

The IT Risk Manager

... based on practice rather than theory.

August 2, 2015

Agile Tools don't fix problems, they reveal them.

By theitriskmanager

Sometimes we are lucky to work with someone whose insight into the work we do has a profound affect on the way we see what we are doing. One such person is Tony Grout who changed the way I look at Agile. "Agile doesn't tell you how to write software" Tony said. "Agile provides you with the minimal tools that show you where something is wrong, tools that show you where you have a problem." As Dan North would say Agile practices illuminate or visualise problems, they don't necessarily solve them.

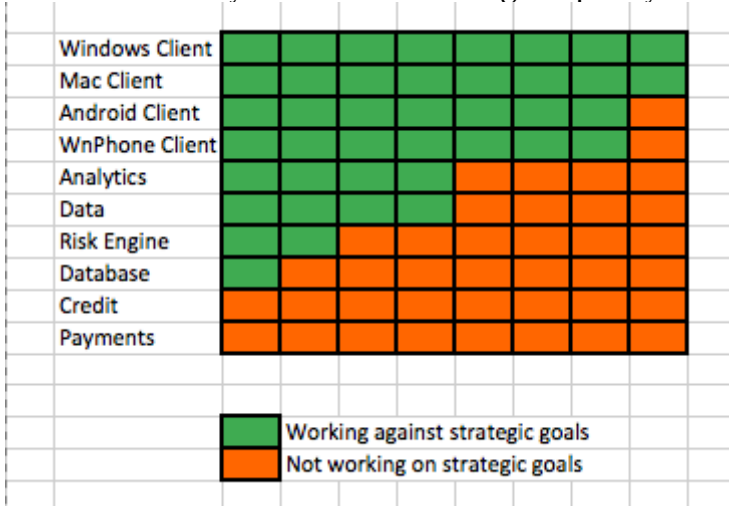
Over the past few weeks I have been working with a client to set up Capacity Planning and Metric Mapping (the name for the practice of linking a hierarchy of metrics using hypotheses). I've done both a few times now so I was able to take a step back and observe what was going on. Both Capacity Planning and Metric Mapping show you problems to be solved. Neither tell you how to solve the problem.

Unlike some Agile Scaling Frameworks, Capacity Planning does not tell you how to make decisions. Capacity Planning does not care whether you use HiPPOs, or business cases or some messed up version of Weighted Shortest Job First. All Capacity Planning tells you is that you have to make a decision. This allows the organisation to evolve how to make decisions. It allows the organisation to take "safe to fail" steps with change, cultural, process or otherwise. All Capacity Planning does is show the problems, it does not dictate the solutions.

What are the problems that Capacity Planning reveals?

1. It is common for the product owner to have one-to-one conversations with different stakeholders/decision makers. Each stakeholder applies pressure to the product owner to get what they want. When the product owner and team fail to deliver they are blamed even though the problem was caused by unrealistic expectations. I call this the hub and spoke model. Each stakeholder (who may be a product owner themselves) individually applies pressure on the product owner who has to resolve these organisational priorities. The Capacity Planning process puts all of the stakeholders in a discussion together and takes the product owner out of the middle. The product owner now simply states the capacity they have available and the effort necessary to perform each piece of the stakeholder's initiatives. The stakeholders are now forced into a conversation where they have to work out the priority of the work amongst themselves. Capacity Planning does not tell you how to resolve the priorities but it means everyone needs to think of the organisation's goals rather than just their own.
2. Capacity Planning reveals if the organisation's reward mechanisms and appraisal system creates behaviour that are directly in conflict with the goals of the organisation. The organisation wants the most valuable work to flow through the constraints. The reward system drives individuals to achieve their own personal goals rather than the goals of the organisation. Once again, Capacity Planning does not tell you how to fix your reward system, it just highlights the problem.

3. In order to run a Capacity Planning session it is necessary for each initiative owner to get a Sweet Wild Asses Guess (SWAG) estimate from any team that will need to contribute to the initiative. Capacity Planning does not tell you how to come up with a SWAG other than to warn against putting too much effort into a throw away piece of work. The SWAG indicates that product owner has acknowledged the existence of the work and feels confident enough to give an estimate. In effect, The SWAG is a record that the owner of the initiative has had a conversation with the product owner of the contributing team. Capacity Planning forces the conversation. It does not say what happens in the conversation and it does not say how the estimate should be calculated. The SWAG simply provides the Capacity Planning facilitators with a mechanism to track the conversations. The SWAG forces organisations to have conversations before initiatives start rather than when they are mid flight.
4. One of the outcomes of Capacity Planning is a list of the teams that are constraining the organisation because they do not have enough capacity. It also shows the teams that have excess



capacity.

