I LOVE the smell of data in the morning

Getting started with data science

@t_magennis | troy.magennis@focusedobjective.com

All slides and spreadsheets: Bit.ly/SimResources

Agile alliance Code of Conduct: bit.ly/Agile2017COC

Slides, spreadsheets, and other stuff

Bit.ly/SimResources

Everything you see is freely available

Questions…

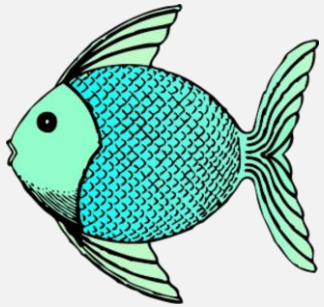
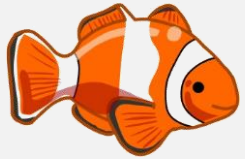
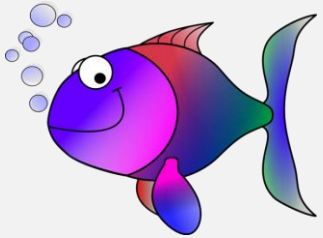
- How many defects? (delivery readiness)
- How are we doing? (process opportunities)
- How big? (size of features)
- How long? (expected time to deliver from “here”)
- Where is the constraint? (sensitivity analysis)

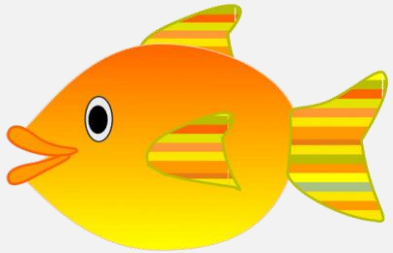
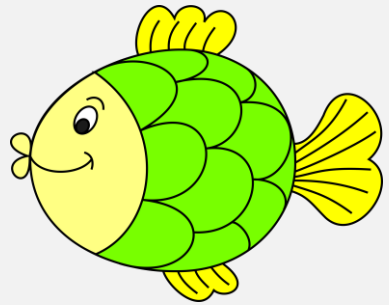
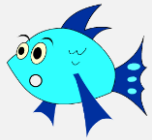
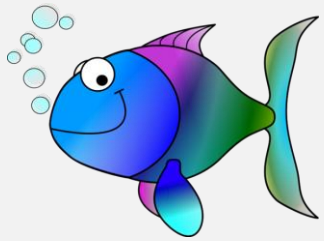
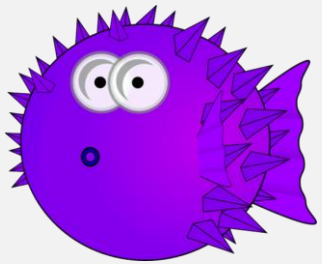
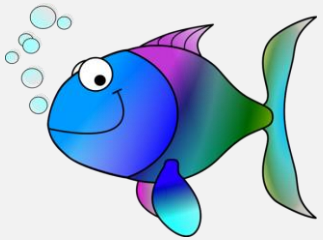
Q: How Many Defects?

Understanding if more testing is worth it, or can (should) we release yet!

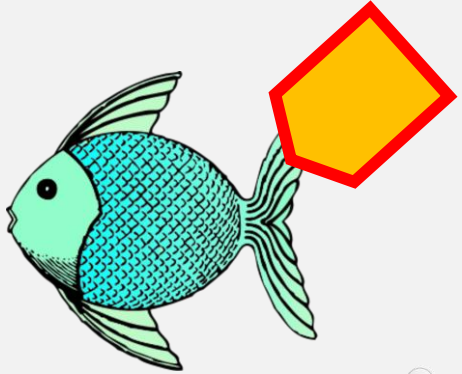
How Many Fish?

(assume you can't see them, and you shouldn't kill them)

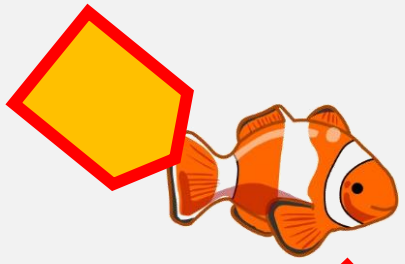







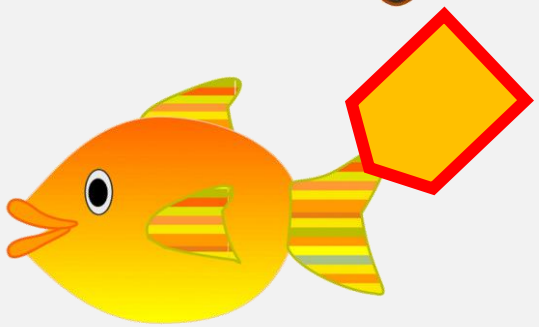
5 fish caught

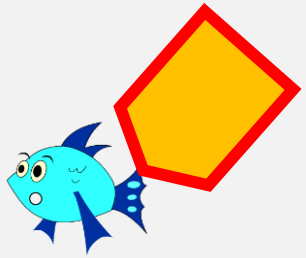


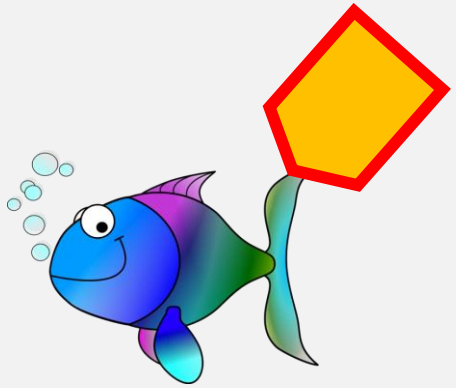
We Go Fishing, Tag & Return

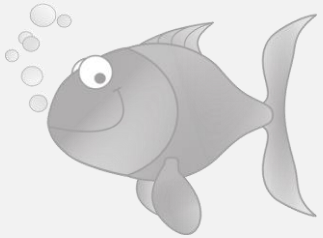
We got FIVE fish.

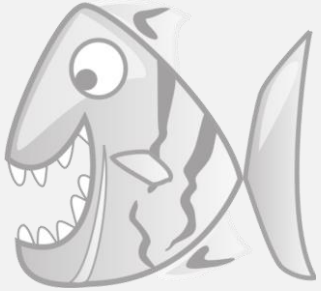


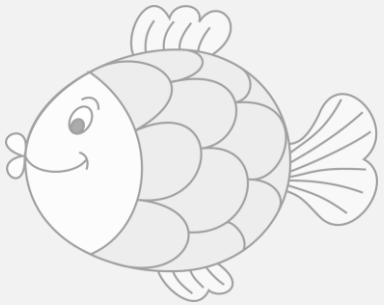


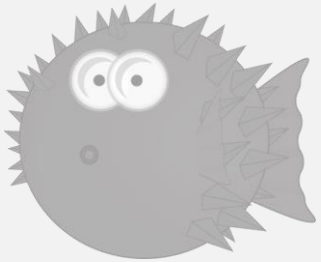


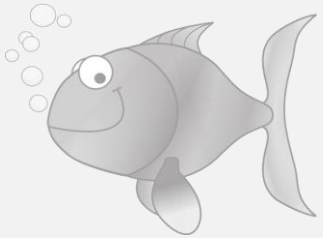








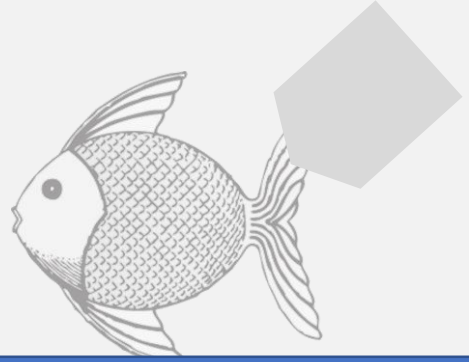




4 fish caught

2 x tagged

2 x un-tagged



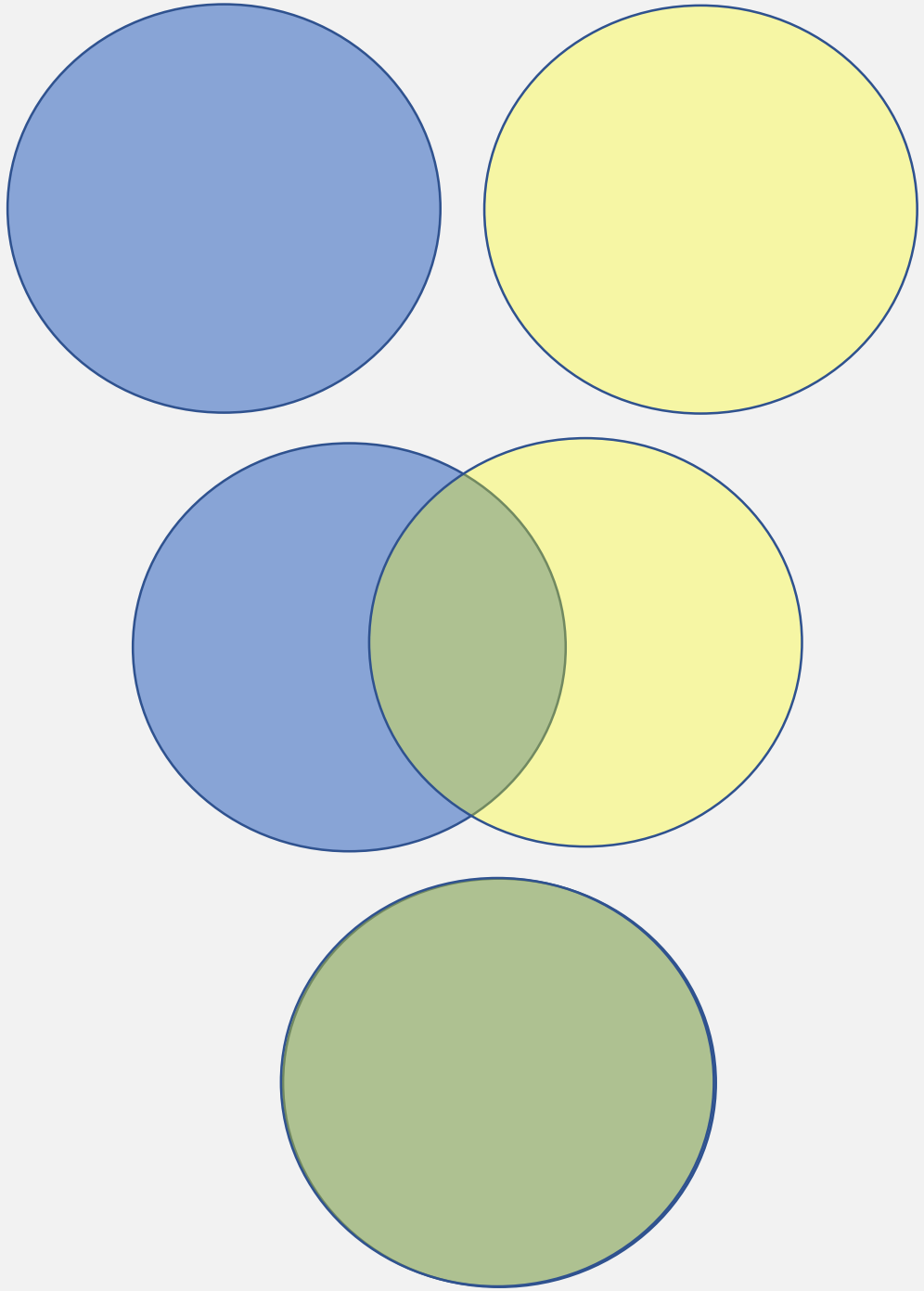
We Go Fishing AGAIN

Ratio of Tagged vs Untagged Answers Our Question
(knowing what we know, 50/50 is the expected result)

$$\frac{\text{Total Caught First Time} \times \text{Total Caught Second Time}}{\text{Total Caught Both Times (tagged fish)}}$$

$$\frac{5 \text{ fish} \times 4 \text{ fish}}{2 \text{ tagged fish}} = \frac{20}{2}$$

= 10 fish total



LOTS MORE TO FIND

MORE TO FIND

MOST FOUND

Latent Defect

$$\text{Estimated total defects} = \frac{\text{Found by group A} \times \text{Found by group B}}{\text{Found by BOTH group A and B}}$$

Defect	Found by Group A	Found by Group B
Fish 1	Yes	No
Fish 2	Yes	No
Fish 3	No	Yes
Fish 4	Yes	Yes
Fish 5	Yes	No
Fish 6	Yes	Yes
Fish 7	No	Yes

Total A	5
Total B	4
Only A	3
Only B	2
Both	2
Total	7

Total Estimated Defects

10

Estimated Un-discovered Defects

3

http://Bit.Ly/SimResources

The properties of the “few,”

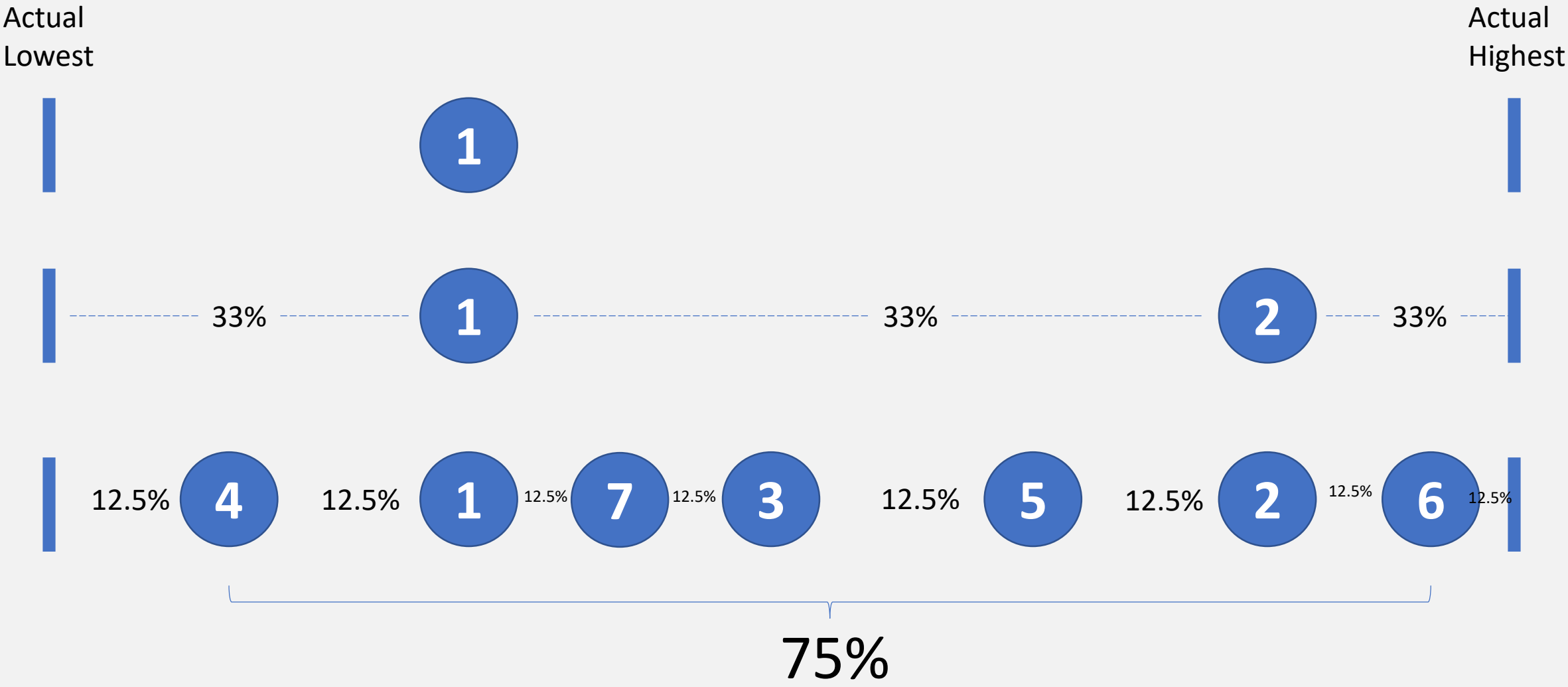
Help understand the properties of the “all.”

Likely low and upper ranges of values

Likely future values

Common versus un-common

Defining “Few” (How few? = how certain?)



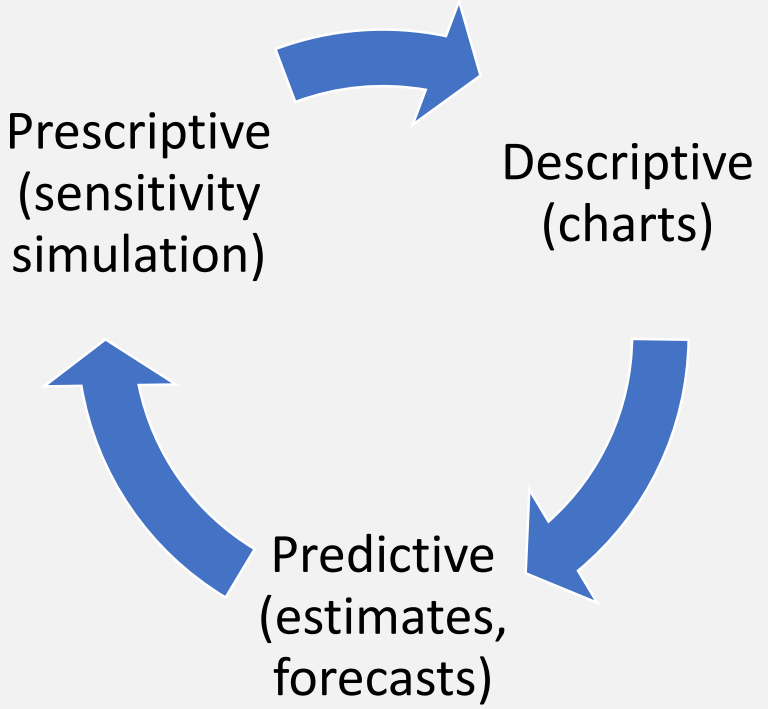
Observations So Far (n)	Below lowest seen so far	Within range seen so far	Above highest seen so far
1	-	-	-
2	33%	33.3%	33%
3	25%	50%	25%
4	20%	60%	20%
5	16.7%	66.6%	16.7%
6	14.3%	71.4%	14.3%
7	12.5%	75%	12.5%
8	11.1%	77.8%	11.1%
9	10%	80%	10%
10	9.1%	81.8%	9.1%

Assumptions (rarely perfectly true)

- The samples are taken at random. Convenient isn't random!
- The distribution is relatively uniform; All values have similar chance.
- The probability is on average.

