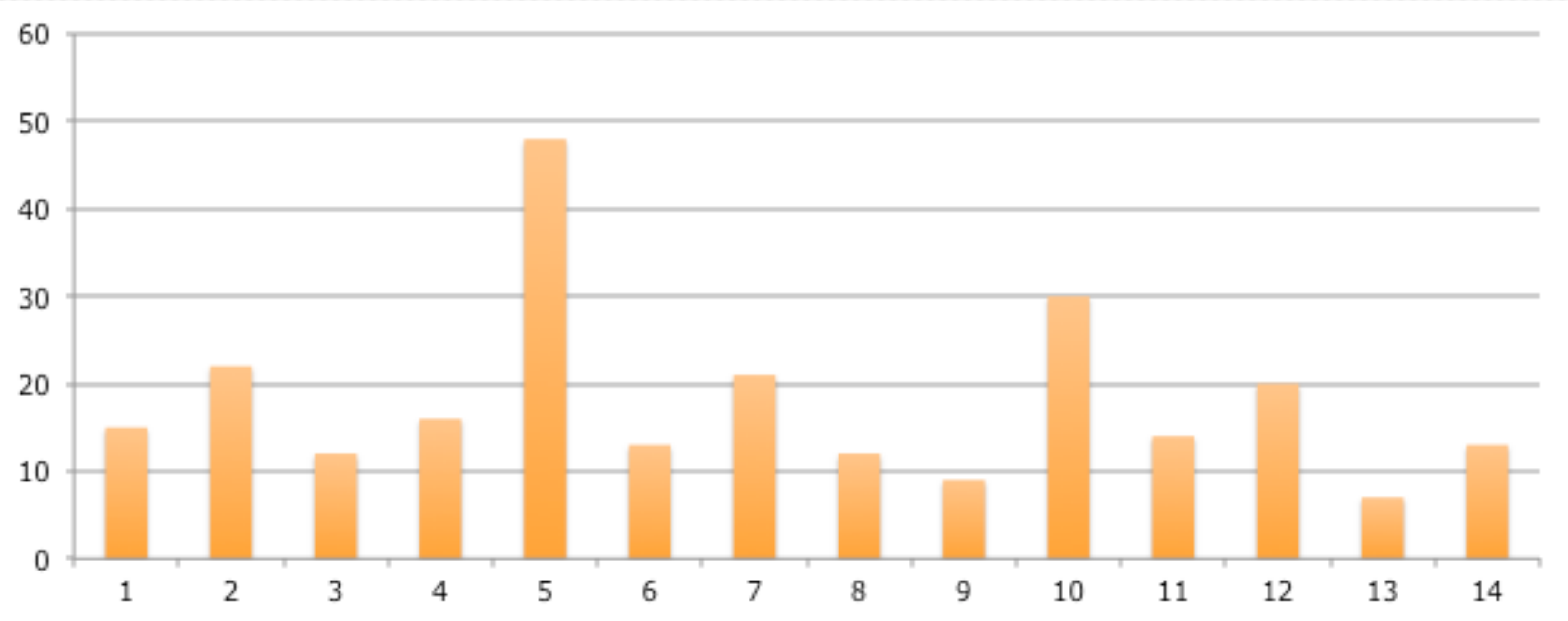
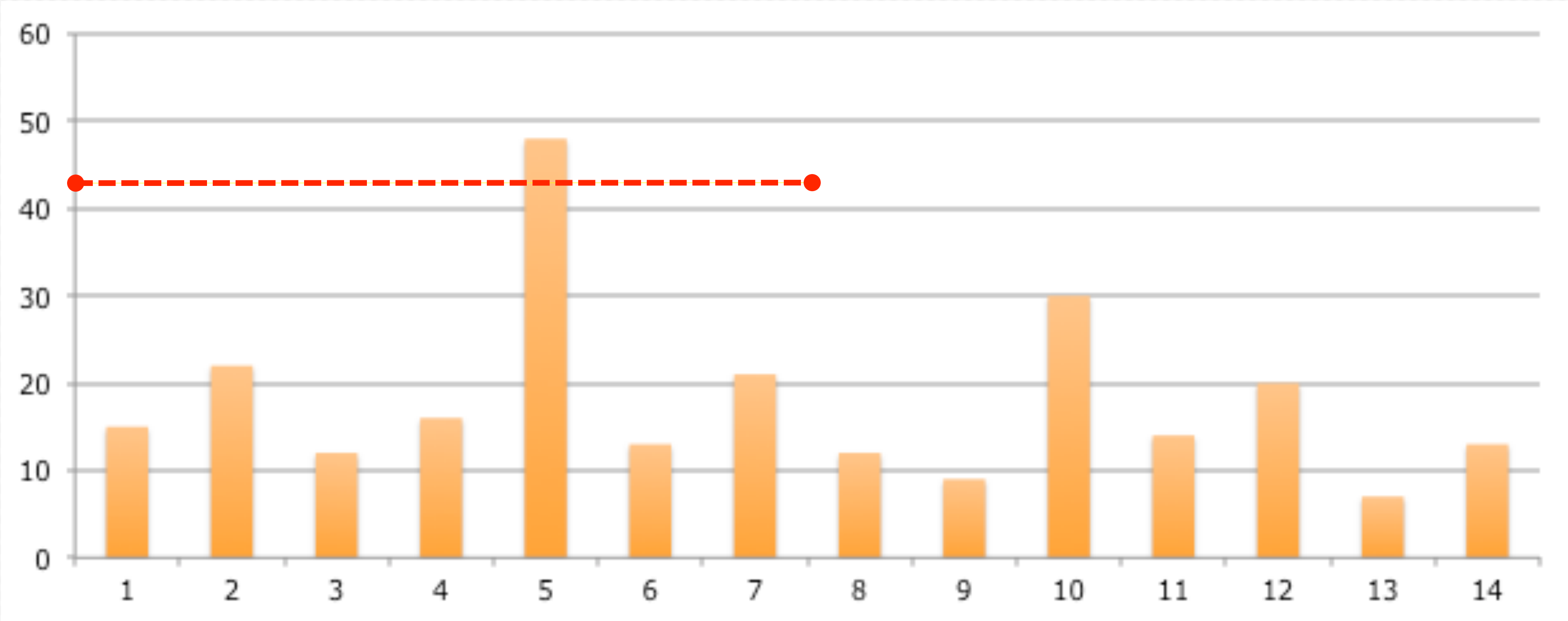
SmartChain