(<https://theitriskmanager.files.wordpress.com/2014/02/capacity-plan.png>). Capacity Planning does not tell you how to move “work to the teams”, or “teams to the work” (Another Tony’ism). It simply shows you where you need extra capacity and where you already have excess capacity. There are many solutions for “moving the work to the team” or “moving the team to the work”. (The Bank of America Case Study (<https://scrumalliance.org/community/articles/2013/2013-april/how-to-form-teams-in-large-scale-scrum-a-story-of>), offers one of the more interesting solutions as it is more dynamic than others than mandate long lived teams.)

Capacity Planning makes it clear that the ability of an organisation to deliver over the short term (three months) is based on the capacity of the constrained teams and its staff liquidity, and has little to do with the budget for the organisation as a whole.

In summary, Capacity Planning shows you where your stakeholders cannot prioritise, it shows you where initiative owners are not communicating with the teams that they need to deliver value, It shows you where capacity is in the wrong place. It does not tell you how to solve these problems, it just makes them transparent.

Metric Mapping is the process by which a Product Manager picks three or four key metrics to demonstrate the success of their product. They agree these metrics with their manager. The manager also has three or four key metrics. The manager has a portfolio of (product) metrics and they have a hypothesis that if their product managers improve their metrics, in turn their manager metrics will improve. The product managers have hypotheses that if they improve some aspect of their product, their metrics will improve.

Metric Mapping requires engineering managers to have metrics. Work that is done to improve the “non functionals” of the product. The head of engineering should also have a couple of metrics. Ideally one for customer perceived quality, and the other lead time based (actually duration or

weighted lead time).

All work should be mapped to these metrics. This allows everyone to have a portfolio view of the investments made by the organisation and adjust investment strategies accordingly. Metric mapping does not tell you what the metrics should be. Metric Mapping does not tell you how the metrics should be chosen. Metric Mapping ensures that there are conversations between the different levels of the organisation. These conversations ensure consistency and coherence between investments and what the organisation considers success to be. The portfolio view shows where there is too much or too little investment. It does tell you how to fix this problem.

Metric Mapping allows everyone to see whether investments are coherent, and whether metrics between different levels of the organisation are coherent. For example:

- A product manager says they are investing to reduce web site page load times to improve customer satisfaction. This is coherent.
- A product manager says they are adding adverts to the web page to improve customer satisfaction. This is incoherent.
- A manager has a metric to improve customer satisfaction and so agrees with their product manager that one of their metrics should be call centre call length times. This is incoherent

Managers should set goals based on their metrics. They should not set goals based on their subordinates metrics. Furthermore, they should never tell the subordinate how to improve a metric unless the subordinate asks for their advice. They can of course point out where investments to move metrics are not coherent, and they can help them to understand how to move metrics as a coach. Metric Mapping helps organisations to see problems and incoherent investments. It allows everyone in organisation to identify investments and hypotheses that are not coherent. It does not tell you how to solve the problems.

And so Capacity Planning and Metric Mapping align with Tony Grout's observation about Agile. Both show problems but don't tell you how to solve them. The solution depends your context. In your context, you may form a release train to plan your work for the next quarter. Or you may bring everyone in your department into a room and ask them to self organise. Or you may use a tick list. Depending on your context, one of these solutions might be good or bad. There are however some solutions that are less likely to work than others.... We Shall Just Forget (WSJF) those.

About theitriskmanager



Currently an "engineering performance coach" because "transformation" and "Agile" are now toxic. In the past, "Transformation lead", "Agile Coach", "Programme Manager", "Project Manager", "Business Analyst", and "Developer". Did some stuff with the Agile Community. Put the "Given" into "Given-When-Then". Discovered "Real Options"

This entry was posted on Sunday, August 2nd, 2015 at 8:34 am and posted in Uncategorized. You can follow any responses to this entry through the RSS 2.0 feed.

8 responses to "Agile Tools don't fix problems, they reveal them."