Types of Analytics

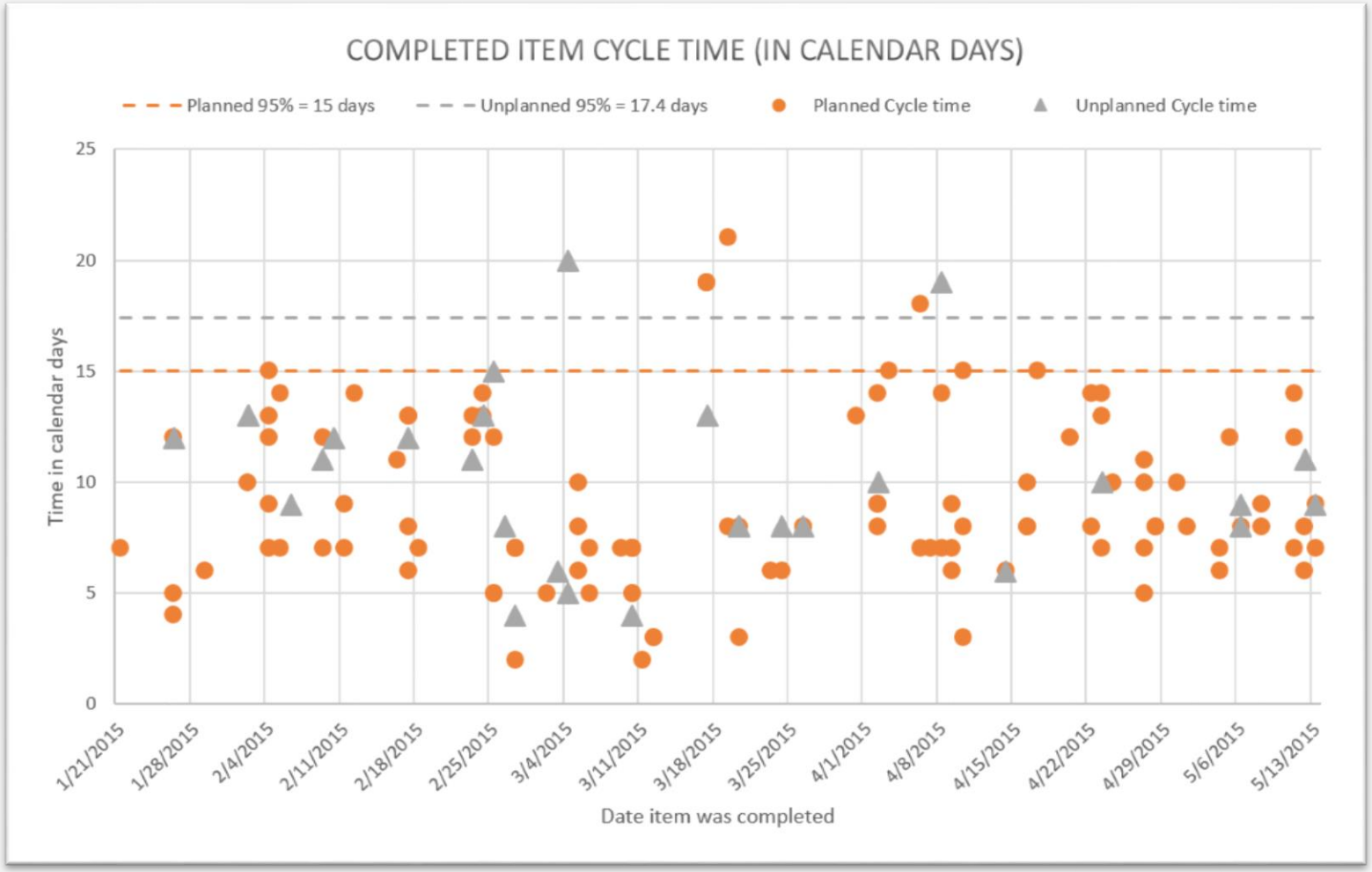
- **Descriptive Analytics** "What has happened?"
  - Data aggregation and data mining to provide insight into the past
- **Predictive Analytics** "What could happen?"
  - Statistical models and forecasts techniques to understand potential futures
- **Prescriptive Analytics** "What should we do?"
  - Optimization and simulation algorithms to give advice based on possible outcomes

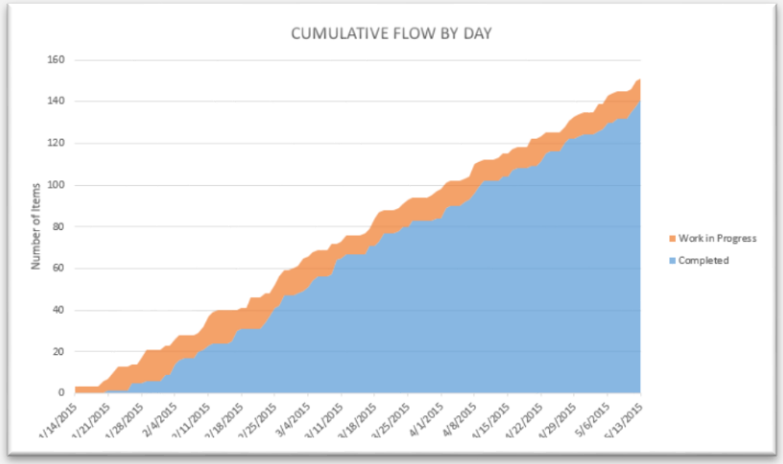


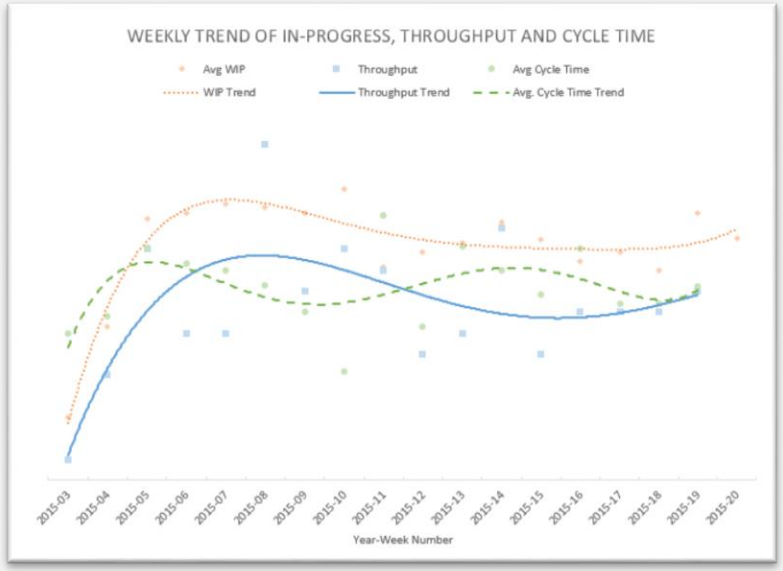
Q: How Are We Doing?

Understanding process improvement

Descriptive Analytics AKA Charts and Stats





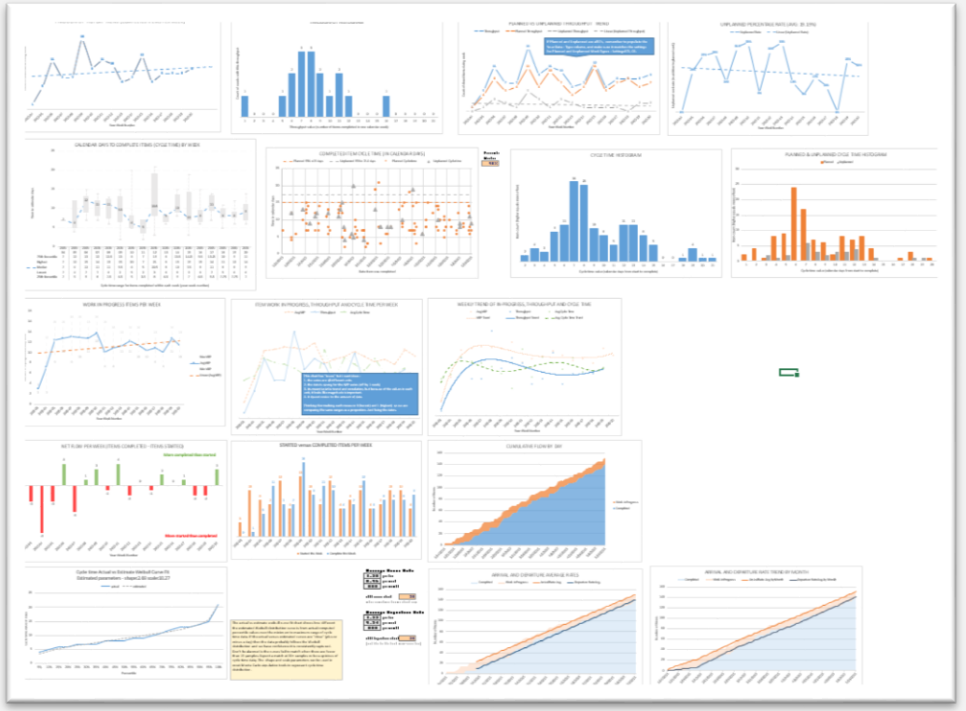


But How Do I Get the Data?

Completed Date	Start Date (optional)	Type (optional)
1/21/2015	1/14/2015	
1/26/2015	1/14/2015	Story
1/26/2015	1/14/2015	Defect
1/26/2015	1/21/2015	Story
1/26/2015	1/22/2015	Story
1/29/2015	1/23/2015	Story
2/2/2015	1/23/2015	Story
2/2/2015	1/20/2015	Defect

=

18 Charts




Troy Magennis
@t_magennis
Multiple NOT single metric: Easy to improve one metric. But at what cost? Use a set of metrics that capture damage of over-driving any one.
1. Quality
(how well)
• Escaped defect counts
• Forecast to complete defects
• Measure of release "readiness"
• Test count (passing)
3. Responsiveness
(how fast)
• Lead time
• Cycle time
• Defect resolution time
2. Productivity
(how much, delivery pace)
• Throughput
• Velocity
• Releases per day
4. Predictability
(how repeatable)
• Coefficient of variation (SD/Mean)
• Standard deviation of the SD
• "Stability" of team & process
SDPI
RETWEETS
36
LIKES
52
11:43 AM - 18 Feb 2017
3
36
52

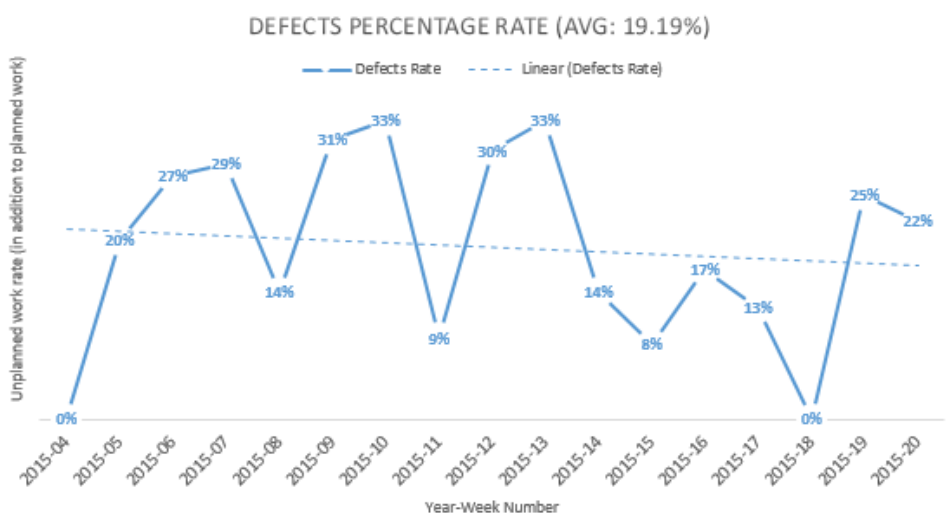
My Awesome Team Dashboard

(change your team name in the Settings t

Type Filter All (Note be patient, it can take 10-20 se

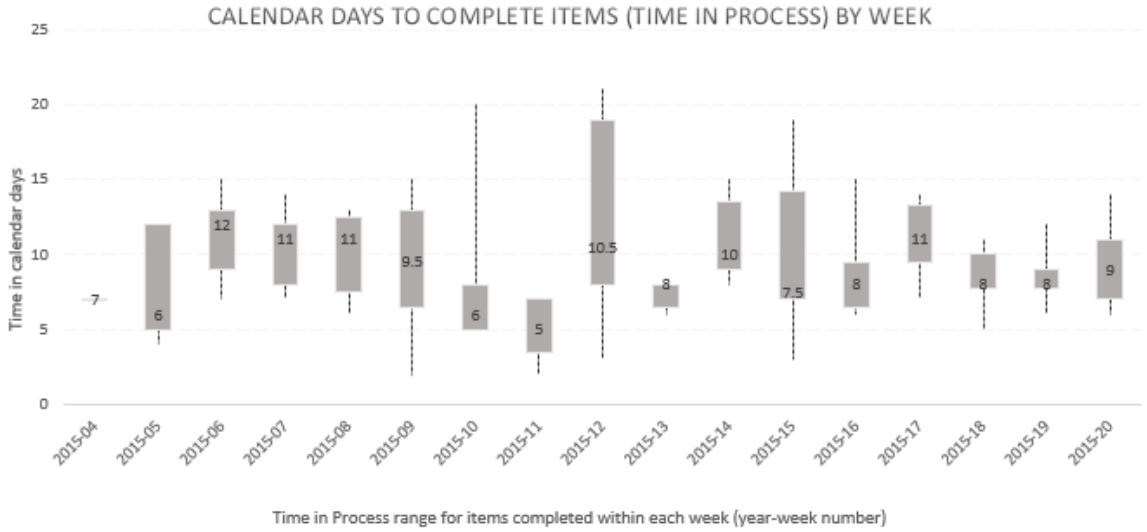
Quality - How Well

How much defect debt do we carry that impedes consistent delivery?



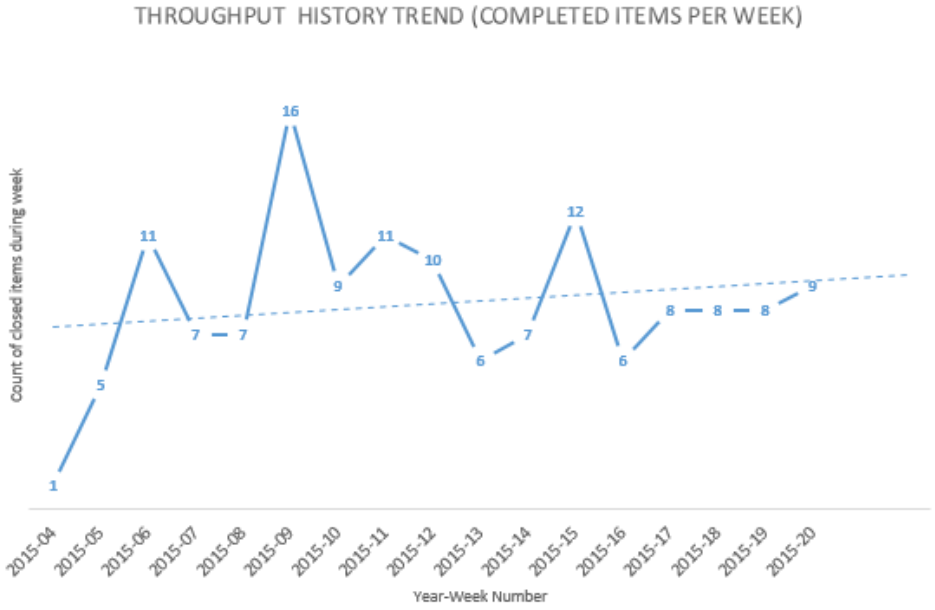
Responsiveness - How Fast

How fast do we deliver from starting something to finishing it? Goal: Look for ways to reduce time and variability for similar work.



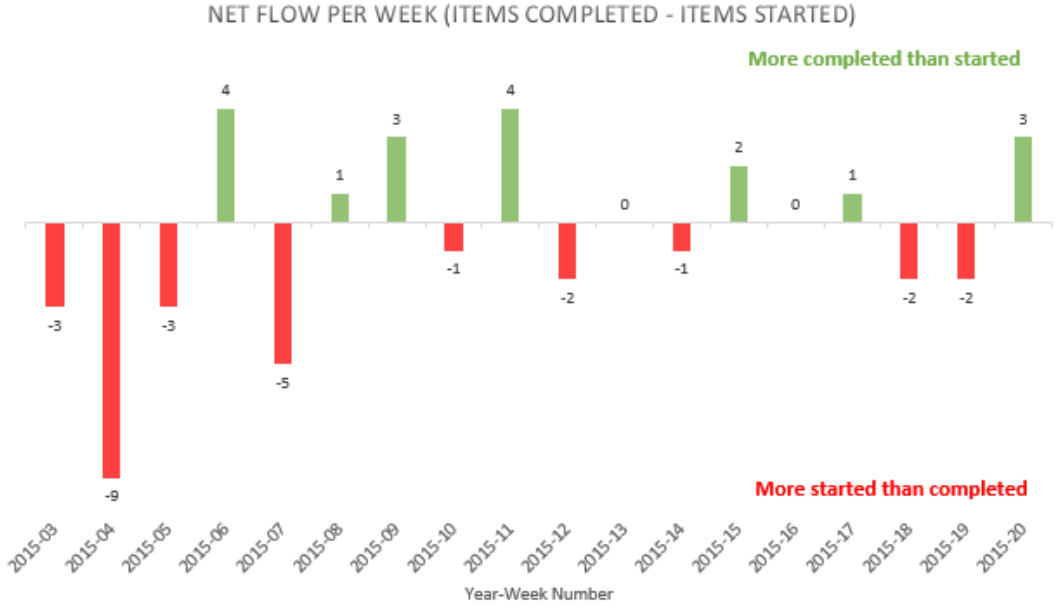
Productivity - How Much

What pace do we deliver work? Goal: Look for ways to increase. Watch quality if you overdrive.