Low Frequency Orders
May 2016


60
50
40
30
20
10
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14

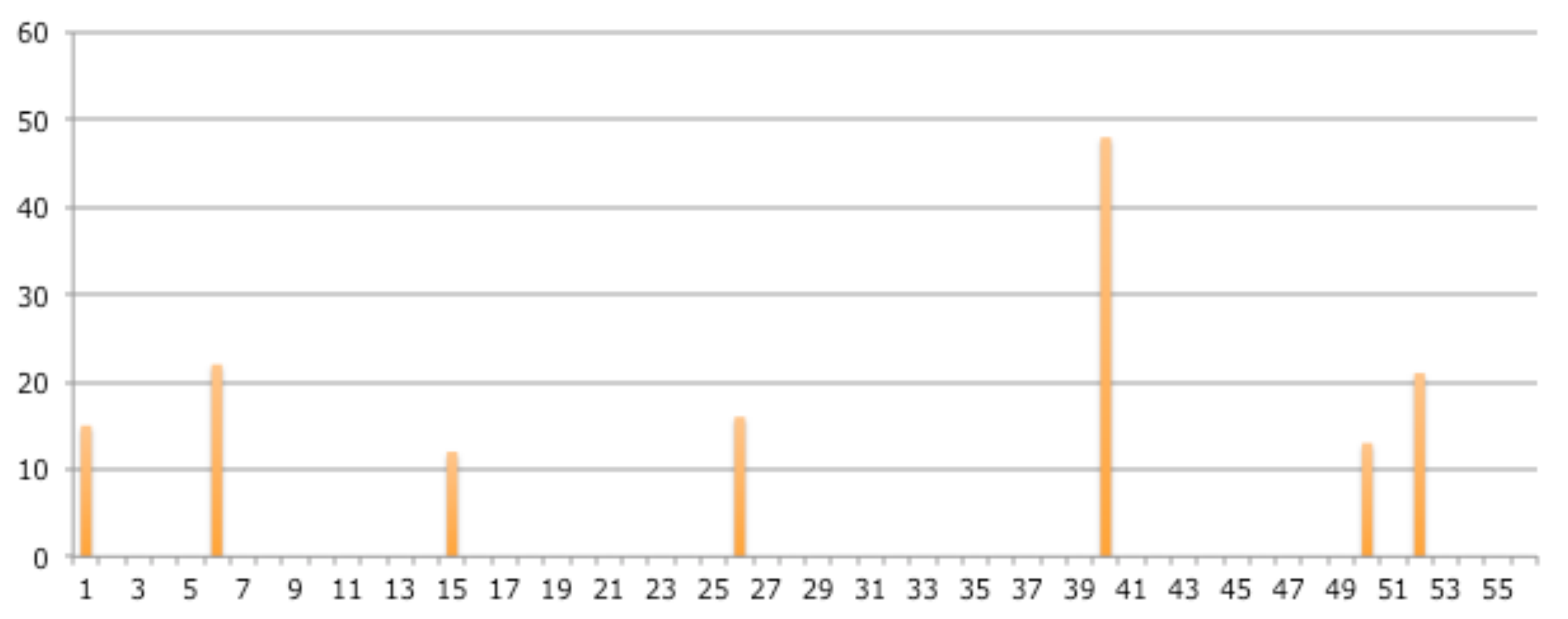


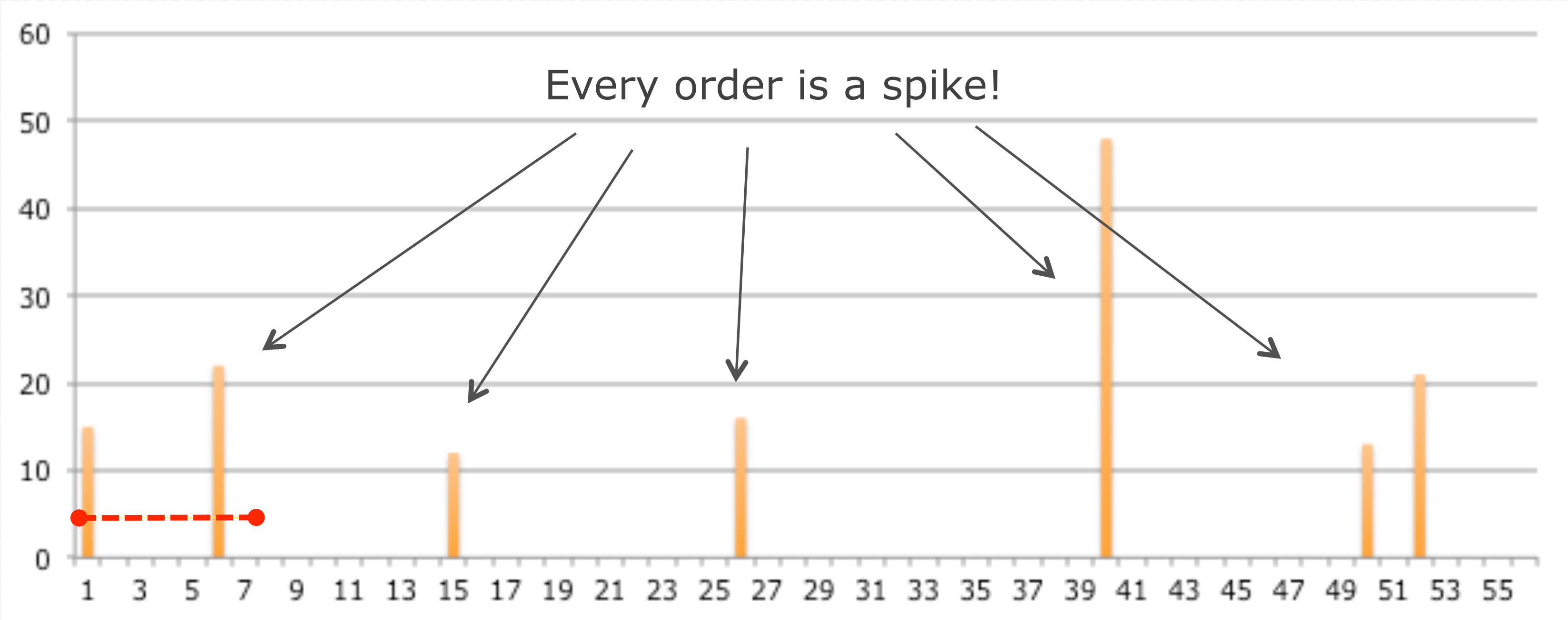
ADU = 18
DLT = 7 days
LT factor = 0.5
Var factor = 0.33

