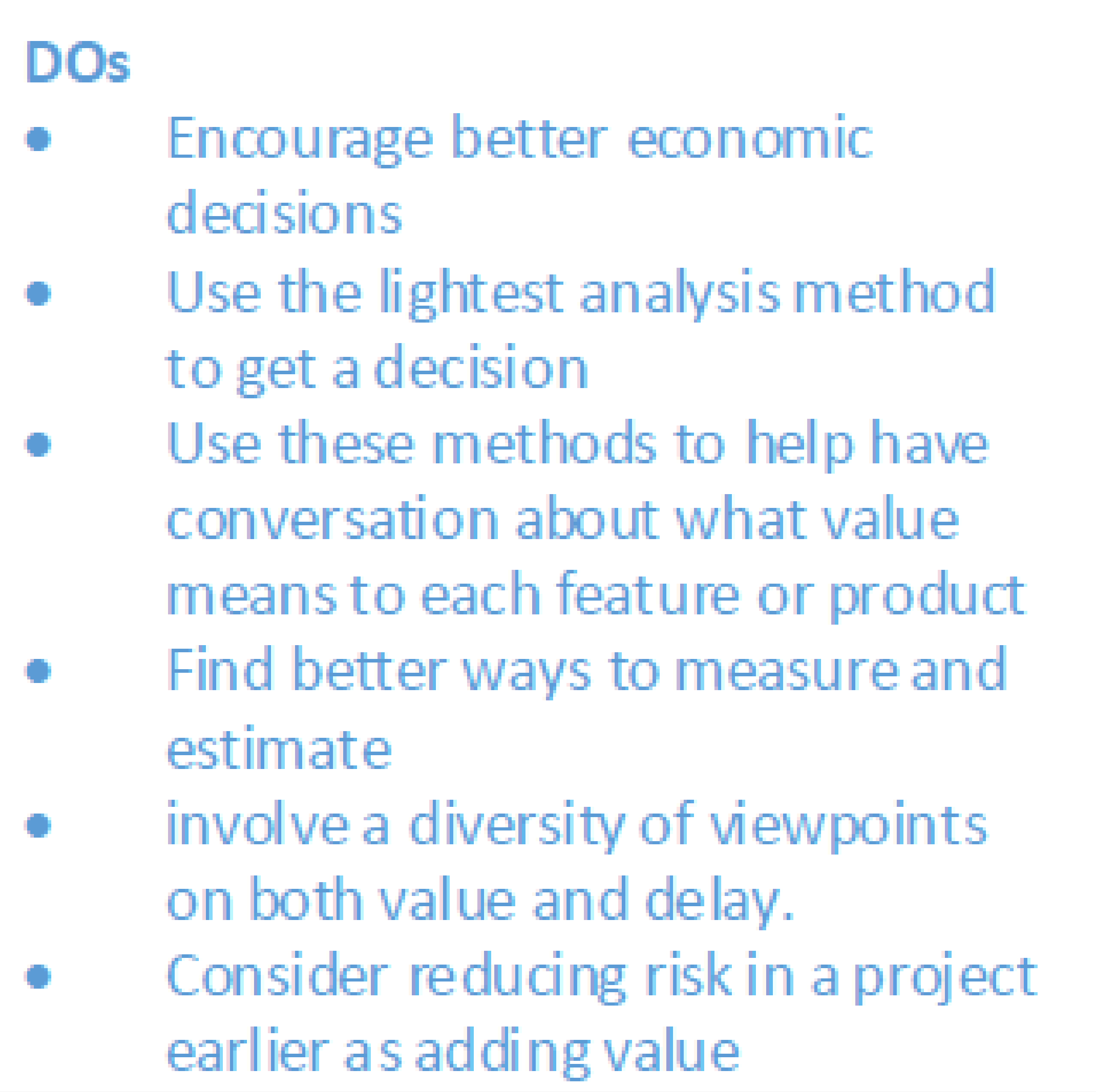
Trade offs

Chance of making a sub-optimal decision

Effort required to get optimal decision

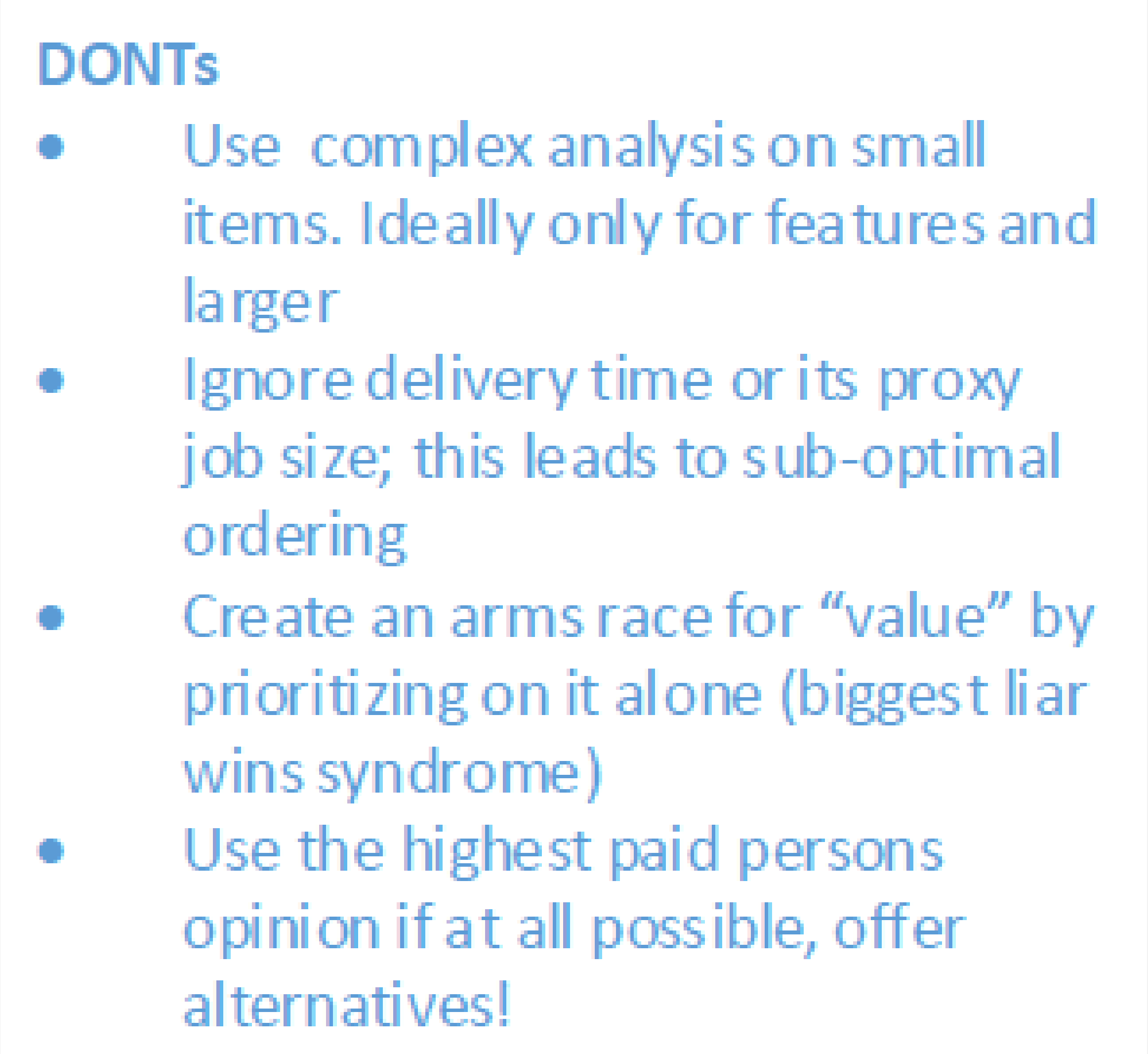
X Don't use. Better option at similar effort

@t_magennis



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!

http://bit.ly/BetterPrioritization

Questions and Discussion…

- Too shy now?
  - Email: troy.magennis@gmail.com
  - Twitter: @t_magennis
- Want the slides or spreadsheets or resources?
  - http://Bit.Ly/**S**im**R**esources
  - http://Bit.Ly/**B**etter**P**rioritization
- Please consider doing the feedback and review…