Predictability - How Repeatable

How consistent is our completion pace? Goal: Keep around zero. Complete something in preference to starting.



Team Dashboard.xlsx spreadsheet

Bit.Ly/SimResources

Getting started with Descriptive Analytics

- Start capturing Start Date, Complete Date and Work Type Data
- Start making this data available for analysis
- Start talking about this data in retrospectives and other meetings
- Confirm process improvements have desired impact and don't have un-intended consequences elsewhere

Team Dashboard.xlsx spreadsheet at http://Bit.Ly/SimResources

Predictive Analytics

Q: How Big?

Understanding the size of a feature or project with less effort

Step 1

Feature 2
3 stories

Feature 3
7-15 stories

Feature 1
15 stories

Step 2

Feature 4
?

Step 3

Feature 2
3 stories

Feature 3
7-15 stories

Feature 1
15 stories

Feature 4
10-15 stories

Known as Reference Class Forecasting

Forecasting Total Story Count

- Question: How an I estimate the size of a feature or project without analyzing every piece of work?

- Theory: The “size” patterns of randomly sample epics, will persist through all other epics. Analyze a few and compute for the many…

http://bit.ly/StoryCountForecaster

Sampling based Monte Carlo story count forecasting Excel spreadsheet

Feature or Epic Name	Estimated # Stories or points (before starting)
Feature 1	5
Feature 2	3
Feature 3	8
Feature 4	4
Feature 5	2
Feature 6	7
Feature 7	
Feature 8	
Feature 9	
Feature 10	
Feature 11	
Feature 12	
Feature 13	
Feature 14	
Feature 15	

Process to estimate total size –

1. Pick a 5-10 features at random
2. Build sets of 15 re-samples (say 1000 times)
3. The number of sets that reach certain story count levels give probability

1. How many total features do you want to forecast? 15 total features entered on input sheet:

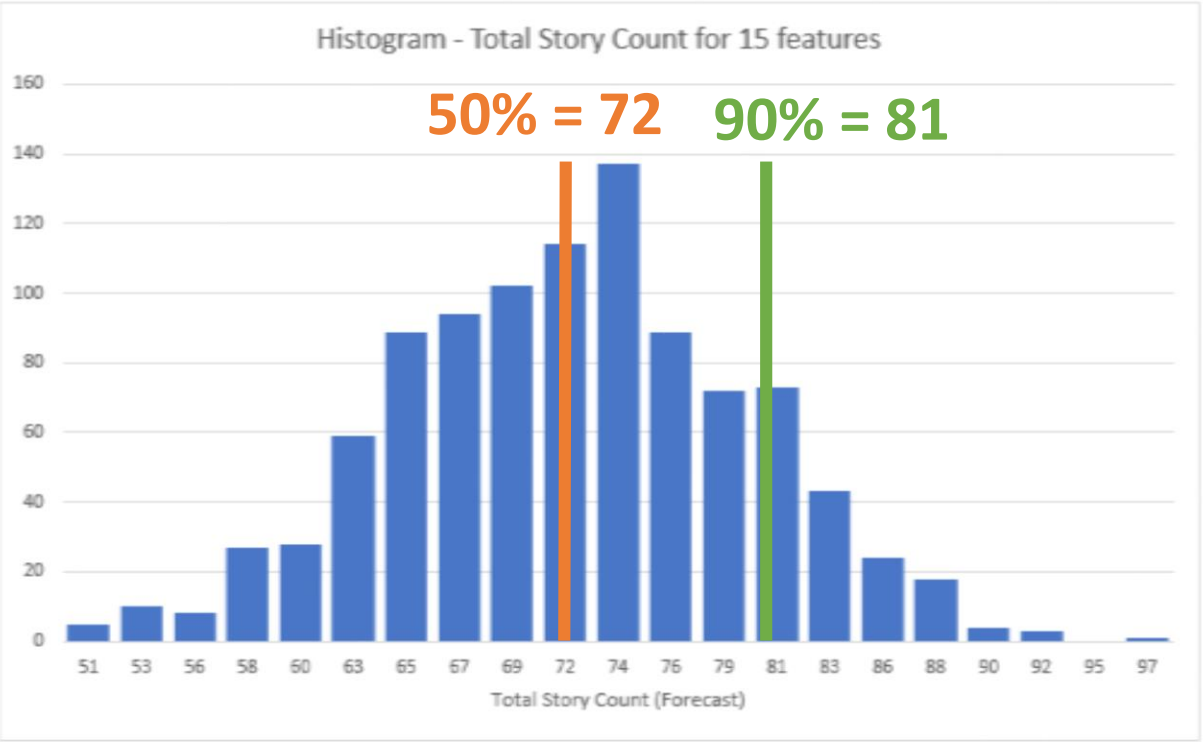
Enter the total number of features or epics you wish to forecast. The patterns exhibited by the story count breakdown of the samples fetures and epics will be extrapolated to this many total features.

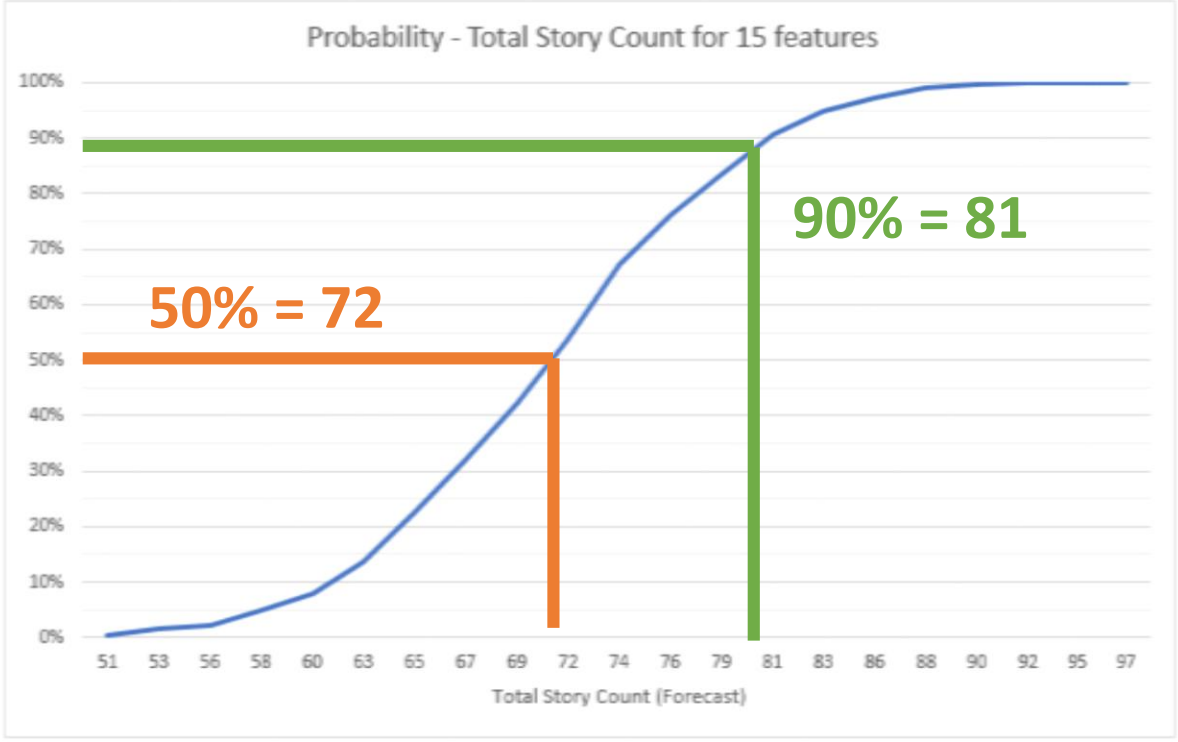
2. What rate do you expect work to split? low guess 1 high guess 1 actual

Work often splits into smaller pieces when started by the team. Also, new work gets discovered through defets and learning. Account for that here. 1 no change, 2 means every one item might be split into two, 3 means every item might become three items, etc. Most commn range I've seen is 1 to 3.

3. Result: Forecast total story count or total story points

Likelihood	Total Story Count/points	Odds in english
50%	73	50% = Coin toss odds. Same chance being above or below this story count
85%	81	85% = Pretty sure to be equal or less than this story count.
95%	85	95% = Almost certain to be equal or less than this story count.





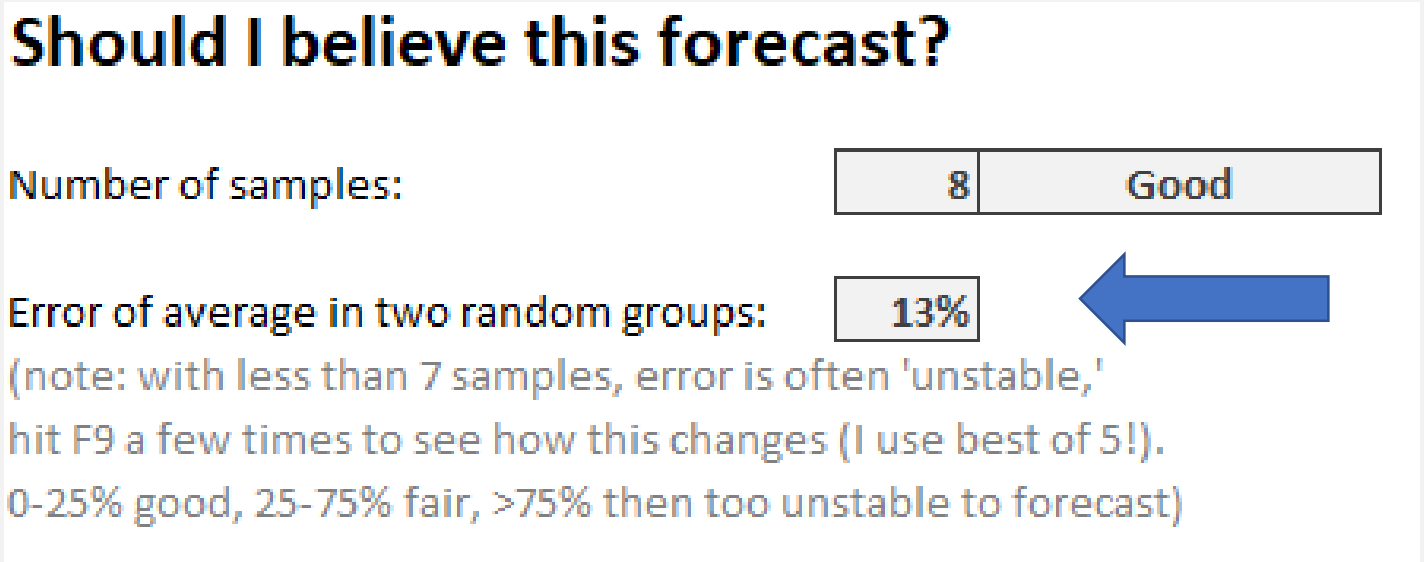
This chart plots how likely each total story count result occurred in the Monte Carlo simulation. The higher the bar, the more often that count of stories occurred. It is used to understand the pattern of the results to see how wide the tail values extend (those either side of the peak). If the range is too wide to make forecasting useful, the only solution is to make the features more consistent in size to avoid the story count estimate range being so huge.

This chart plots how likely each total story count result is as cumulative probability the result is less than or equal to a total story count. If you want to know what total story count is 80% likely, look at where the 80% on the Y-Axis intersects the blue line, and read off the total story count on the X-Axis.

Why should I believe this forecast anyway?

1. Sample Count: Keep cutting data and compare the result
2. Random groups: Split data into random groups and compare

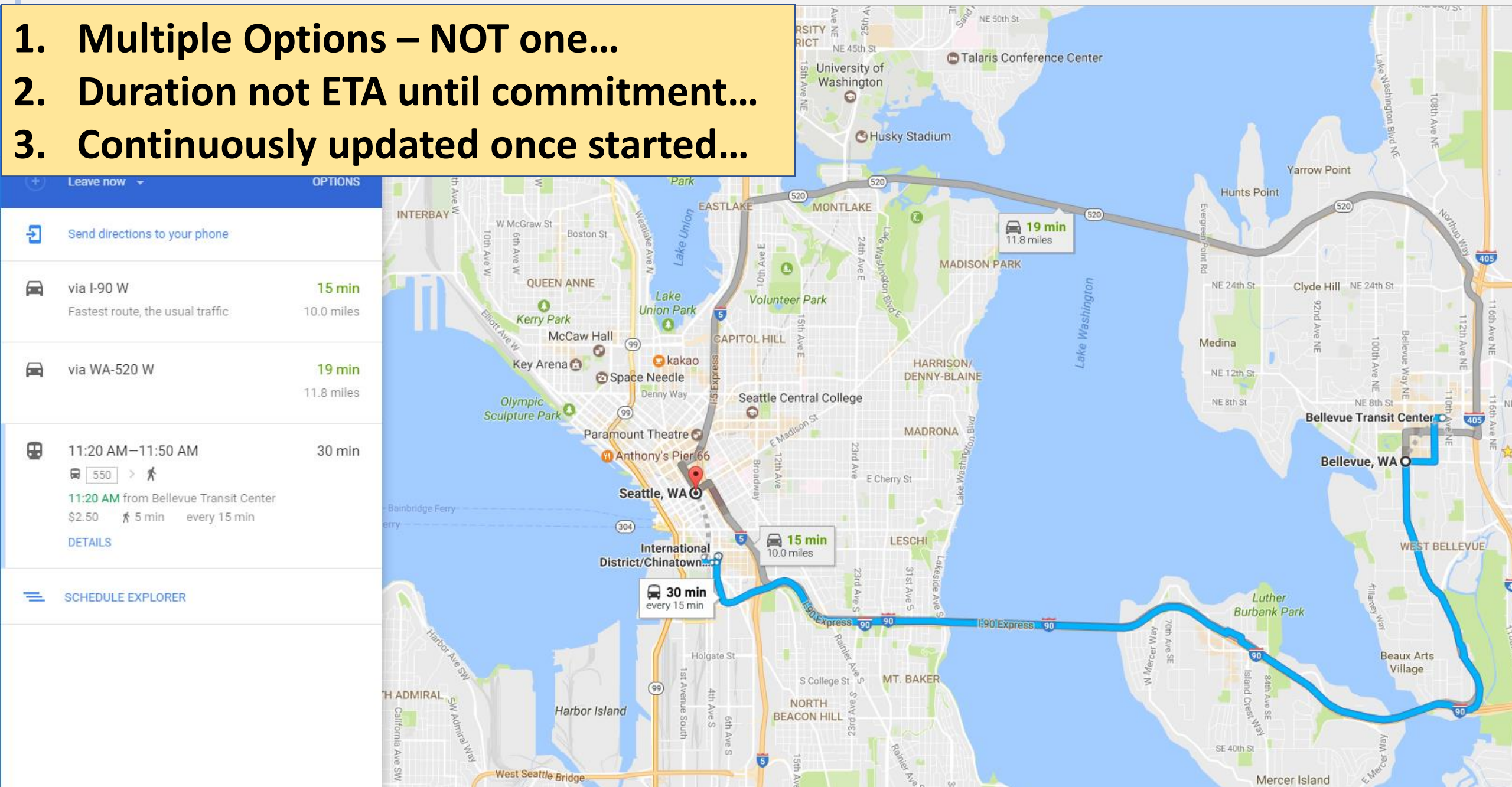
Total for 100 Features using	Total Count 85% Likelihood
36 samples	506
10 samples	494
3 samples	504



Average Error calculation –

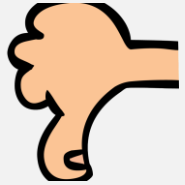
1. Split the samples into 2 groups
2. Calculate the average of both groups
3. Compare the difference as a % of range
   error % = error of avg / (max-min)

1. **Multiple Options – NOT one…**
2. **Duration not ETA until commitment…**
3. **Continuously updated once started…**



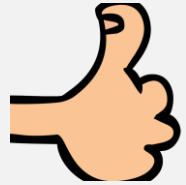
Contrast Google Maps to Software Estimates

Current Way



- Give one forecast even though multiple approaches considered
- Give a calendar date for undefined “complete” & “start”
- If the original date is in doubt we find out near the end
- Appear on-time until we are not. Measure progress from start.

Better Way



- Give multiple options of investment and implementation
- Give a duration and define what started & complete means
- If the original date is in doubt, know earlier and react faster
- Report remaining time to deliver not time since started

“Remember that all models are wrong; the practical question is how wrong do they have to be to not be useful.”