Red Zone = 84 Units

18 x 7 x 0.5 x (1+0.33)

Spike Threshold = 42 Units


60
50
40
30
20
10
0
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31 33 35 37 39 41 43 45 47 49 51 53 55



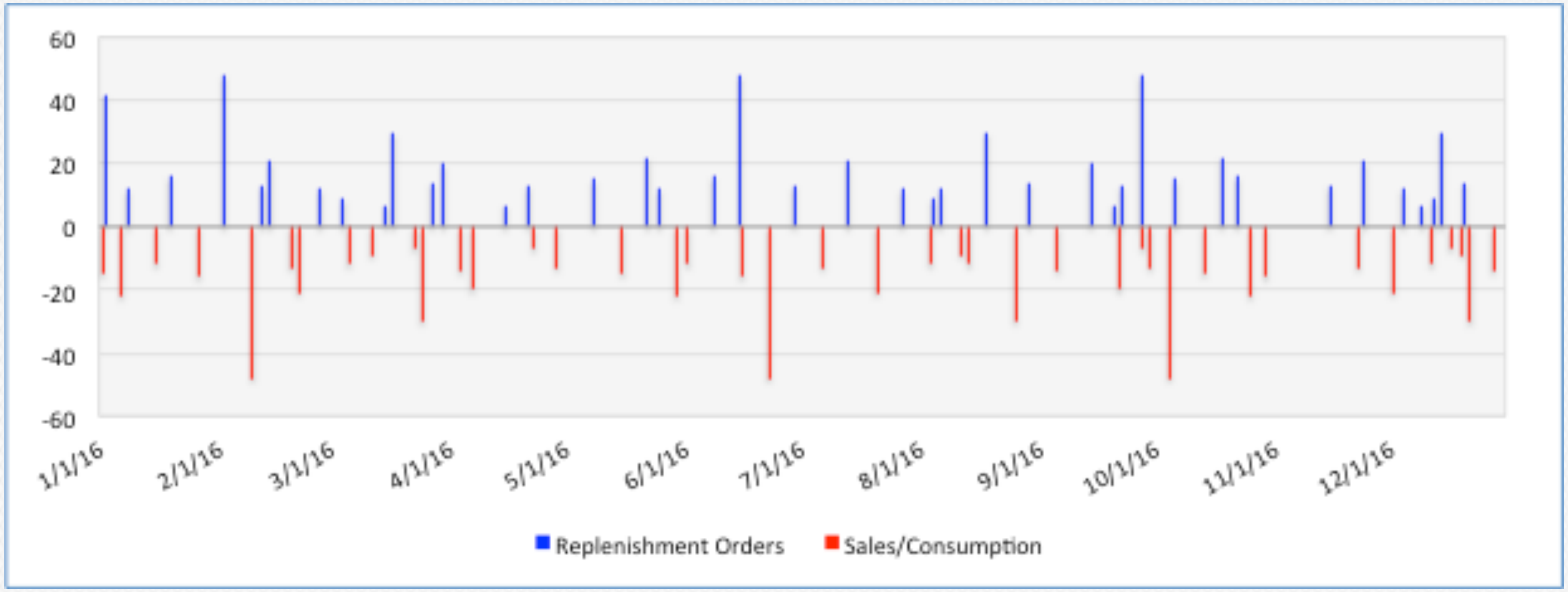
ADU = 2
DLT = 7 days
LT factor = 0.5
Var factor = 0.33

