

The Muda Diet

Seven Wastes – Waste 2 of 7

By Gary Sage (Transformation Director)

Inventory waste, or; 'why you should eat your ice cream and why you shouldn't keep it in your deep freeze'

In this article we are going to briefly explore the second waste of lean – today, we will treat In-Process Inventory and Partially Done Work as a waste category, and we will discuss some ways to identify and deal with it.

Remember the first principle of lean is to, “eliminate any activity in your process that does not add value to your end user”.

Before we continue let's take a moment to further understand this notion of Lean and Waste. Waste is everywhere. In almost every business waste is often well camouflaged – it lurks, often in full view but somehow hidden from sight – it's often the elephant in the room that almost everyone is blind to. Waste hides in places like admin. systems, hiring standards, PSLs, compliance, governance, security protocols, operating models- anywhere where things are required via 'Compliance' to be consistent - regulated.

Finance Systems are often riddled with waste issues – often these Finance Departments (and HR Administrative Departments) unknowingly collude in inefficiencies that 'allow' a company to constantly lose income and haemorrhage talent, revenue, resources and people in the form of Lean Waste (this is why senior management support is needed in any Lean growth initiative).

Waste is destructive - it's like a virus; in the sense that a virus will constantly undermine the wellbeing of a body. More often than not, the very areas that purport to be the most efficient are a harbour for the biggest, 'wastiest' (sic) culprits.

Compliance, HR and Governance teams create (and enforce) wasteful behaviours the moment at which they depart from a culture of collaborating and innovating within the enterprise. Waste starts when these teams fall into what Peter Hawkins calls the Parcifaltrap - they stop serving beyond themselves, (namely the enterprise) and begin to serve themselves to the point of institutionalised narcissism. The moment these departments become self serving, and begin to enforce their 'will' via command and control methods, huge wasteful expense and productivity waste are sure to follow – along with a decline in morale and motivation.

How about knowledge workers and IT projects? As long as partially done software is not fully incorporated with the rest of the software project under development and finally released into production, we have no idea if it will eventually work or solve the business problem. If defined inefficiently (read defined anytime from weeks to years back) there comes the time at release where 'new' software is obsolete resulting in a huge financial problem.

The most obvious effect of carrying wasteful stock is that there is always a need for up front capital to carry the stock – neither position provides a sensible return. The excess simply sits on a shelf. By actively managing the stock required to operate, a business can deploy the capital it requires in order to support those operations far more cost effectively. I have helped in some cases to cut excess stock by over 90 percent! Cost savings here can be, and mostly are quite significant.

Stock affects a great extent of other expenses. In a craft brewery, as stock levels go up, the capital investment in warehouse space, refrigeration, security and even statutory up front duty also increases. Big warehouses require longer processing and retrieval times to move product into storage and out of storage locations. Consider that high stock levels of unfilled beer bottles also serve to pile up and provide clutter and excess stock expenses.

The incidence of stock damage also increases, as more stock is stored for longer times. Craft beer is perishable, so the longer it sits before being consumed, the longer thirsty people are deprived of fresh beer. Then there is the admin. aspect - the larger the stock, the more resources are required to manage the stock, which means record keeping, a further expense. A brewery that has two days' stock on hand will have a fast and easily available physical stock system compared to a plant with 6 months of product in storage.

In the IT space this generally reveals itself as work (story/module/code drop) that is not completely done, based on one's definition of 'Done', and consequently it cannot be demonstrated or it cannot be released (or sometimes even tested). It encompasses code that has not been re-factored, code that has not been unit tested, code that has not been integration tested, code that is not properly documented, code that is not deployable, essentially code that is not fit for purpose. It's analogous to a brewery having hundreds of bottles, all labelled and ready to go – but without beer in the bottles and caps to seal them the pubs and restaurants aren't buying – which means it's not providing any business value. If it is just sitting there not adding value to the end user the business is generating waste – pure and simple.