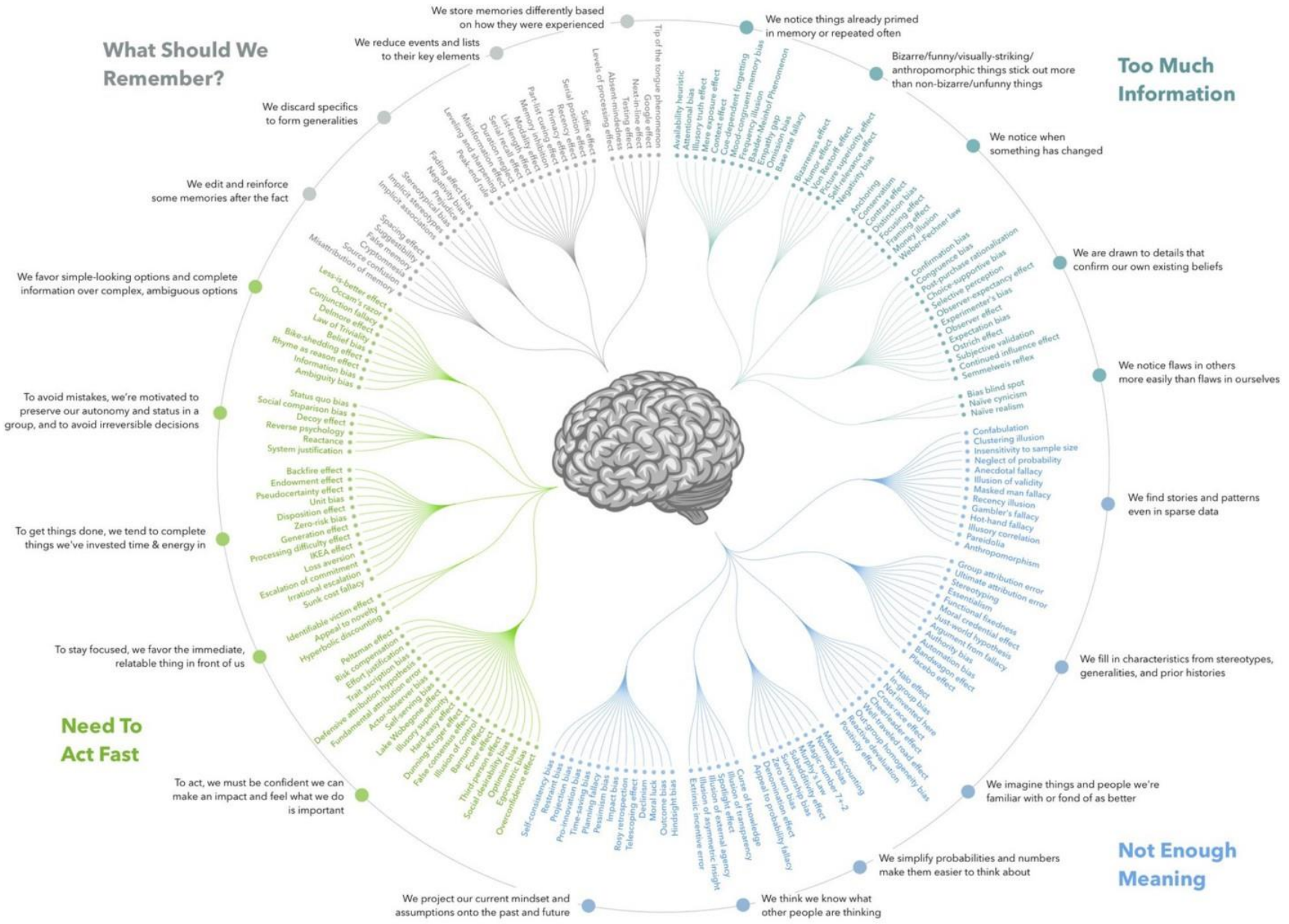
Statistician,
George Box

A. Better than intuition

Not perfect. Not exact. Not always right.

Just better than what you do now, or even equal (just less expensive)

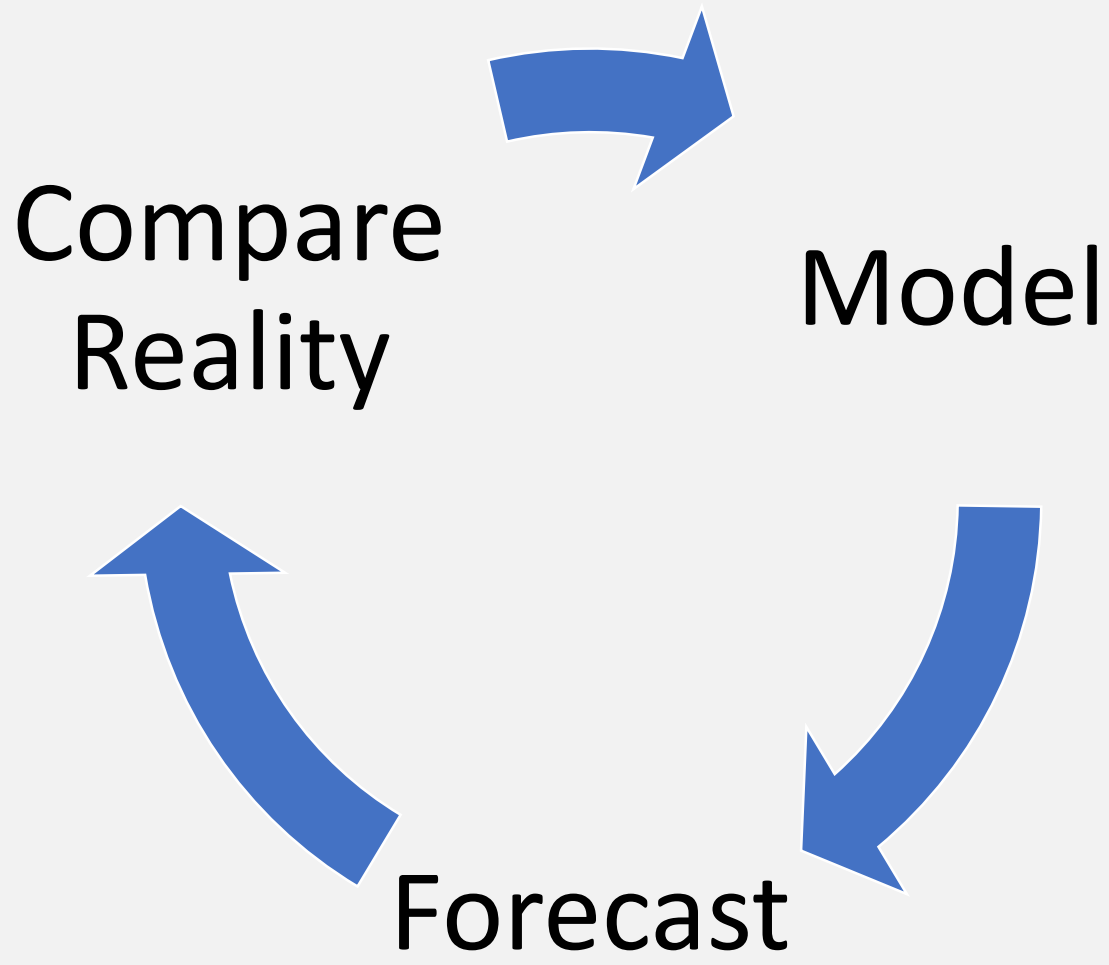
COGNITIVE BIAS CODEX, 2016



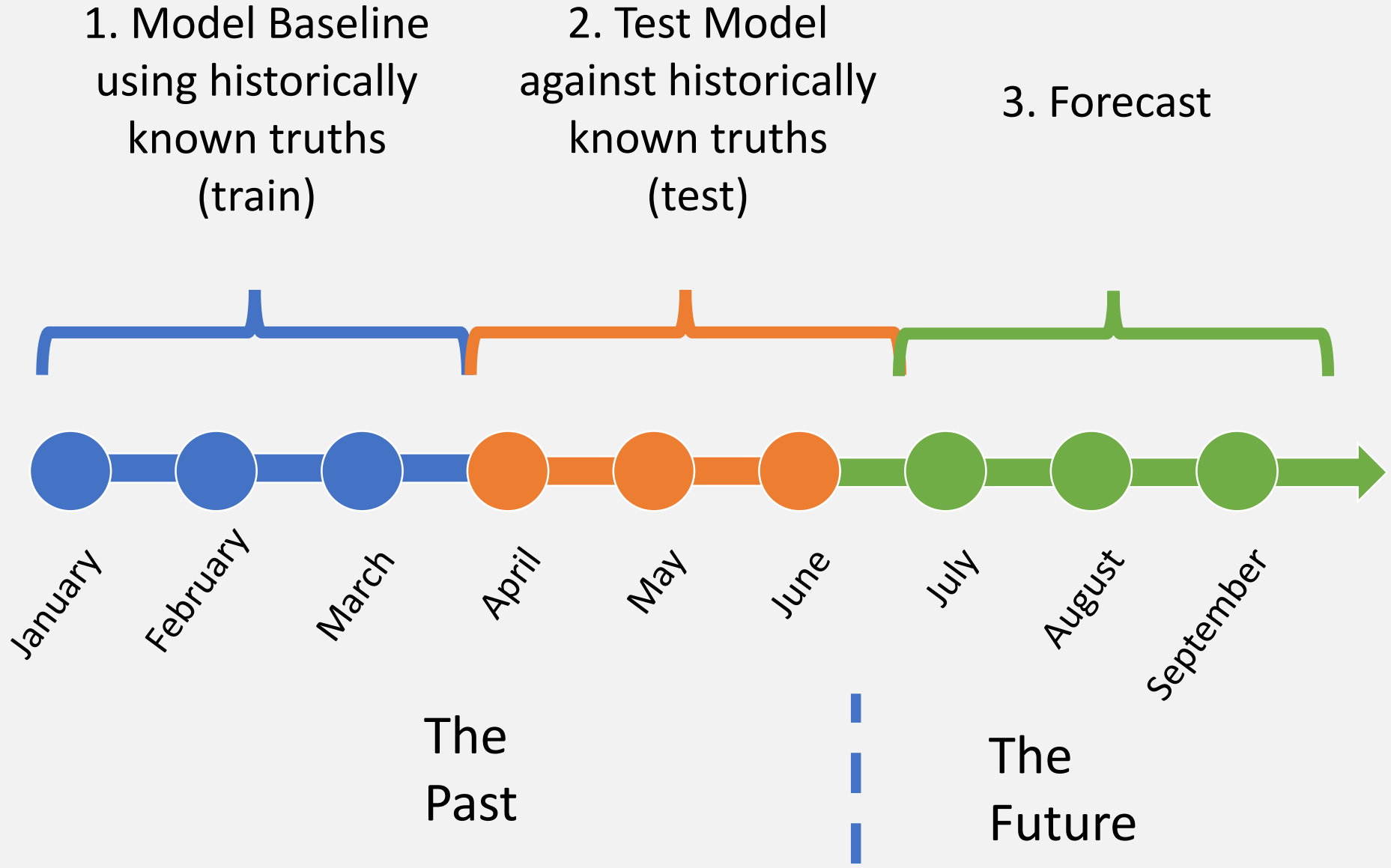
ALGORITHMIC LAYOUT + DESIGN BY JM3 · JOHN MANOOGIAN III // CONCEPT + METICULOUS CATEGORIZATION BY BUSTER BENSON // DEEP RESEARCH BY WIKIPEDIANS FAR + WIDE

http://ritholtz.com/2016/09/cognitive-bias-codex/

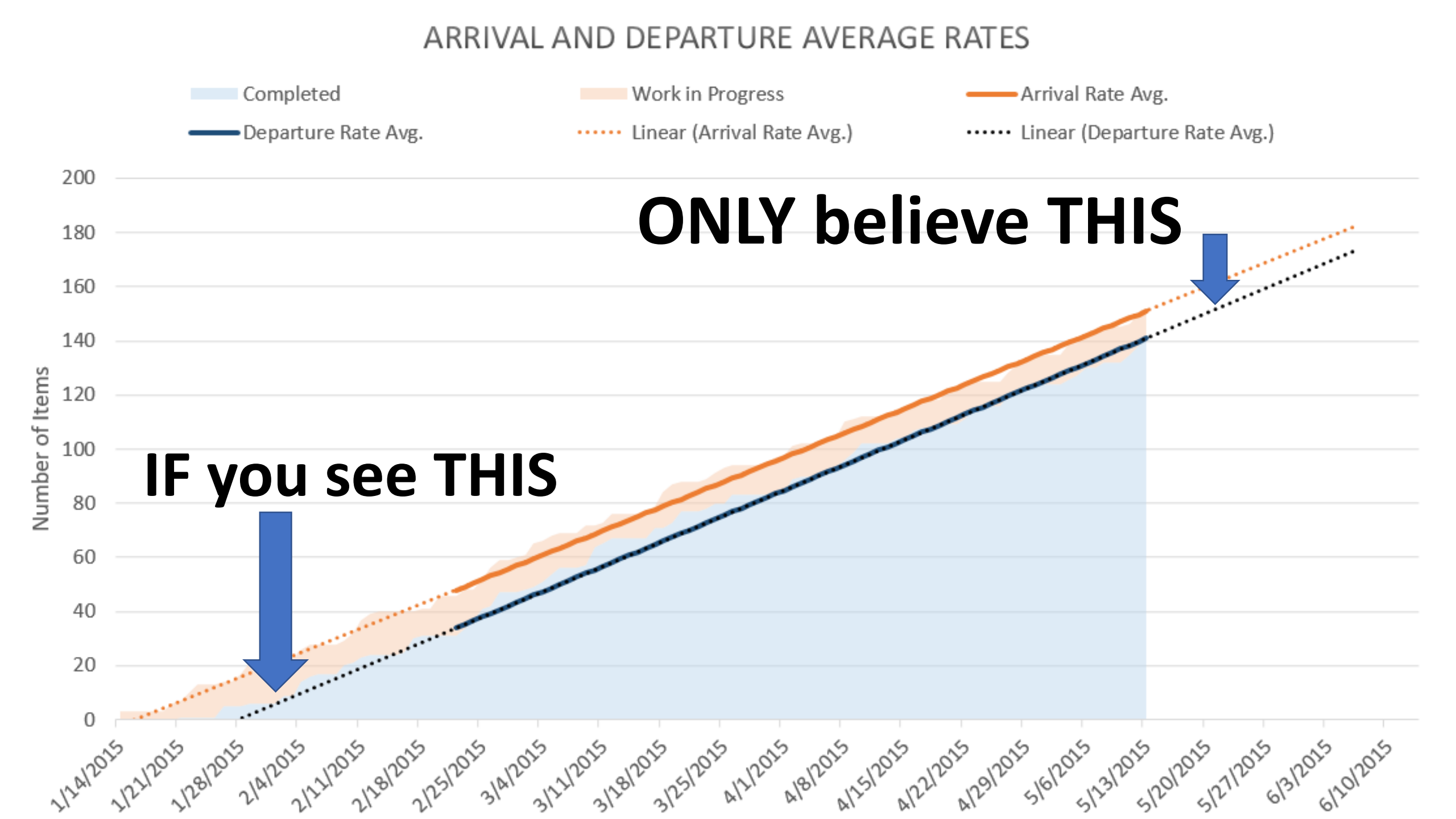
http://ritholtz.com/2016/09/cognitive-bias-codex/



Models are always wrong.
It’s all about understanding why.