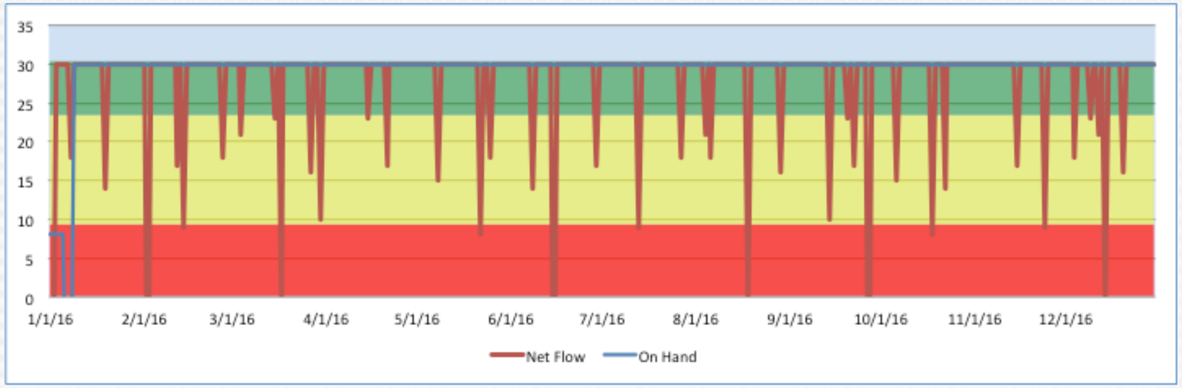
Red Zone = 9 Units

2 x 7 x 0.5 x (1+0.33)

Spike Threshold = 4.5 Units

- When every sales order is a spike, then the system acts as a MTO: for every sales order you place a replenishment order within lead time.
- This defeats the purpose of the buffer completely





- We were doing simulation with hundreds or SKU’s that have this kind of sales profile
- This is not a major issue when MOQ is >> ADU
- For all others we had to correct the issue by manually adjust the red zone and/or the threshold % factor with crazy numbers
- This was fully a trial and error exercise and extremely time consuming
- There must be a better way, a systematic way with some mathematical foundation that can be used to automate this process!

15 0 0 0 0 22 0 0 0 0 0 0 0 0 12 0 0 0 0 0 0 0 0 0 0 16 0 0 0 0 0 0 0 0 0 0 0 0 0 48

15 22 12 16 48

Maybe we should calculate the RED ZONE by ignoring all the Zeroes and taking only the days with sales into account?

- Mathematically this is easy
- If there were **41 days with sales in a year**, that is on average **one day every nine**

1 in 9

- Then we just multiply the ADU for the Red Base calculation by 9
- ADU = 2 → *2 x 9 = 18*
- ADU for the Red Zone Base calculation = 18