My favourite 'war' story in the form of *In Process Inventory and Partially done work waste*. It is a true story about a particular 'late' project that was so shockingly managed that the hardware it was architected to run on was obsolescent and out of support before the software was even in a position to be tested. Worst of all the company bought all the hardware up front 'for a good price'.

This particular Machiavellian masterpiece of torpid waste was 'mitigated' through a politically oppressive and baleful steering committee, (who even managed to proscribe the CTO from revealing the elephant in the room). It was mismanaged so devilishly well (sic), that it was able to drag on, overdue, for four years and tens of millions in added costs.

Admittedly this project was plagued by many more of the seven wastes.

However, if this company had a dedicated Lean Team in place to audit the project it would have been stopped in the first month. Interestingly, this waste phenomenon is not uncommon – many of you dear readers will have a war story – and for my part, the anecdote above is one of my favourites.

In addition, partially done software gets in the way of other developments, so any work that is not finished triggers, for example, huge delays in the development (Waste number 4) and later on when someone tries to pull it all together we run into another waste, namely Extra Processing and Rework (Waste number 6). In many cases these delays impact contracted experts, who are kept 'just in case' in order to hold on to talent and subject expertise or they are 'let go' until needed (causing rework and relearning and handover issues) - either way the knock-on effects are costly and wasteful . . . and avoidable, when managed correctly.

When the pickers delivered the grapes to the Château de Beaucaste during August 2010, the cool nights had provided grapes from 19 varieties, all with near perfect acidity. The wine team under Marc Berrin made 7500 cases from that year's harvest. Today that wine is available as Château de Beaucastel - Châteauneuf-du-Pape, it sells for £100 a bottle – it's rated 8th best wine of 2013. Minimising waste is about seizing the moment, because no-one buys Château de Beaucaste raisins at those prices.

Partially done work is arguably the nastiest of all the wastes, because for many people their job may be completed individually to a high standard, however the fruits of their labours are laid to waste – waiting unappreciated and unused – furthermore unvalued.

This in-process inventory form of waste is really a sign of bad management or lack of competency. People who are charged with creating the partially done items are paid for their effort – and however perfect the components of the unfinished business, the work that is done is valueless – while it sits in a zone of 'uselessness' (i.e. it costs money to create build or design but has no value while it is not usable.) Worst of all, we need to realize that until this work is done, one cannot know whether there are quality issues, so you cannot know that the customer will be happy.

So what do we do about this waste?

- **Examine and review needs with the product owner** - really understand the functionality of the story, code or function, make it a mission to get an understanding of what value it adds to the final output. In IT this is where most issues arise. Communication is key.
- **Agree to prioritize each set of requirements into some form of execution plan or sprint.** This is the point where the professionals differentiate themselves from the fools - this prioritization requires discipline, superb synchronization and true collaboration between the delivery team and a product owner.
- **Openly and transparently evaluate the technical complexity** – if it's too complex, suggest the product owner splits it - or find an implementation approach that fits your context.
- **Take a Post-Agilist approach** and avoid the Mini-Waterfall approach like the plague.
- **Create a culture of coordination and collaboration** between the team and the product owner. Occasionally a product owner may not recognize the need for some technical dependencies - help the product owner.
- **Become systemically transparent and inclusive.** Show stakeholders how things work.
- **Begin to apply notion of 'Real Options'** to your planning and prioritisation.
- **Quicken your feedback loops.** Shorten your iterations. Close the gaps.
- In the case of excesses in in-process inventory, eliminating breaks in the work in process can be accomplished by **designing a continuous flow through the production processes.** Every interruption in a process requires work to be stored and then moved between processes. Eliminating these breaks eliminates the need to check incomplete work in and check it out again.
- **Apply realistic forecasting, and design thinking.** Moving to pull based scheduling systems will help to reduce the extraneous 'safe level margins' and variations that lead to overproduction and increases in inventory levels.
- **Provide clear incentives** to motivate employees to minimize excess stuff or product. Management must treat capital tied up in unused stock or work (story/module/code drop) as an undesirable situation that is to be minimized.
- Unfortunately, a full stockroom has the propensity to cover up for other failings, it also often makes people think inventory is essential – an asset which is misguidedly accorded value, when in reality it's the equivalent of stockpiling ice cream for the winter. Realistically, correcting the underlying problems will make the stock unnecessary – **try to shift from 'just in case' thinking (guessing?) to constantly asking 'how much and by when and for whom' and monitoring.**

See more tips in It's a Post-Agilism world! Simply [click here](#) to read the article.