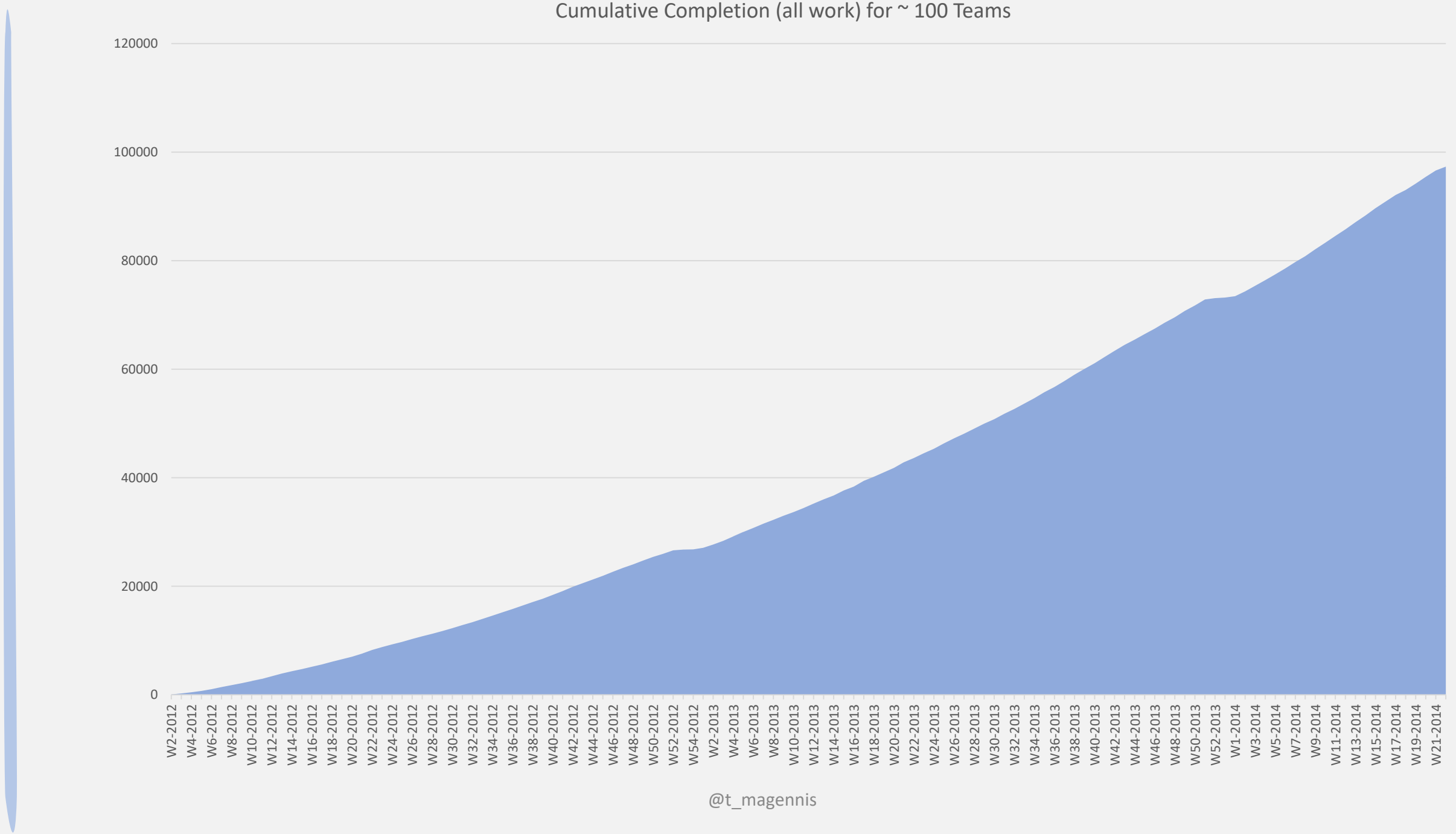


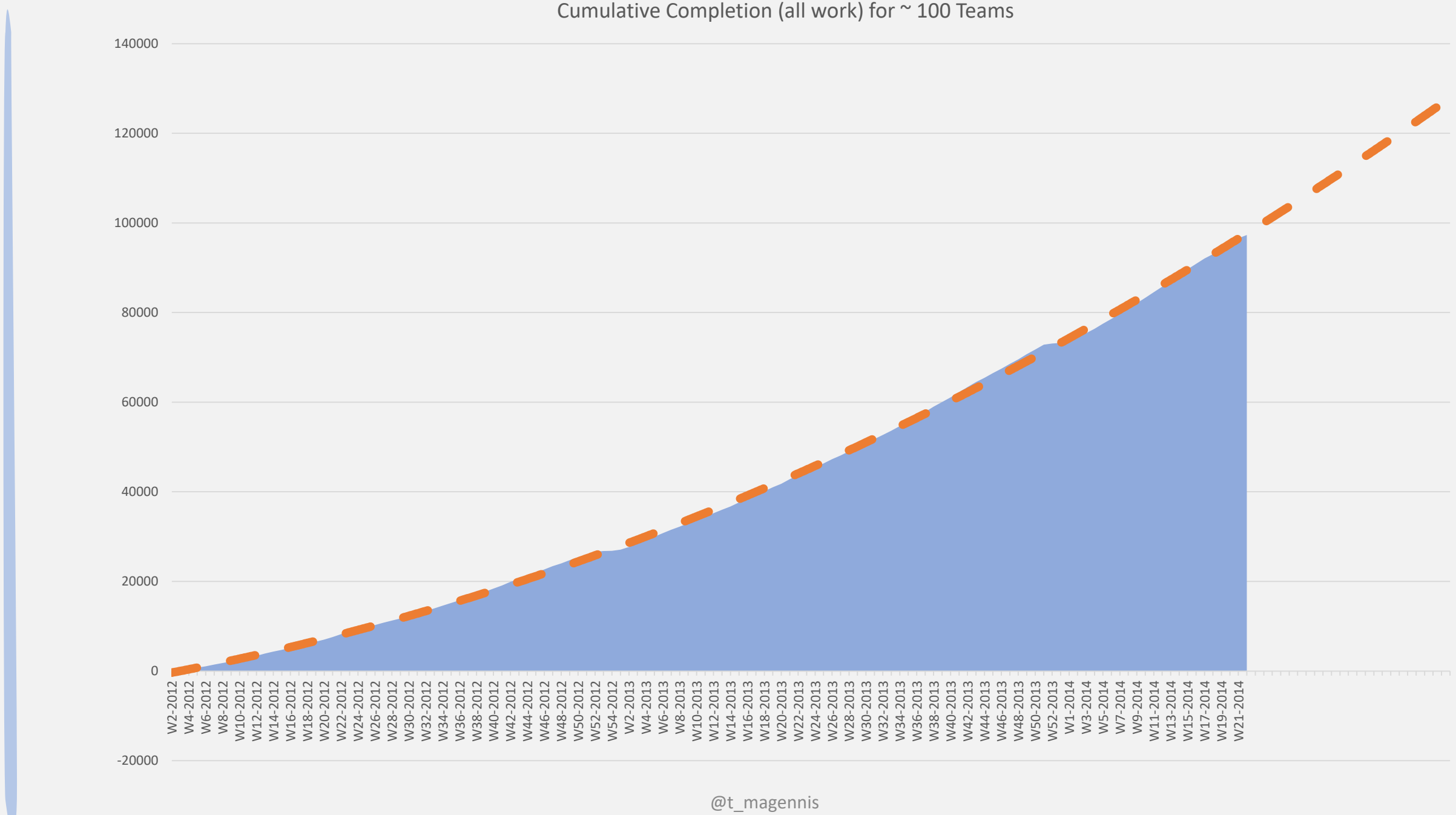
Back-testing: Using the data you have, predict something known

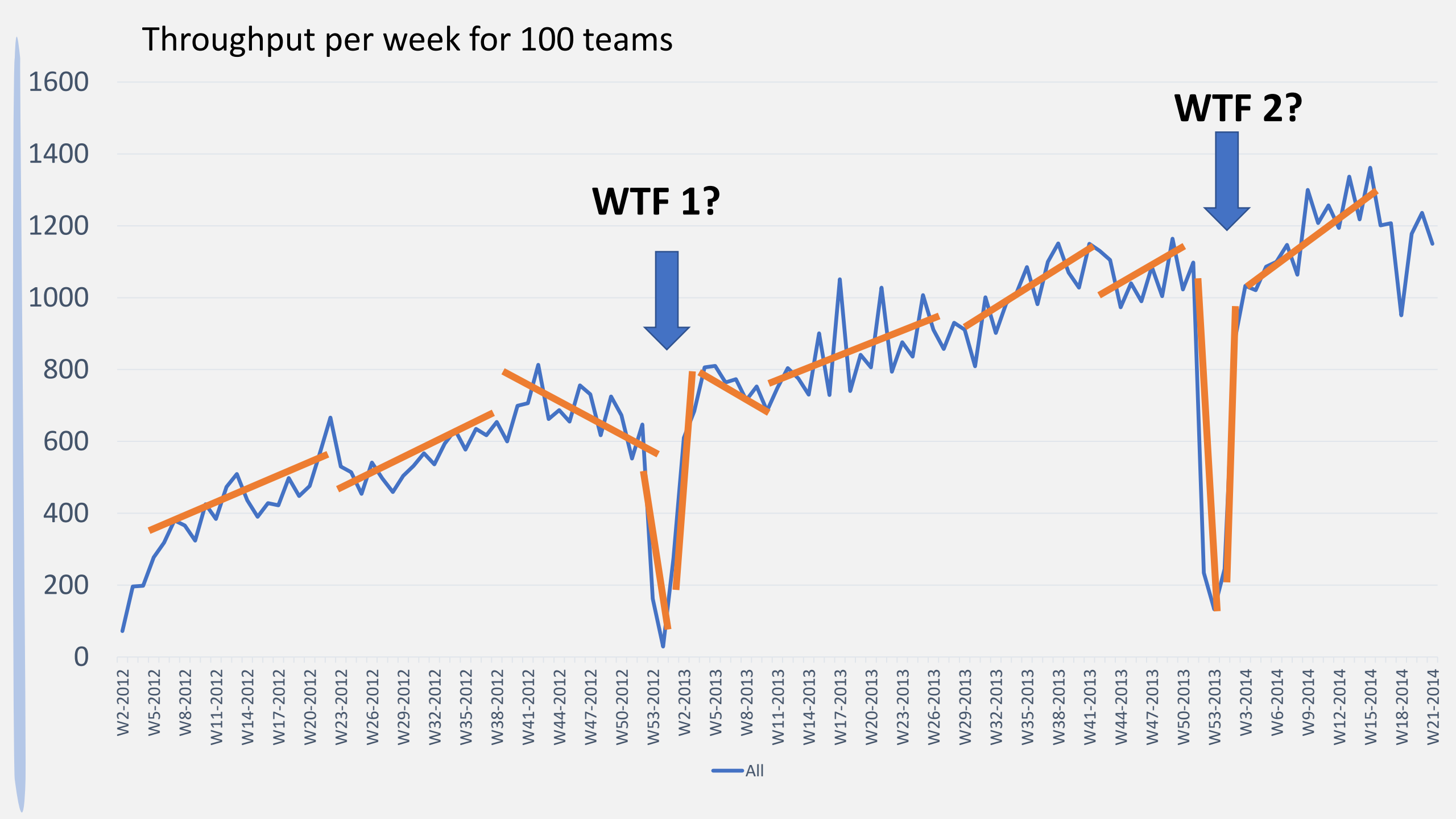

ARRIVAL AND DEPARTURE AVERAGE RATES
Completed
Work in Progress
Arrival Rate Avg.
Departure Rate Avg.
Linear (Arrival Rate Avg.)
Linear (Departure Rate Avg.)
Number of Items
200
180
160
140
120
100
80
60
40
20
0
1/14/2015
1/21/2015
1/28/2015
2/4/2015
2/11/2015
2/18/2015
2/25/2015
3/4/2015
3/11/2015
3/18/2015
3/25/2015
4/1/2015
4/8/2015
4/15/2015
4/22/2015
4/29/2015
5/6/2015
5/13/2015
5/20/2015
5/27/2015
6/3/2015
6/10/2015
ONLY believe THIS
IF you see THIS

Simple Regression Line Forecasting

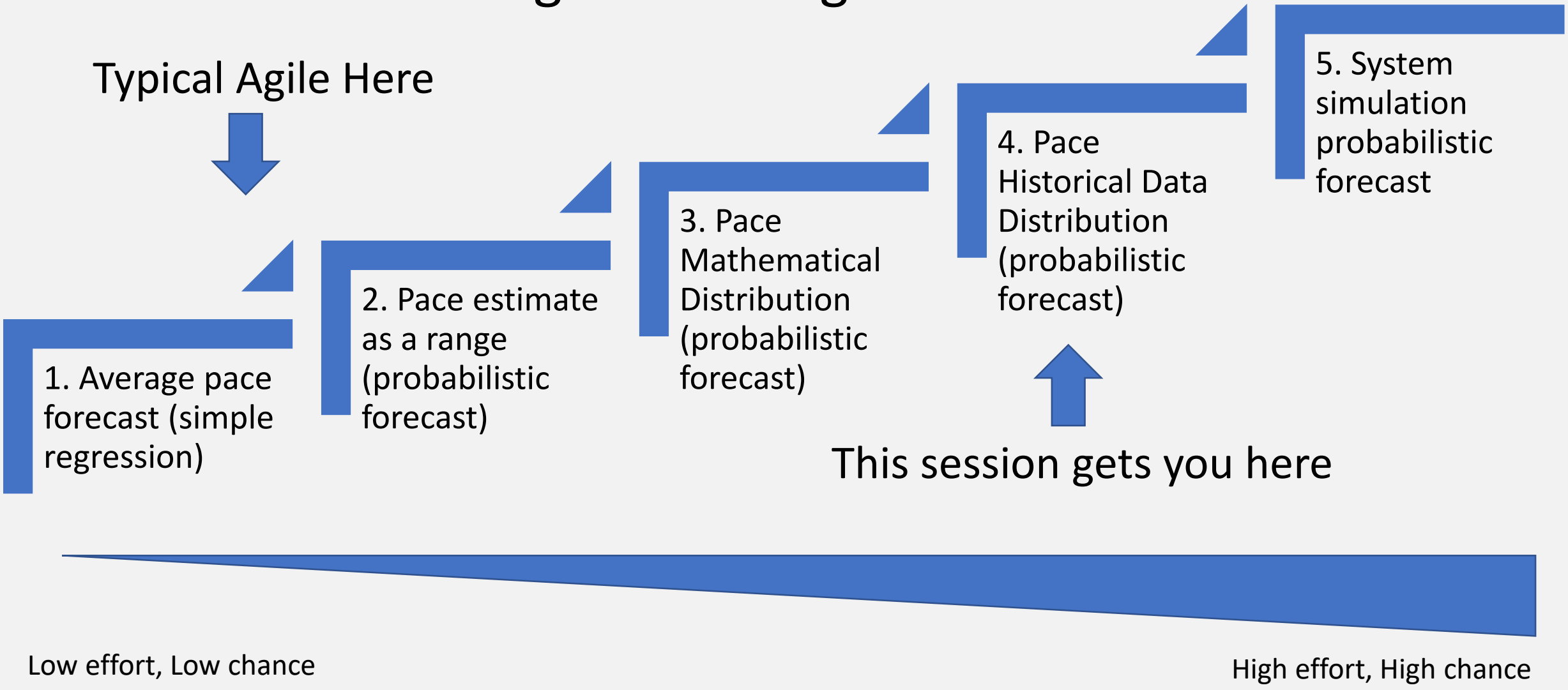
- Use the most recent 5-11 samples
  - Less than that, too little data to make sense
  - More than that, too exposed to context changes
- Test backwards, before forecasting forward
  - Remove recent 1/3 data and see if first 2/3 would have predicted it
- Understand the limitations in your context
  - Organizational disruption > local variability in many cases
  - It is the simplest model that MAY work


Cumulative Completion (all work) for ~ 100 Teams
120000
100000
80000
60000
40000
20000
0
W2-2012
W4-2012
W6-2012
W8-2012
W10-2012
W12-2012
W14-2012
W16-2012
W18-2012
W20-2012
W22-2012
W24-2012
W26-2012
W28-2012
W30-2012
W32-2012
W34-2012
W36-2012
W38-2012
W40-2012
W42-2012
W44-2012
W46-2012
W48-2012
W50-2012
W52-2012
W54-2012
W2-2013
W4-2013
W6-2013
W8-2013
W10-2013
W12-2013
W14-2013
W16-2013
W18-2013
W20-2013
W22-2013
W24-2013
W26-2013
W28-2013
W30-2013
W32-2013
W34-2013
W36-2013
W38-2013
W40-2013
W42-2013
W44-2013
W46-2013
W48-2013
W50-2013
W52-2013
W1-2014
W3-2014
W5-2014
W7-2014
W9-2014
W11-2014
W13-2014
W15-2014
W17-2014
W19-2014
W21-2014
@t_magennis


Cumulative Completion (all work) for ~ 100 Teams
140000
120000
100000
80000
60000
40000
20000
0
-20000
W2-2012
W4-2012
W6-2012
W8-2012
W10-2012
W12-2012
W14-2012
W16-2012
W18-2012
W20-2012
W22-2012
W24-2012
W26-2012
W28-2012
W30-2012
W32-2012
W34-2012
W36-2012
W38-2012
W40-2012
W42-2012
W44-2012
W46-2012
W48-2012
W50-2012
W52-2012
W54-2012
W2-2013
W4-2013
W6-2013
W8-2013
W10-2013
W12-2013
W14-2013
W16-2013
W18-2013
W20-2013
W22-2013
W24-2013
W26-2013
W28-2013
W30-2013
W32-2013
W34-2013
W36-2013
W38-2013
W40-2013
W42-2013
W44-2013
W46-2013
W48-2013
W50-2013
W52-2013
W1-2014
W3-2014
W5-2014
W7-2014
W9-2014
W11-2014
W13-2014
W15-2014
W17-2014
W19-2014
W21-2014
@t_magennis


Throughput per week for 100 teams
WTF 1?
WTF 2?
1600
1400
1200
1000
800
600
400
200
0
W2-2012
W5-2012
W8-2012
W11-2012
W14-2012
W17-2012
W20-2012
W23-2012
W26-2012
W29-2012
W32-2012
W35-2012
W38-2012
W41-2012
W44-2012
W47-2012
W50-2012
W53-2012
W2-2013
W5-2013
W8-2013
W11-2013
W14-2013
W17-2013
W20-2013
W23-2013
W26-2013
W29-2013
W32-2013
W35-2013
W38-2013
W41-2013
W44-2013
W47-2013
W50-2013
W53-2013
W3-2014
W6-2014
W9-2014
W12-2014
W15-2014
W18-2014
W21-2014
All

Forecasting “How Long” Models



Low effort, Low chance

High effort, High chance

Effort AND how likely model represents actual outcome

Q: How Long?

Forecasting duration if nothing else was done…

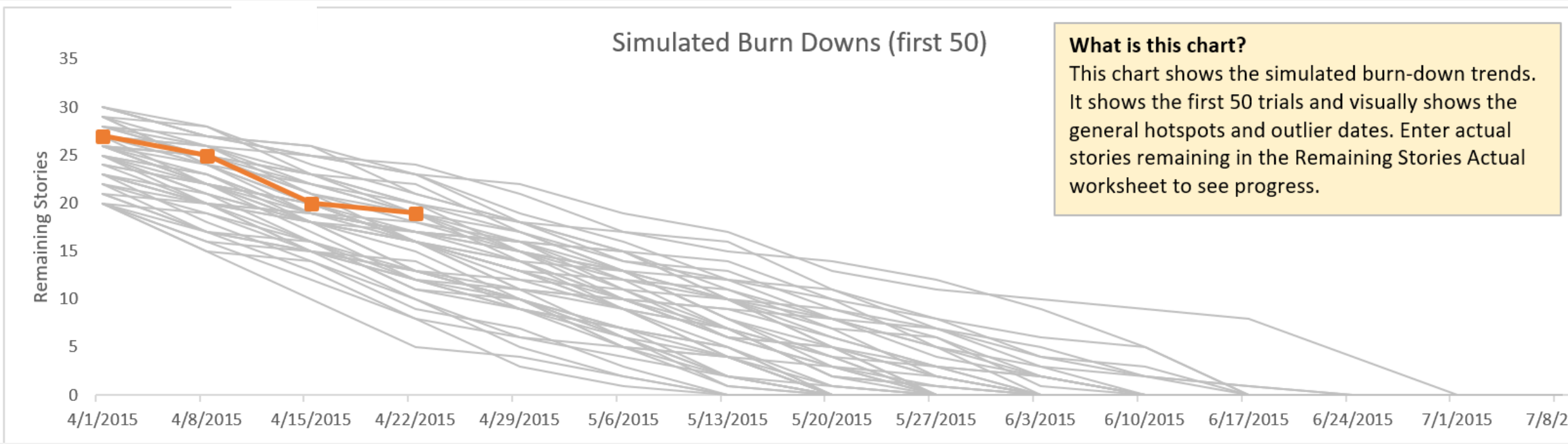
Forecasting Duration (and delivery date)

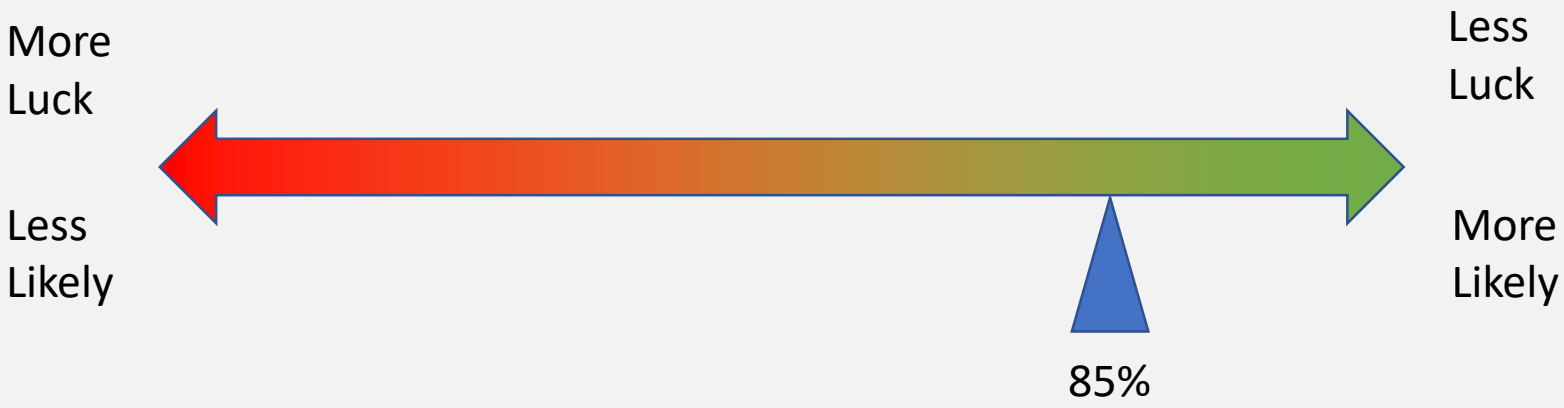
- Question: How can I estimate the amount of time it will take to deliver a feature or project?

