
How fast can a charity raise money?

Applying the Theory of Constraints to fundraising operations

By JAKE DELL

You've heard it countless times.

"A chain is only as strong as its weakest link; a team is only as good as its worst player."

The logic is simple, obvious, and we all agree.

But in practice this isn't much more than a pithy saying. We don't often apply this wisdom to our everyday lives.

But we should.

At least that's what practitioners of the [Theory of Constraints](#) (TOC) would have us do.

First popularized over 20 years ago by Eliyahu Goldratt in his "business novel" (and movie) [The Goal](#), TOC posits that organizations are very much like chains: they only work as well as their "weakest" link.

In manufacturing the weakest link might be the process which takes the longest time. Take industrial baking, for instance: fast mixers, a bountiful supply of flour, rapid-rise yeast – none of these change the fact that whole process is held up waiting for the dough to rise.

Finished goods – loaves of bread – *can only be produced at the rate at which the dough rises*. If we mix more flour and water but our dough isn't done rising, do we speed production? No, we just get a big dough ball. If we preheat our ovens before the dough is done rising, do we make bread faster? No, we make hot air and increase our utility bill.

The rate at which a bakery can bake bread will *never* be faster than the rate at which the dough rises. In the bakery's production "chain" (that sequence of events which transforms flour and water into crusty Italian bread or buttery brioche) the weakest link is the time it takes for dough to rise.

In TOC lingo this is what is known as a *bottleneck*.

So how do we make bread faster to feed an insatiable public? In TOC terms, how do we increase *throughput*? Well, we might open more production lines. We might set our development team to the task of cutting dough-rise time.

What we *don't* do is flood the system with more flour and water. Nor do we replace our *legacy* mixers and ovens with a new cutting edge, enterprise-wide, integrated "solution."

Why?

Because mixing more flour and water or buying faster mixers or hotter ovens *won't make the dough rise any faster*. And it's the time it takes the dough to rise which *constrains* the entire process. It's the *drum* which sets the beat to which the whole bakery marches.

So what would TOC have us do?

First we find the bottleneck in our process. Then we schedule *backwards* from the bottleneck. As long as we're baking bread, there should *always* be dough rising. We can release the day's worth of flour and water, the mixers can mix and then shut down, the kneaders can wait for the mixers, *but the bottleneck process should never be idle*. If it does then you will increase your production time and you *will not* be able "to make it up later." You will miss your production goal. Your customers will go elsewhere.

TOC and fundraising

So what does this have to do with fundraising?

Well, let me ask you: do terms like raw materials, inventory, WIP, bill of materials, throughput, and finished goods mean anything to folks in fundraising?

Probably not – at least those aren't terms I've encountered in fundraising.

Does a fundraising organization have resources? Does it have processes (a sequence of events to raise money)? If yes, can and should these resources be scheduled? (If not – well, if not – then what claim can fundraising make that it's a managed operation?)

Fundamentally TOC is nothing more than a method for scheduling resources and processes.

If we agree that fundraising requires applying resources to processes (cultivators making phone calls to potential donors) *and* we agree that this is subject to scheduling (let's put some organization back in the organization) *then* it follows that TOC can be applied to the raising of money.

Why Johnny can't raise more money

A basic (and obvious) flow exists in fundraising. It is the flow of donor assets to the charity's bank account. The finished good is a gift – cash in hand.

Donor's assets and mailing lists rented from other organizations are our raw materials; wills, pledges and our active appeals are our work-in-progress (WIP) and inventory.

There is more than one way for a donor to give: bequests, gifts of art, real estate or other tangibles, cash or buying sweepstakes tickets. These are products and product lines.

Certain things have to happen in order for a gift to be realized. Appeals must be designed and developed. Envelopes must be stuffed and mailed. Potential donors must be identified, called, met and asked for money. The charity must inspire in the donor a desire to give. These are our processes.