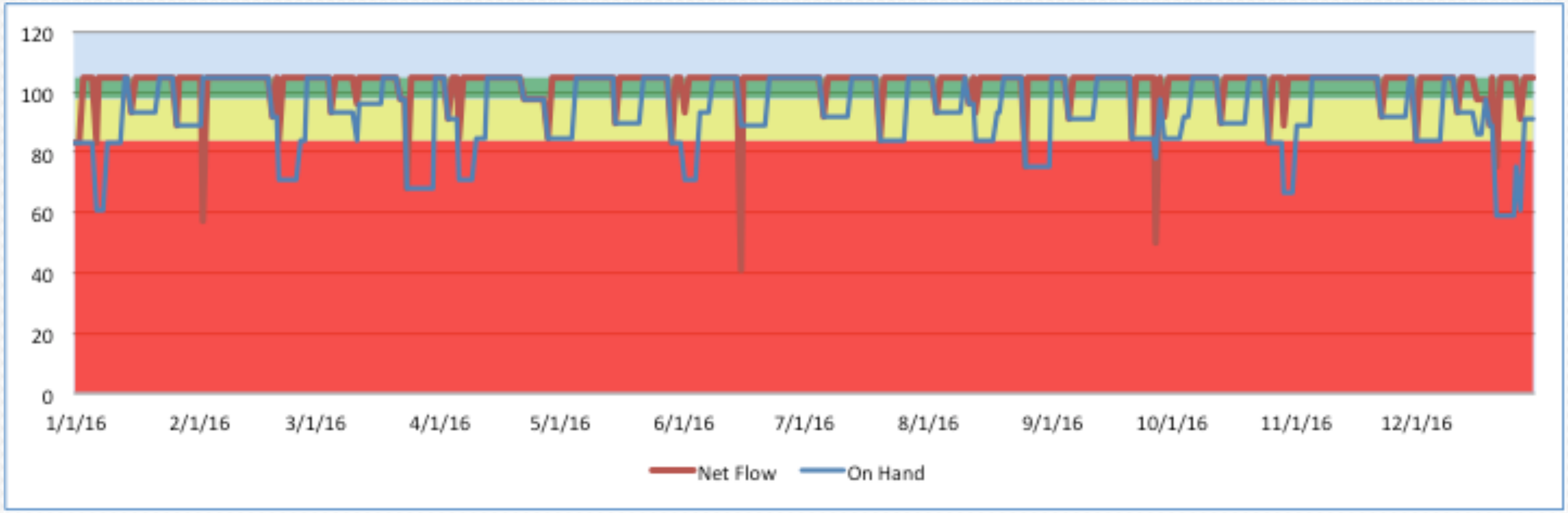
Red Zone = 84 Units

2 x 7 x 0.5 x **9** x (1+0.33)

Spike Threshold = 42 Units

This is looking great, because this is the same Red Zone and Threshold we had originally......BUT!

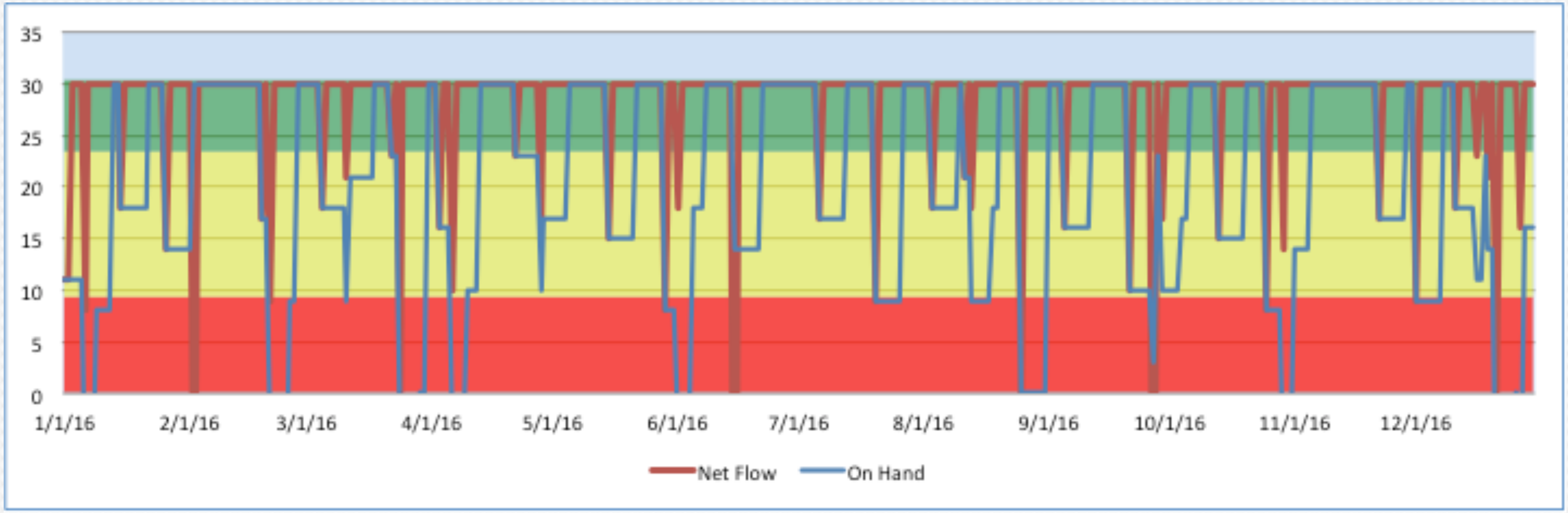
...It feels like this red zone just seems TOO BIG!