- Theory: Using a range estimate or actual team delivery rate data, calculate how many of those periods of time to complete delivery

http://bit.ly/ThroughputForecast

Estimate or Sampling based Monte Carlo
duration and date forecasting Excel spreadsheet


Simulated Burn Downs (first 50)
Remaining Stories
35
30
25
20
15
10
5
0
4/1/2015
4/8/2015
4/15/2015
4/22/2015
4/29/2015
5/6/2015
5/13/2015
5/20/2015
5/27/2015
6/3/2015
6/10/2015
6/17/2015
6/24/2015
7/1/2015
7/8/2
What is this chart?
This chart shows the simulated burn-down trends. It shows the first 50 trials and visually shows the general hotspots and outlier dates. Enter actual stories remaining in the Remaining Stories Actual worksheet to see progress.


More Luck
Less Likely
Less Luck
More Likely
85%

Forecast Completion Date

1. Start Date 4/1/2015

2. How many stories are remaining to be completed?

(enter the range estimate of stories. Tip: start wide and narrow as certainty increases)

Low guess 20 Highest guess 30

3. Stories are often split before and whilst being worked on. Estimate the split rate low and high bounds.

(often the throughput in the backlog is pre-split, but captured throughput post-split. Adjust for this here)

Low guess 1.00 Highest guess 1.00

4. Throughput. How many completed stories per week or sprint do you estimate low and high bounds?

Throughput estimate/samples are per Week 7 days

Use historical throughput data **OR** enter a low and high estimate below. Use: Estimate

Low guess 1 Highest guess 5

Can I use velocity rather than throughput?
Yes. If you do have estimates in story points, then you can sume all of the estimates and use that for input 2 and estimate or use historical team velocity for input 4. The benefit of using throughput (count of completed stories per week/sprint) is that the individual stories don't require estimation in story points.

@t_magennis

Results

Likelihood	Duration in Week's	Date
100%	14	7/8/2015
95%	12	6/24/2015
90%	11	6/17/2015
85%	11	6/17/2015
80%	10	6/10/2015
75%	10	6/10/2015
70%	10	6/10/2015
65%	9	6/3/2015
60%	9	6/3/2015
55%	9	6/3/2015
50%	9	6/3/2015
45%	8	5/27/2015
40%	8	5/27/2015
35%	8	5/27/2015
30%	8	5/27/2015
25%	7	5/20/2015
20%	7	5/20/2015
15%	7	5/20/2015
10%	7	5/20/2015
5%	6	5/13/2015
0%	5	5/6/2015

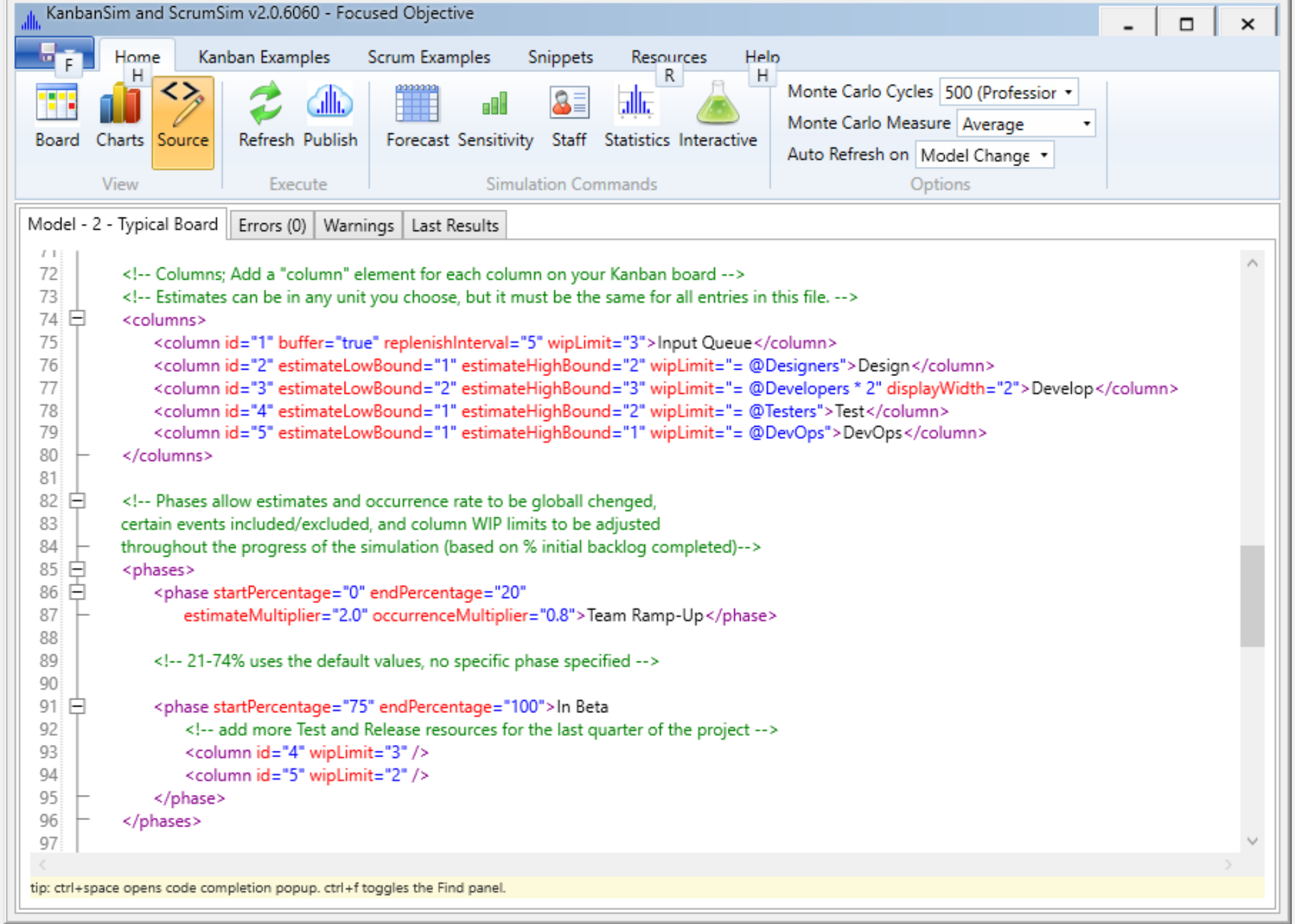
Almost certain

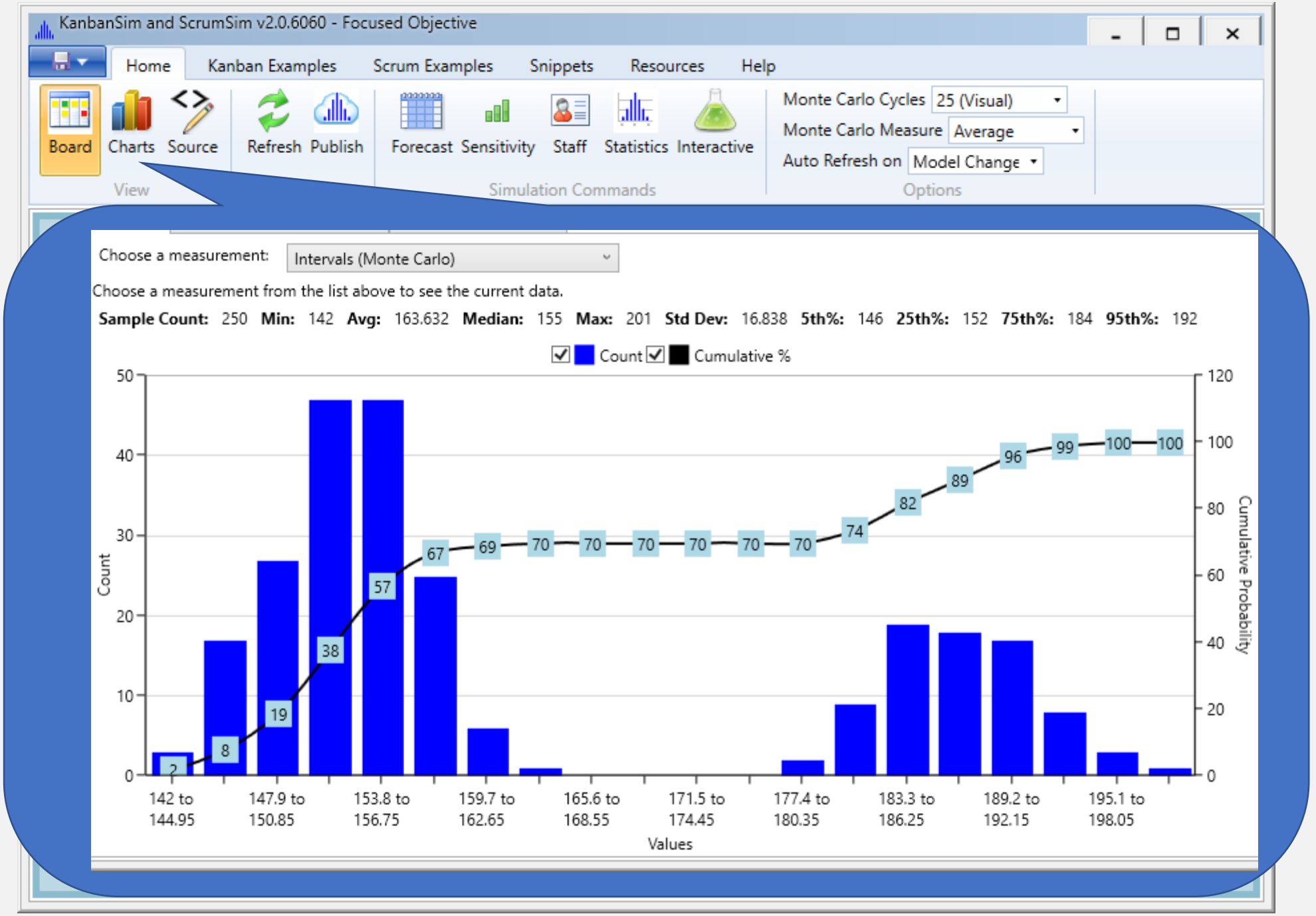
Somewhat certain

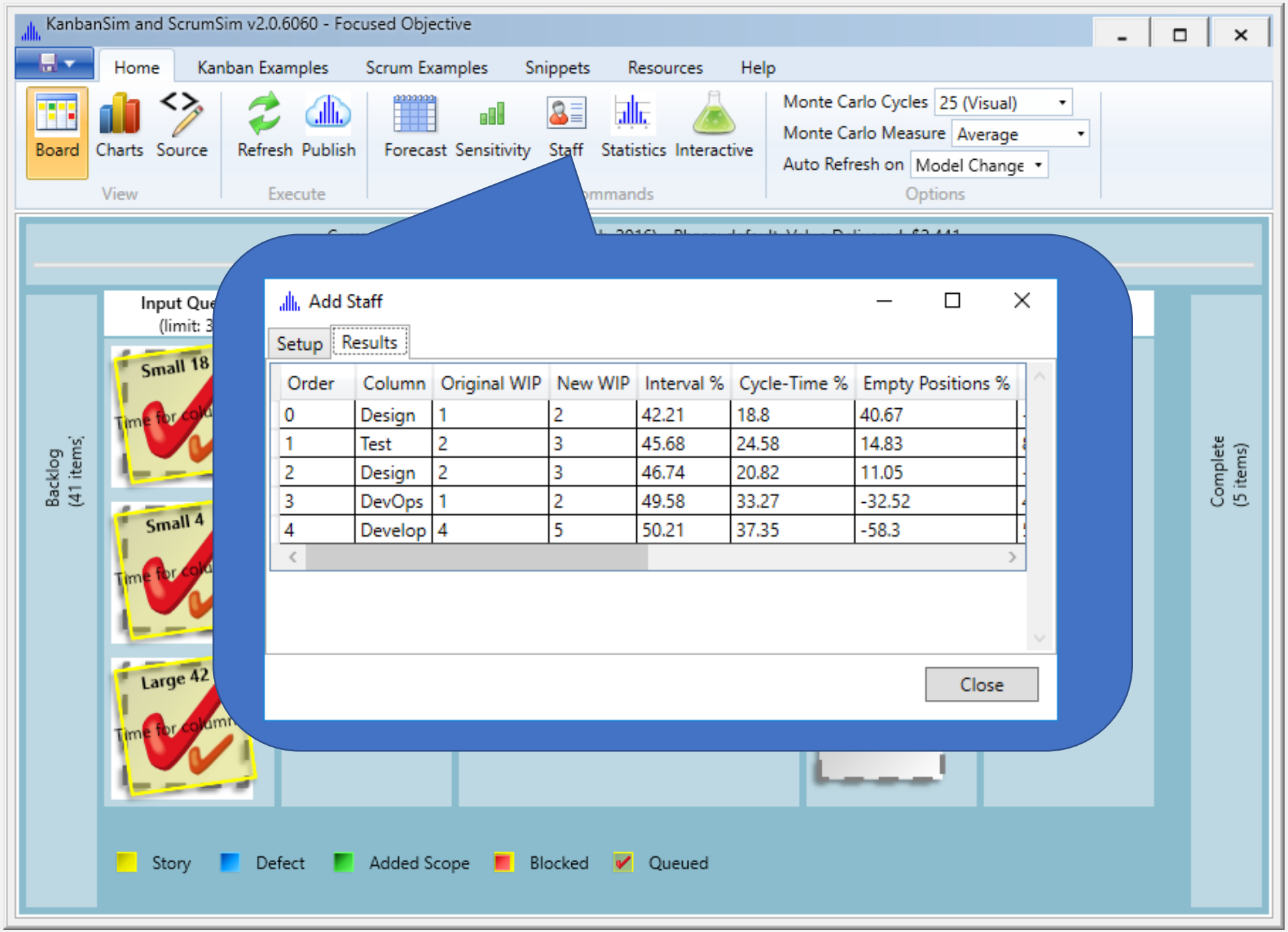
Less than coin-toss odds. But if you are game?

Q: Where is our constraint?

Understanding the process in detail


KanbanSim and ScrumSim v2.0.6060 - Focused Objective
F
Home
Kanban Examples
Scrum Examples
Snippets
Resources
Help
H
H
R
H
Board
Charts
Source
Refresh
Publish
Forecast
Sensitivity
Staff
Statistics
Interactive
View
Execute
Simulation Commands
Monte Carlo Cycles
500 (Professior
Monte Carlo Measure
Average
Auto Refresh on
Model Change
Options
Model - 2 - Typical Board
Errors (0)
Warnings
Last Results
<!-- Columns; Add a "column" element for each column on your Kanban board -->
<!-- Estimates can be in any unit you choose, but it must be the same for all entries in this file. -->
<columns>
<column id="1" buffer="true" replenishInterval="5" wipLimit="3">Input Queue</column>
<column id="2" estimateLowBound="1" estimateHighBound="2" wipLimit="= @Designers">Design</column>
<column id="3" estimateLowBound="2" estimateHighBound="3" wipLimit="= @Developers * 2" displayWidth="2">Develop</column>
<column id="4" estimateLowBound="1" estimateHighBound="2" wipLimit="= @Testers">Test</column>
<column id="5" estimateLowBound="1" estimateHighBound="1" wipLimit="= @DevOps">DevOps</column>
</columns>
<!-- Phases allow estimates and occurrence rate to be globall chenged,
certain events included/excluded, and column WIP limits to be adjusted
throughout the progress of the simulation (based on % initial backlog completed)-->
<phases>
<phase startPercentage="0" endPercentage="20"
estimateMultiplier="2.0" occurrenceMultiplier="0.8">Team Ramp-Up</phase>
<!-- 21-74% uses the default values, no specific phase specified -->
<phase startPercentage="75" endPercentage="100">In Beta
<!-- add more Test and Release resources for the last quarter of the project -->
<column id="4" wipLimit="3" />
<column id="5" wipLimit="2" />
</phase>
</phases>
tip: ctrl+space opens code completion popup. ctrl+f toggles the Find panel.


