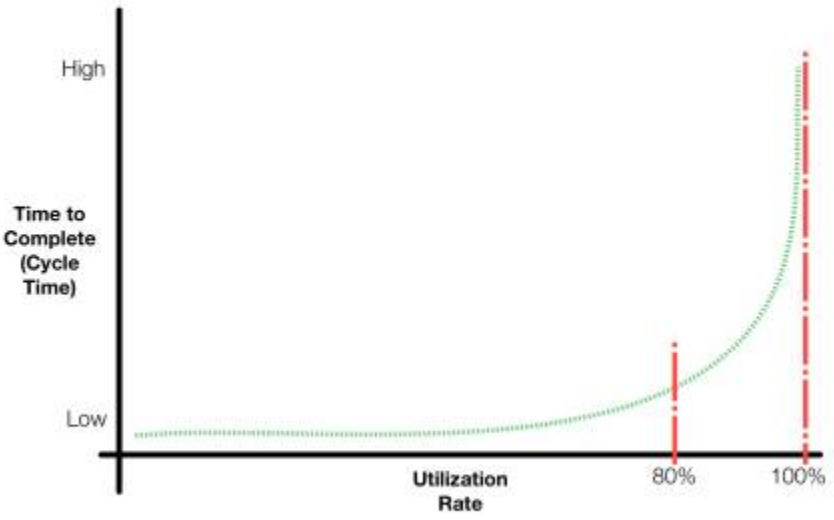
Table of Contents

What Is Wrong With 100% Utilization Thinking?

As a person who has spent way too many years managing many projects, working with a lot of financial people, and who has worked to improve the results of a lot of organizations, one thing I was expected to focus on was “utilization rates” for people. The thinking seems straight-forward - if we are paying a certain amount of money for a person we can increase the return (value produced) associated with that person by making sure, at a minimum, that the person is working all the time and is always busy. As result I'd work to ensure that people were 90%, 100% utilized and be happy when people reported 110% utilization.

Here's the problem with this type of thinking. Its wrong. Its wrong in so many ways. And the problem is that its wrong in ways that are counter-intuitive to what seems like a pretty simple relationship between cause (people busy) and effect (more stuff).

Understanding Real Cause and Effect of Utilization



I want you to have a look at the chart[1)]:

This is what actually happens when you increase the utilization rate beyond a certain threshold. Basically it says that “as we approach 100% utilization the time it takes to process something becomes exponentially large.” This is an application of “queuing theory”. It turns out that every time you halve the amount of excess capacity, you double the time it takes to process something. So as you move from 60% to 80% utilization you double the time it takes to process something, and moving from 80% to 90% will double it again.

What does this say about loading up a person's work to 100% utilization? Basically if you are focussing on utilization rates there will come a point when you are valuing someone being “busy” over someone producing something in a timely manner.

Interestingly you instinctively understand this effect even if you have not applied this to software. If you have a road system filled to capacity with cars and you are a car trying to get from point “A” to point “B” what are your chances? A “road system filled to capacity” is a traffic jam and so you will take a long time to get to your location.

Now What Are You Going To Do?

I am pretty sure thats not what you want.

As I've worked on projects I've had utilization rates of 95% and above, but it is pretty clear that this is not a good thing. Discussions with just about every organization I've worked with has similar thinking.

But also know that this is a very hard thing to change in an organization when you have been thinking this way for years.

Think, for example, about your reporting to finance, or project reporting, and you scratch the surface of why this is so hard to change. Think of the “upfront planning mindset” which implies “we have this resource pool of people and to maximize the amount of work we put through our system, we need to make sure we have allocated everyone to work” which results in managing matrices of “resources” and allocating people 1/10 of the time in 4 months time to this project task.

The first step along the way is education. People need to be made aware of the problem. Then we can start working the issue. There are lots of approaches we can take (for example, reducing the amount of work we have in progress thereby also increasing the focus we have on high priority work), but the first step is to make this thinking clear to all. If you have an organization that “believes” high utilization rates are required for success, then you have a problem until that view can be changed.

Once you have this in place queuing theory gives us six rules for reducing software development cycle time [2)]:

1. Limit work to capacity (don't try to take on too much at a time)
2. Even out the arrival of work (budgeting, approval and contracting processes tend to mean arrival of work is in big lumps and not even)
3. Minimize the number of things-in-process (work-in-progress, or WIP)

4. Minimize the size of the things-in-process (small batch sizes - see What is the Effect of Batch Size on How Long Something Takes to Get Done)
5. Establish a regular cadence (establishes a predictable workflow and a reliable pace, allows for faster feedback and for staged delivery)
6. Use pull scheduling

Want to Know More?

- "Slack: Getting Past Burnout, Busywork, and the Myth of Total Efficiency" by Tom DeMarco. Discussion about the counterintuitive principle that explains why efficiency efforts can slow a company down. Fights against the concept that, for example, 100% utilization of people is the most efficient way to get something done.
- "Managing the Pipeline" by Mary Poppendieck. Mary is one of the original thinkers in the are of lean principles applied to software development. Her essays are always useful and get more useful as you need to grow your implementation of agile and lean to the overall organization.
- Economic Value of Slack Time
- How Does Utilization Impact Lead Time of Work: Excellent interactive model by @TroyMagennis allowing you to explore different how the Kingman formula works.
- A Myth of 100% Utilization

Enterprise, Utilization, Slack, Risk, RiskManagement, FAQ, PresentationIdea, LPM, portfoliomanagement

1)

this version of chart results come from The Principles of Product Development Flow: Second Generation Lean Product Development - Don Reinertsen which was originally from the Kingsman Formula, although it has been liberally interpreted to aid in understanding.

2)

List of things you can do comes from "Managing the Pipeline" by Mary Poppendieck