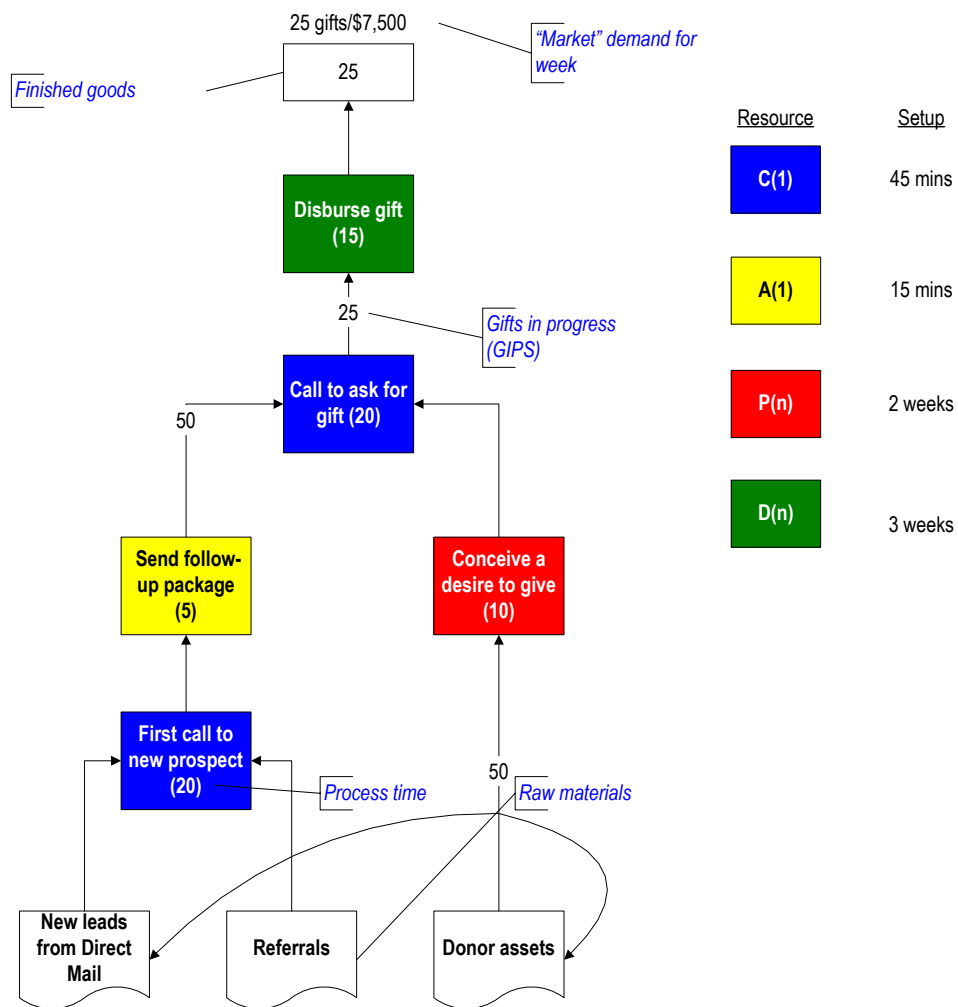
A concept familiar to sales units in for-profit companies has also been adopted by non-profits: the *pipeline*. The pipeline is a measure of gifts in progress (WIP or, GIP, if you will) and also is a rudimentary attempt at *scheduling* resources. Those gifts closest to being funded get the most attention.

So where are our bottlenecks? Where is our “rising dough” process – the drum which beats the pace and determines *how fast* a charity can raise money?

Consider the following process diagram depicting a generic annual giving program. The processes are listed on the left and the resources which those processes consume are listed on the right.

The number in parenthesis in the resource box is the number of resources of that type available at any one time. So, for instance, there is only one C or A resource available at any given time, but there are n number of P and D resources available.

Keep that in mind.



Where is the bottleneck? Where is the dough just sitting there rising?

Is it the two week setup time for the P resource? Is that why 50 units of "raw material" (assets from potential donors) are backed up in front of it?

Possibly.

But what if the set-up time for resource P was much, much shorter – say 10 minutes? Would it make a difference? At 20 minutes each, can one C resource make 50 calls in one day? Of course not!

The long setup times for the P and D resources are deceptive because they give the appearance of creating a backlog. But in reality the P and D resources are virtually unlimited. They represent the vast pool of people waiting to be called – prospects and donors. The only limit here is the number of them we can call.

Let's put that another way: what keeps this system from reaching more people? There are more P and D resources to call than C resources to call them. The A resource can make packets four times faster than the C resource can make calls. The P, D, and A resources all have *excess capacity* while the C resource is maxed out.

So it should be clear: our sample fundraising system is *constrained* by the number of calls which can be made in one day. Many prospects and donors are simply out of reach because the system cannot extend its grasp further.

Charitable throughput

In our bakery example we said that for as long as we're baking bread there should be dough rising. Mixers and kneaders can stand idle but dough must rise. There is no making up for lost time here. And if there is no dough rising we're losing time.

We said that in order to make sure our bottleneck is never idle we need to schedule backwards so that the kneaders and the mixers and the release of flour and water all work together so that a dough ball is set to rise "just in

time” – when one batch moves on and the resource is ready for the next batch.

In TOC this is called *exploitation* and *subordination*. We discover the constraint, decide how best to use it, and then arrange everything else around it.

In our scenario, the C resource (those people we’ve hired to cultivate potential donors) is limited by the number of calls which can be made in one day. If callers aren’t calling then gifts are getting backed up in the system. Money stops flowing from the donor to the charity. Throughput goes down. Goals are missed.

The C resource sets the pace. A fundraising operation marches to the beat of the C resource drum. It *can’t* raise money any faster than what the capacity of C resource allows.

It seems obvious doesn’t it? Most fundraising managers know that if their callers aren’t calling, money isn’t coming in.

But are fundraising organizations really organized around this principle? Are they doing every thing can to keep their C resources busy *all the time*? How often are the scarce C resources tied up in meetings and non-throughput generating activity? How often are the C resources idle because “raw materials” (leads, referrals) aren’t released on time?

Are the goals and tasks of other departments, such as Information Technology, subordinated to the needs of the C resource?

Faced with a choice between helping a user recover a lost word-processing document (as important as the document may be), generating a report for

upper management, or feeding leads from the Direct Mail database to the C resource, which one *should* you chose?

Which one, more often than not, *gets* chosen?

Subordination means clearing the decks so that your scarcest resources – the system's constraints – are always put to good use.

So, how fast can a charity raise money?

No faster than its capacity to reach potential donors. Let's make sure we don't do anything to slow it down more.

Jake Dell is a business analyst and writes from Guilford Connecticut.