- Maybe we can solve the problem by just applying the factor 9 to the spike threshold

Red Zone = 9 Units

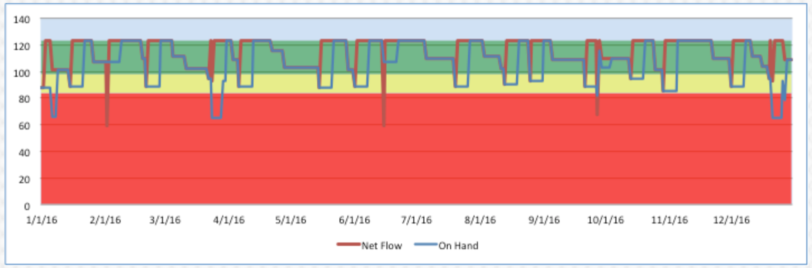
Spike Threshold = 4.5 Units → 4.5 x 9 = 42 Units



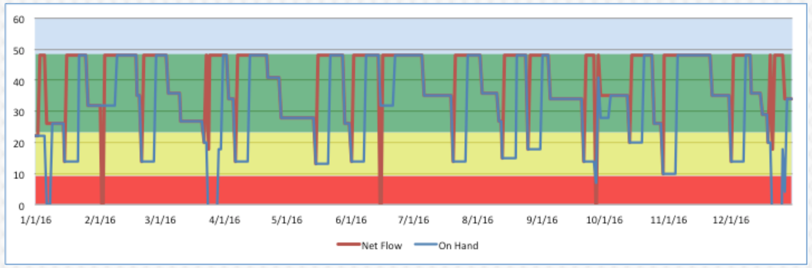
Now the Red Zone is not sufficiently big to protect the buffer!

- Both solutions result in a **Spike Threshold of 42 Units**, which is exactly what we want, but the red zone is either too big or too small
- Is there a middle ground?

Factor 9x applied to Red Base



Factor 9x applied to Threshold



- What about splitting the factor in two components and applying each component to the Red Base and the Threshold respectively?
- We used the Square Root of the factor