Some Criteria for Success:

Generate buy-in. Senior management will need to commit themselves to vigorously supporting waste elimination. (Starting a pilot programme often works exceedingly well.)

Be brave – it takes a brave soul to face off and go toe to toe with the ‘keepers’ of inefficiencies that are governance driven.

Start Small - many forward thinking organisations start dealing with their wastes by starting small – they establish a small team, perhaps bringing in an outside experienced Lean Specialist for a short period.

Fall in love – with value stream maps - they can be used to map the flow of work from the start of a process (ex: client request) to the end of the delivery process (ex: system delivered to the client). They are analogous to traditional process diagrams, however the differentiator is they contain much less detail, the idea is to begin getting things re-presented into understandable process areas. First you must see it – paint a picture - **MAKE IT VISIBLE!**

Communicate and collaborate - removing waste relies on both high levels of competency, design thinking, ingenuity and a lot of common sense so transparency and collaboration are vital to your success.

Ensure there is nothing to hide - make transparency a priority, to achieve this you will need to make decisions based on data. Ensure that all data is visible and create one working version of the ‘truth’ - root out selective transparency by banning ‘eyes only’, private or ‘sensitive’ reports in changing environments. Measure and publish the cost of breakages, late delivery, bugs defects and returns. Make sure data is visual and presented as measurement – do not ever use data to assign blame – instead use data to discover and identify the root cause of a problem and use it to justify transformation.

Learn to disagree gracefully - disagreement does not constitute intolerance - intolerance is when we close our minds to other mental models and suppress any challenges. Try to ‘walk a mile in another man’s moccasins’ - and (again) resist selective transparency.

Never equate bad communication skills with stupidity or bigotry. F Scott Fitzgerald maintained that “The test of a first-rate intelligence is the ability to hold two opposing ideas in mind at the same time and still retain the ability to function” - so a question versed in bad grammar, or a problem pointed out by the cleaner still needs to be heard.

Deal with it in chunks. Many newly formed lean teams make the mistake of trying to do too much in one cycle. The best solutions arise out of iterative cycles or sprints that when completed, rely on a series constant improvement and monitoring for long term sustainable success.

A tip on food waste:

Don’t waste ice cream: Have you wondered how, ice cream which is stored in a deep freezer, somehow ‘grows’ those annoying ice crystals and gross ice lumps? They form when the water in ice cream separates from the fat making it gritty and eventually it develops even more annoying larger ice granules.

The result is a gritty grainy texture, then there are those gross ice lumps that are equivalent in experience as the fishbone the chef forgot to remove from your sashimi. In low-fat ice cream this phenomenon is even more ubiquitous and horrible.

As long as the water remains surrounded by an emulsion with fat in ice cream, the original ice crystals will not get larger. To save ice cream from developing these yucky large ice crystals, do not refreeze ice cream once it has been opened, and also do not store ice cream below -15° C / 5°F for more than 3 days.

In the case of ice cream – don’t spoil it by keeping it – that would be wasteful. Eat it! Technically a deep freezer should be used to prepare your ice cream not store it. Ice cream should be eaten fresh, not stored. That’s why it always tastes better in Venice and Rome and on the waterfront in Ascona.

FURTHER INFORMATION AND RESOURCES:

- If you have a war story that equals (or tops) my anecdote above, please drop me a line and let me know.
- In the following article in this series we will probe deeper, into one of the more misunderstood wastes, that of Waste No. 3: Motion waste (Multiple Tasking, Task Switching) and we will explore some ways to identify and deal with it.
- To read the previous article in this series (waste 1 of 7 :Transportation waste) please [click here](#)