KanbanSim and ScrumSim v2.0.6060 - Focused Objective
Home
Kanban Examples
Scrum Examples
Snippets
Resources
Help
Board
Charts
Source
Refresh
Publish
Forecast
Sensitivity
Staff
Statistics
Interactive
View
Simulation Commands
Monte Carlo Cycles 25 (Visual)
Monte Carlo Measure Average
Auto Refresh on Model Change
Options
Choose a measurement: Intervals (Monte Carlo)
Choose a measurement from the list above to see the current data.
Sample Count: 250 Min: 142 Avg: 163.632 Median: 155 Max: 201 Std Dev: 16.838 5th%: 146 25th%: 152 75th%: 184 95th%: 192
Count
Cumulative %
Count
Cumulative Probability
Values
142 to 144.95
147.9 to 150.85
153.8 to 156.75
159.7 to 162.65
165.6 to 168.55
171.5 to 174.45
177.4 to 180.35
183.3 to 186.25
189.2 to 192.15
195.1 to 198.05


KanbanSim and ScrumSim v2.0.6060 - Focused Objective
Home
Kanban Examples
Scrum Examples
Snippets
Resources
Help
Board
Charts
Source
Refresh
Publish
Forecast
Sensitivity
Staff
Statistics
Interactive
View
Execute
Monte Carlo Cycles
25 (Visual)
Monte Carlo Measure
Average
Auto Refresh on
Model Change
Options
Add Staff
Setup
Results
Order
Column
Original WIP
New WIP
Interval %
Cycle-Time %
Empty Positions %
0 Design 1 2 42.21 18.8 40.67
1 Test 2 3 45.68 24.58 14.83
2 Design 2 3 46.74 20.82 11.05
3 DevOps 1 2 49.58 33.27 -32.52
4 Develop 4 5 50.21 37.35 -58.3
Close
Backlog
(41 items)
Small 18
Small 4
Large 42
Complete
(5 items)
Story
Defect
Added Scope
Blocked
Queued

Key Takeaways

- Start collecting started, finished and type of work data
- Start using data during team decision meetings
- You need much less data that you think to be better than “intuition”
- Use the values of a “few” to forecast the “many” – 7 to 11 samples
- Beware the limitation of simple regression forecasting
- Start using probabilistic forecasting tools
  - Mine are free: Bit.ly/SimResources

Get everything here: Slides and tools:

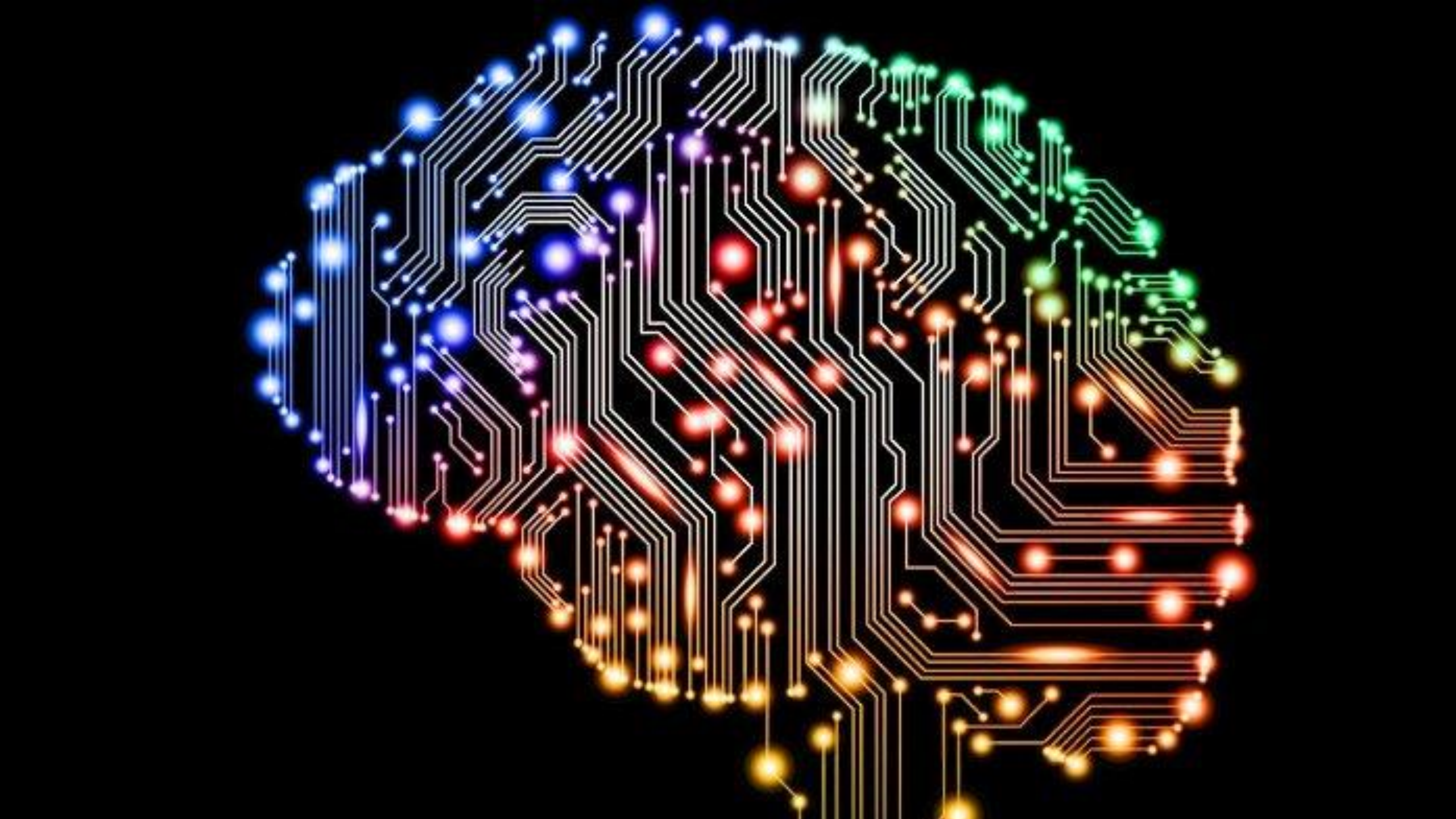
Bit.ly/SimResources

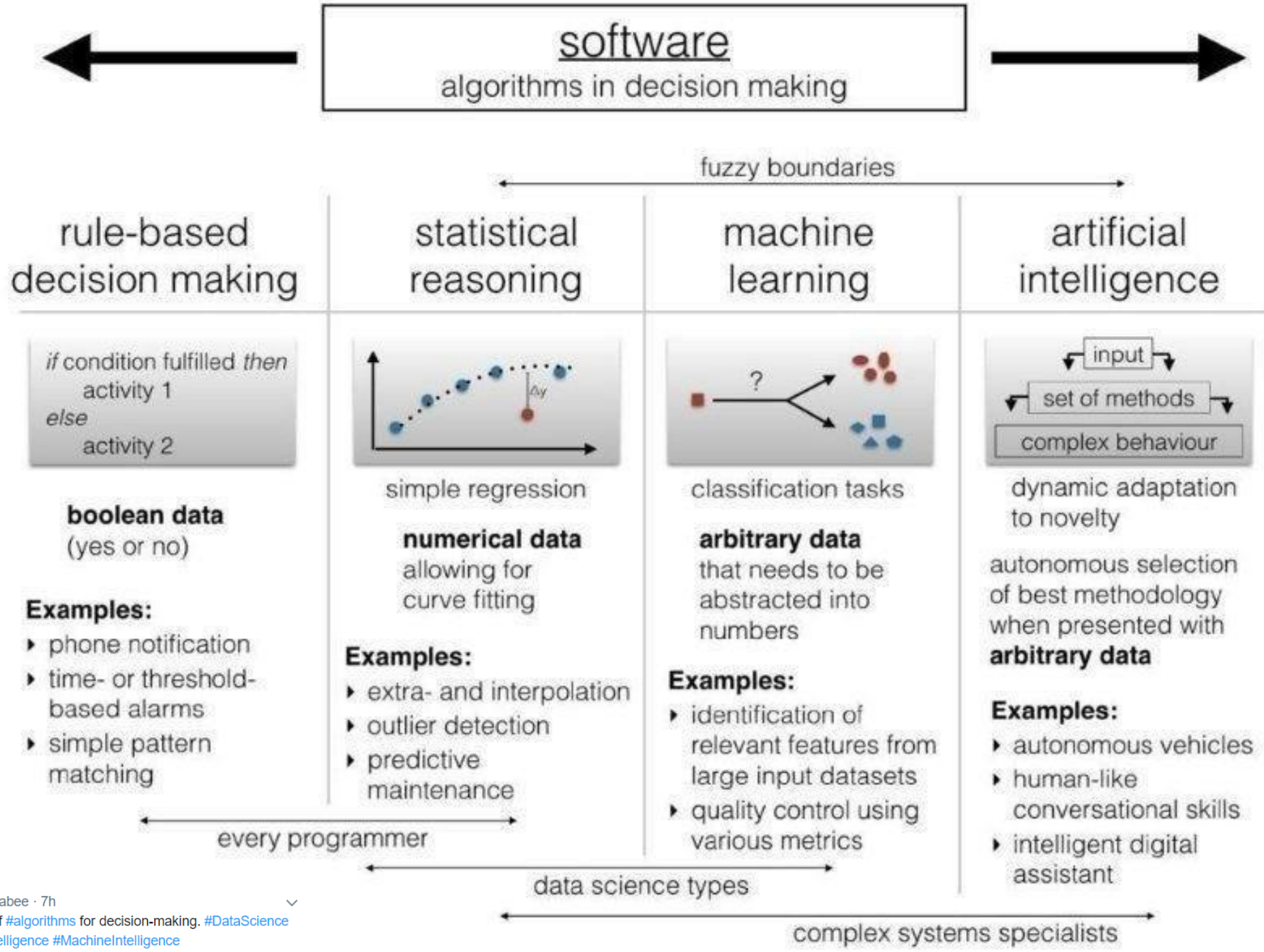
Me on Twitter

@t_magennis

About me…

- What I do
  - Teach how to use data for forecasting
  - Teach simple math to executives, especially “demand > supply”
  - Teach how to know (earlier) that you are on the wrong side of an expectation
- What I did
  - Started in software 1986. I actually liked Assembler & Cobol
  - Have worked at senior exec level, and now beside them for major corporations so I have some insight into what passes their decision filters
- How to reach me
  - Twitter: @t_magennis or email: troy.magennis@focusedobjective.com
  - Lots of free spreadsheets and stuff at FocusedObjective.com





source

You Retweeted

Hayim Makabee @hayim_makabee · 7h

Understanding different types of #algorithms for decision-making. #DataScience #MachineLearning #ArtificialIntelligence #MachineIntelligence

Training Set

Holdout Set

	A	B	C	D	E	F	G	H	I	J
1	**survived**	**pclass**	**name**	**sex**	**age**	**sibsp**	**parch**	**ticket**	**fare**	**cabin**
2	Survived	1	Allen, Miss. Elisabeth Walton	female	29	0	0	24160	211.3375	B5
3	Survived	1	Allison, Master. Hudson Trevor	male	0.917	1	2	113781	151.5500	C22 C26
4	Perished	1	Allison, Miss. Helen Loraine	female	2	1	2	113781	151.5500	C22 C26
5	Perished	1	Allison, Mr. Hudson Joshua Creighton	male	30	1	2	113781	151.5500	C22 C26
6	Perished	1	Allison, Mrs. Hudson J C (Bessie Waldo Daniels)	female	25	1	2	113781	151.5500	C22 C26
7	Survived	1	Anderson, Mr. Harry	male	48	0	0	19952	26.5500	E12
8	Survived	1	Andrews, Miss. Kornelia Theodosia	female	63	1	0	13502	77.9583	D7
9	Perished	1	Andrews, Mr. Thomas Jr	male	39	0	0	112050	0.0000	A36
10	Survived	1	Appleton, Mrs. Edward Dale (Charlotte Lamson)	female	53	2	0	11769	51.4792	C101
11		1	Artagaveytia, Mr. Ramon	male	71	0	0	PC 17609	49.5042	
12		1	Astor, Col. John Jacob	male	47	1	0	PC 17757	227.5250	C62 C64
13		1	Astor, Mrs. John Jacob (Madeleine Talmadge Force)	female	18	1	0	PC 17757	227.5250	C62 C64
14		1	Aubart, Mme. Leontine Pauline	female	24	0	0	PC 17477	69.3000	B35
15		1	Barber, Miss. Ellen "Nellie"	female	26	0	0	19877	78.8500	
16		1	Barkworth, Mr. Algernon Henry Wilson	male	80	0	0	27042	30.0000	A23
17		1	Baumann, Mr. John D	male		0	0	PC 17318	25.9250	
18		1	Baxter, Mr. Quigg Edmond	male	24	0	1	PC 17558	247.5208	B58 B60
19		1	Baxter, Mrs. James (Helene DeLaudeniere Chaput)	female	50	0	1	PC 17558	247.5208	B58 B60
20		1	Bazzani, Miss. Albina	female	32	0	0	11813	76.2917	D15

We pretend we don’t know something we do know. We predict and compare.

Simple Decision Tree

Passenger

Sex

Male

Female

73% survived in group
36% of the passengers

Guess: Survive

Age

Younger than
9.5 years

Older than
9.5 years

17% survived in group
61% of the passengers

Guess: Perish

Family Size

Family Group Size
< 2.5 people

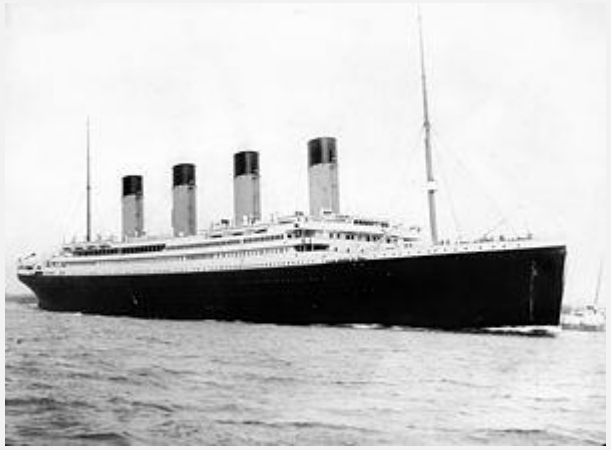
Family Group Size
> 2.5 people

5% survived in group
2% of the passengers

Guess: Perish

89% survived in group
2% of the passengers

Guess: Survive



What if you could predict…

- What test cases NEED to be run
- What test cases are consistently FALSE positives
- What code areas when touched causes the most future defect reports
- How risky is releasing a hotfix for an area of code

Learning for Test Prioritization: An Industrial Case Study

Benjamin Busjaeger
Salesforce.com
San Francisco, CA 94105
bbusjaeger@salesforce.com

Tao Xie
University of Illinois at Urbana-Champaign
Urbana, IL 61801
taoxie@illinois.edu

http://taoxie.cs.illinois.edu/publications/fse16industry-learning.pdf

ABSTRACT

Modern cloud-software providers, such as Salesforce.com, increasingly adopt large-scale continuous integration environments. In such environments, assuring high developer productivity is strongly dependent on conducting testing efficiently and effectively. Specifically, to shorten feedback cycles, test prioritization is popularly used as an optimization mechanism for ranking tests to run by their likelihood of revealing failures. To apply test prioritization in industrial environments, we present a novel approach (tailored for practical applicability) that integrates multiple existing techniques via a systematic framework of machine learning to rank. Our initial empirical evaluation on a large real-world dataset from Salesforce.com shows that our approach

can also be used for test selection in resource-constrained environments by running the top-k ranked tests [6].

Test optimization mechanisms are particularly important for modern cloud-software companies, such as Salesforce.com, given the high change frequency and massive scale of the companies' code bases. Engineers typically continuously integrate their work into the mainline from which releases are deployed directly into production [5, 16]. Code health of the main branch is vital and small improvements in efficiency yield high gains in productivity. In particular, test prioritization can help in two main ways. First, test prioritization can be applied before submitting code to run a subset of tests most likely to fail. Doing so gives engineers fast feedback before switching context and can prevent major regressions

CRANE: Failure Prediction, Change Analysis and Test Prioritization in Practice - Experiences from Windows

Jacek Czerwonka[1], Rajiv Das[1], Nachiappan Nagappan[2], Alex Tarvo[1], Alex Teterev[1]
[1] *Windows Sustained Engineering, Core Operating Systems Division, Microsoft Corporation*
[2] *Microsoft Research*
{ jacekcz, rajivdas, nachin, alexta, alextet } @ microsoft.com

http://cs.brown.edu/~alexta/Doc/pubs/ICST2011_CRANE.pdf

Abstract **- Building large software systems is difficult. Maintaining large systems is equally hard. Making post-release changes requires not only thorough understanding of the architecture of a software component about to be changed but also its dependencies and interactions with other components in the system. Testing such changes in reasonable time and at a reasonable cost is a difficult problem as infinitely many test cases can be executed for any modification. It is important to obtain a risk assessment of impact of such post-release change fixes. Further, testing of such changes is complicated by the fact that they are applicable to hundreds of millions of users, even the smallest mistakes can translate to a very costly failure and re-work. There has been signifi-**

The amount of effort going into each of these categories varies depending on the nature of the software considered, its intended purpose, size and characteristics of its current user base. However, the software maintenance phase exhibits attributes that are common across different software products:

1. Software maintenance is expensive. It is generally accepted that the maintenance phase consumes the majority of resources required to take a software product throughout its lifecycle, from inception until end-of-life. The total cost of maintenance is estimated to comprise 50% or more of total life-cycle costs. [24]
2. Maintenance work is often done by people who had not created the system. Unless the effective lifetime of a

Top Three Forecasting Fail Reasons

Reasons you shouldn’t have hired me five years ago

Fail 1: Start Date On-Paper != Reality

- The assumed Start Date is often ONLY on paper

Start Date of Feature B is the finish date of Feature A

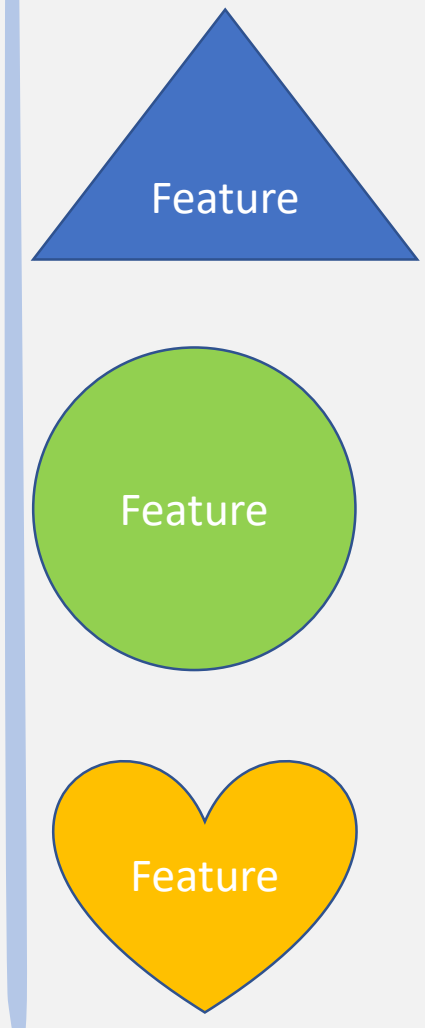
What is the team doing now?