- [Anthony Green \(@anthonycgreen\)](#)

[August 2nd, 2015 at 12:02 pm](#)

“‘Agile doesn’t tell you how to write software’ Tony said.” Could you expand on what you interpret that statement to mean. I wouldn’t claim XP tells you how to write software but it does advise you on the things the authors have experimented with and found to catalyse toward the desired state: loosely coupled, highly cohesive, composable, context-independent units for systems with evolutionary potential.

Reply

o [theitriskmanager](#)

August 2nd, 2015 at 6:14 pm

Hi Anthony

I understand it to mean that although Agile Tools can help you build better software, the real value that they give you is that they help you to understand the problems that you have to solve. For example:

Scrum: Fundamentally we strip everything down to barest and asks “Did we finish what we said we would finish?” It shows us why we cannot ship after a sprint.

Kanban: Where is the constraint? Where do we have to add capacity in order to optimise the flow of deliver.

TDD: What code did we just break?

Pairing: What is wrong with the code I just wrote?

Retrospective: What can be done to improve the process?

They are all structures that help us focus on finding the problem. Once we understand the problem, we can come up with a solution according to our context.

Importantly they are processes to help a pair or group of people clearly identify the problem. They create visibility into a problem so that the group can come to the consensus that it needs to be solved.

In summary, Agile tools do help with solutions, however their real superpower is around creating consensus on where the problems are.

Does that answer your question?

Reply

o [Tony Grout](#)

August 5th, 2015 at 8:31 am

Hi Anthony

My expression is a simplified way of saying that the greatest differentiator that agile has contributed is the maniacal focus on building fast feedback loop method and mindset at the personal, team and organisational level. Without these we wouldn’t have had the self awareness to create the valuable contemporary software development practices that help address the challenges fast feedback allows us to see, fix and immediately benefit from.

Hope that helps clarify my thoughts.

T

o [Anthony Green \(@anthonycgreen\)](#)

October 7th, 2015 at 6:24 am

I'm with Liz "'uncovering better ways of developing software' is not agile's differentiator"
<https://vimeo.com/140951116>

- Anthony Green (@anthonycgreen)

August 2nd, 2015 at 3:24 pm

Where does Metric Mapping differ from Impact Mapping? In it's intended aims? In it's application?

Reply

- theitriskmanager

August 2nd, 2015 at 6:28 pm

By Impact Mapping, I assume you mean Feature Injection. 😊 Feature Injection and Impact Mapping are about mapping features to value.

Metric mapping is about this process at scale, where there is a heirarchy of metrics that possibly map to an organisation hierachy in some way.

Metric mapping solves a number of problems:

1. Product Development creates hypotheses between "Features" and metrics. For it to be effective, it is necessary to use metrics that respond quickly to changes in the product. i.e. we need fast feedback loops. However as a business at a high level we want to move metrics that may be slower to respond. As such we have a mismatch between fast moving low level metrics and slow moving high level metrics.
2. We need high level metrics to coordinate and collaborate across diverse parts of the organisation. PMs need a higher level metric to provide context for prioritisation discussions. e.g. One PM is trying to improve load conversion and the other is trying to improve customer service rating which have nothing in common, however they are both trying to improve "Activity".
3. Linking lower level metrics to higher level metrics through hypotheses allows for a conversation up and down the hierarchy. Everyone can see the hypothesis and challenge on the basis of coherence.
4. Having high level metrics provides managers with a portfolio view of investments which is not possible using features. It also provides them with a steering wheel (e.g. We want to invest more in "Activity Churn" and less in new customer acquisition.) without the Managers micro-managing and telling PMs which low level metric to improve or even worse, what features to build.

I do not believe Impact Mapping addresses these organisational issues, although it does provide a fantastic way to facilitate some of the discussions. (Feature Injection certainly doesn't)

Does this answer your question?

Reply

- thecodecleaner

August 3rd, 2015 at 8:17 pm

Reblogged this on @thecodecleaner.

Reply

- *Agile Tools Reveal Problems—Not Fix Them | AITS Agile*

August 18th, 2015 at 7:32 pm

[...] At his full post, Matts continues on about what metric mapping can and cannot do. You can read it here: <https://theitriskmanager.wordpress.com/2015/08/02/agile-tools-dont-fix-problems-they-reveal-them/> [...]

Reply

Subscribe to RSS

Blog at WordPress.com.