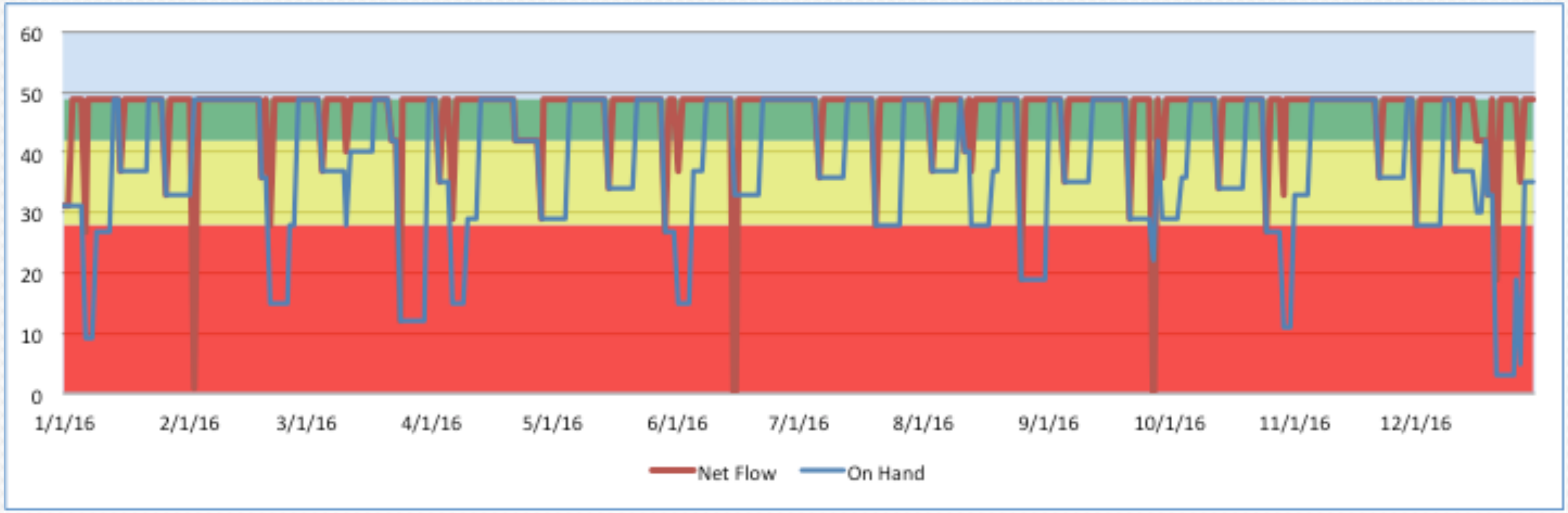
9 = 3 x 3

Red Zone = ADU x DLT x LT factor x 3 x 1.33 = 28 Units

2 x 7 x 0.5 x 3 x (1+0.33)

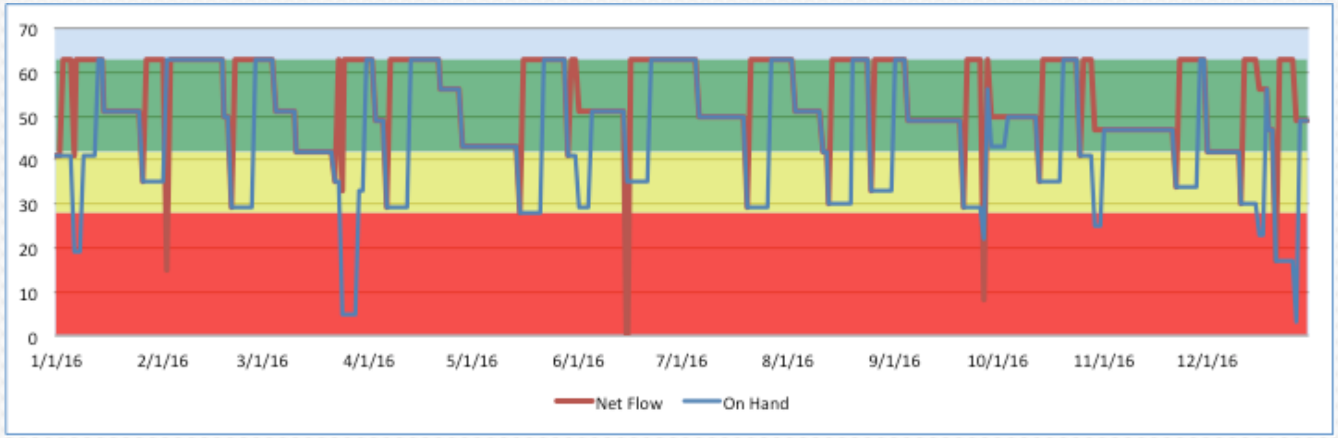
Spike Threshold = 14 units → 14 x 3 = 42 Units

- Finally, we seem to have found an easy way to deal with these products
- When we applied to our simulation we found that with these settings most buffers looked good, with very little adaption needed in some exceptional cases



- The more interspersed the sales are, the higher the factor
- On the other side, when you have sales every day, or almost, then the factor → 1 Sqrt(→1) = →→ 1 The factor doesn't impact the original buffer calculation

- Next step: Explore the benefit of adapting the squared root factor to the Green Zone as well?
  - Benefits: Better buffering of demand variability (because of the small green zone every shipment makes the Net Flow drop in the yellow, thus triggering a replenishment order and passing the variation upstream).
  - Cost: Average inventories goes up



Original Green Zone = 7

	Order Patterns	
	Units	Value USD
Avg. Order	21	21
Min	9	9
Max	64	64
Median	16	16
# Orders	35	

Green Zone = 7x3 = 21

	Order Patterns	
	Units	Value USD
Avg. Order	34	34
Min	22	22
Max	76	76
Median	33	33
# Orders	21	

SmartChain

AGILITY BY DESIGN