- Define what start means
  - Team is dedicated and in-place
  - They are trained and know how to do their work
  - They know and understand what work they need to deliver
  - Nothing inhibits them doing or delivering that work

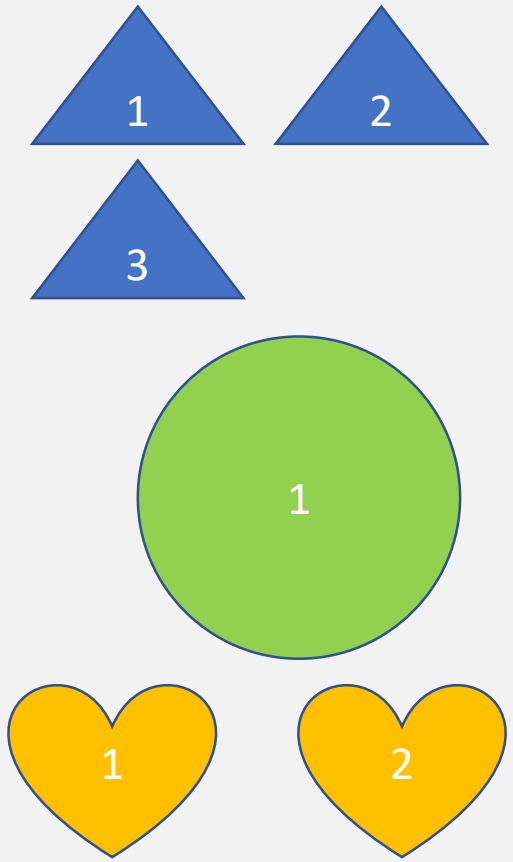
- Team is never fully available on day one!

Fail 2: Backlog Rate versus Delivery Rate

Features



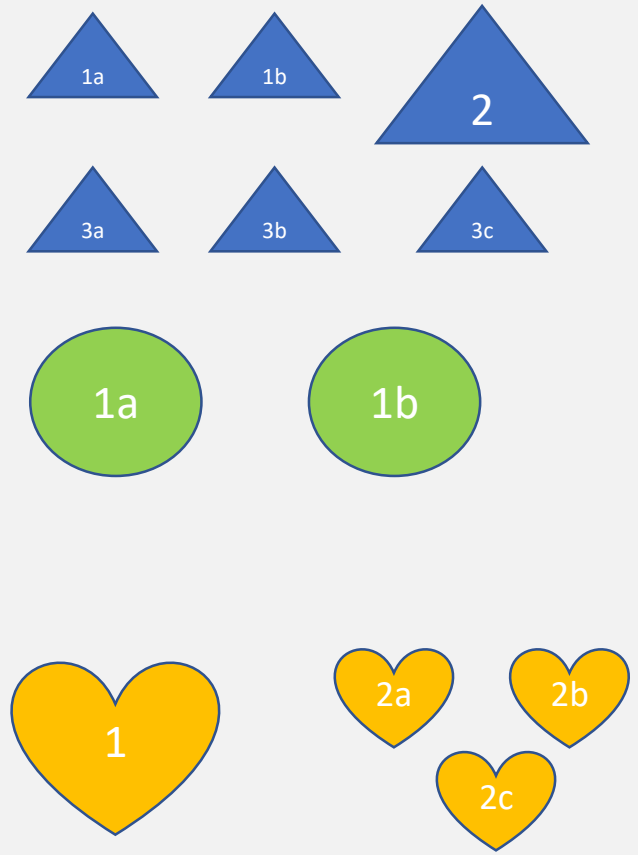
Estimated Stories



Actual Backlog

Rate Delivered = 6

Implemented Stories and Defects



Measured Throughput = 12

Fail 2: Backlog Rate versus Delivery Rate

- Forecast using the “Completion rate” we may under-forecast
  - Backlog is Miles per Hour, Completion rate is Kilometers per Hour
- Normal split rates are between 1 to 3 times (most common seen)
- This means
  - If you don’t account for it, you will UNDER-FORECAST by 1 to 3 times!

3. Stories are often split before and whilst being worked on. Estimate the split rate low and high bounds.

(often the throughput in the backlog is pre-split, but captured throughput post-split. Adjust for this here)

Low guess	1.00	Highest guess	3.00

Feature or Epic Name	Estimated # Stories or points (before starting)	Actual # Stoies or points (after completed)
Feature 1	5	8
Feature 2	3	4
Feature 3	8	10
Feature 4	4	
Feature 5	2	
Feature 6	7	
Feature 7		

5 X **1.6** = 8

3 X **1.3** = 4

8 X **1.25** = 10

actual growth rate range seen: **1.25** to **1.60** (enter actual count after work is completed in the input sheet)

ere.

1 to 3.

This split-rate adjust the rate that the team appears to be completing work with the rte original backlog items are started. This isn't only growth/creep, its adjusting for the way work splits when the team hs a closer look at it.

Fail 3: Ignoring Risks

- Risk = Work that “might” need to be done but we don’t know yet

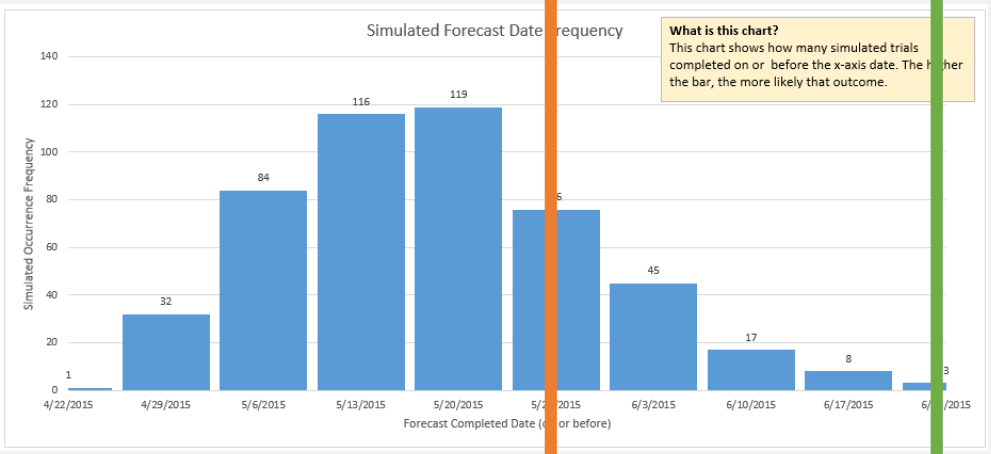
- Some samples
  - Fails on Internet Explorer 6, or now Safari on phones
  - Fails performance testing under load, or uses too much memory
  - CSS alignment issues with German text translations, things wrap
  - Production network security blocks traffic, awaiting vendor to fix
  - Fails on real customer data (we designed for 50 items, they have 500)

WITHOUT RISKS INCLUDED

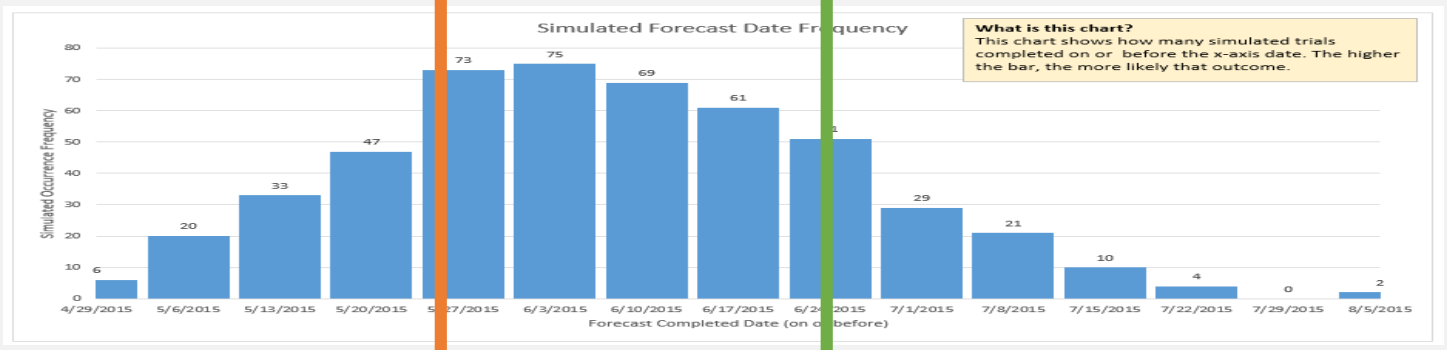
27th May (highest late June)

24th June (highest early August)

WITH RISKS INCLUDED

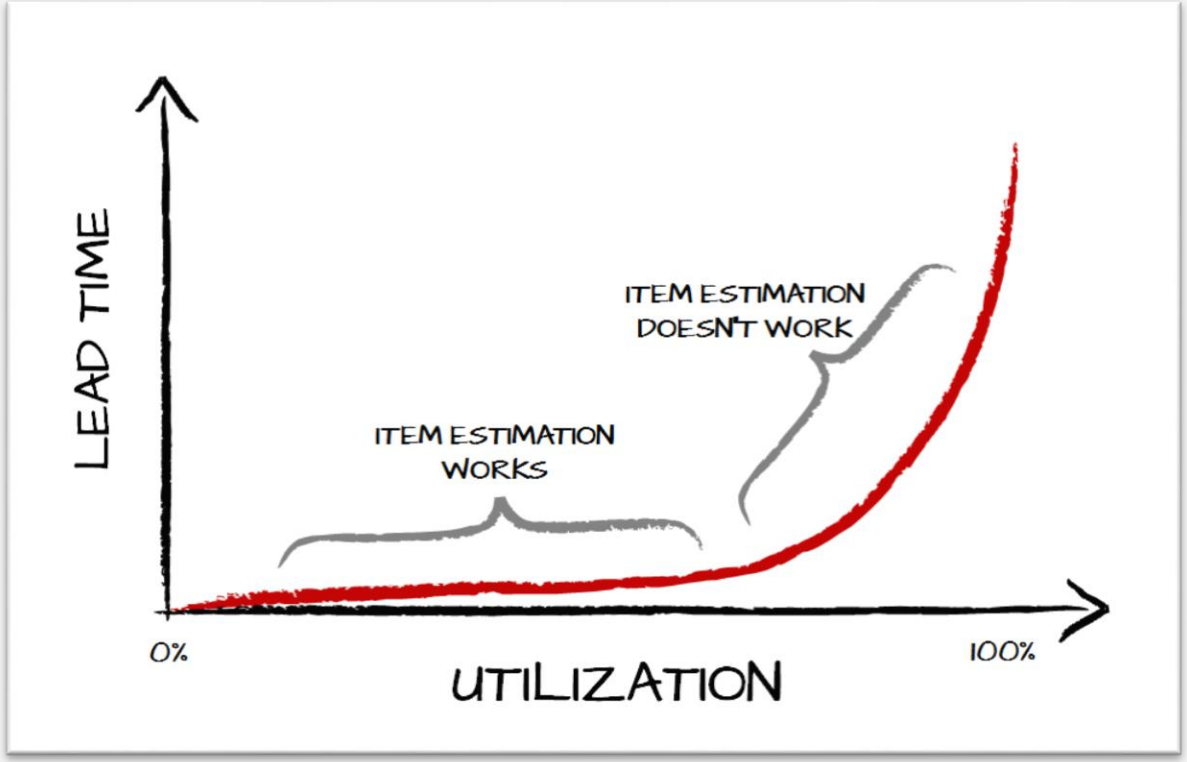


Likelihood	Impact Low	Impact High	Description
50%	7	10	Browser compatibility issues
40%	3	7	Performance under load
30%	5	10	Production configuration

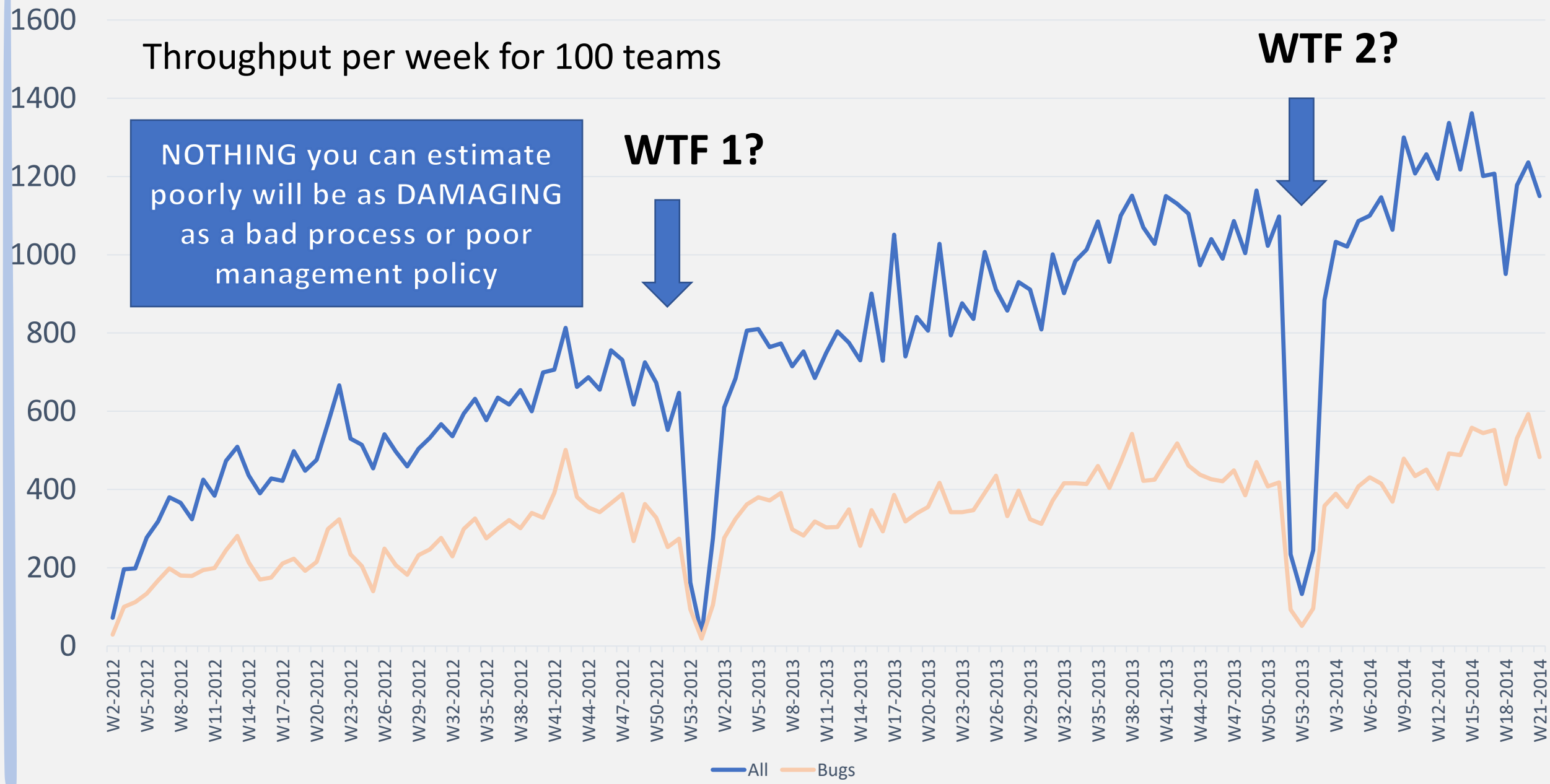


Forecasts shown at 85th Percentile

Bonus fail: High System Utilization



Can’t forecast high utilization systems using item size…


Throughput per week for 100 teams
NOTHING you can estimate poorly will be as DAMAGING as a bad process or poor management policy
WTF 1?
WTF 2?
1600
1400
1200
1000
800
600
400
200
0
W2-2012
W5-2012
W8-2012
W11-2012
W14-2012
W17-2012
W20-2012
W23-2012
W26-2012
W29-2012
W32-2012
W35-2012
W38-2012
W41-2012
W44-2012
W47-2012
W50-2012
W53-2012
W2-2013
W5-2013
W8-2013
W11-2013
W14-2013
W17-2013
W20-2013
W23-2013
W26-2013
W29-2013
W32-2013
W35-2013
W38-2013
W41-2013
W44-2013
W47-2013
W50-2013
W53-2013
W3-2014
W6-2014
W9-2014
W12-2014
W15-2014
W18-2014
W21-2014
All
Bugs

Key Take-aways and Resources

- Forecasting requires a system view,
- Three samples will outperform intuition (use most recent 7 samples)
- Give multiple options, not just one
- Forecast duration NOT date until “Start Conditions” are defined
- Track actual progress versus planned, and update the model continuously

- Get everything here: Slides and tools:

Bit.ly/SimResources