

# TRADE-OFFS

## Idea in brief

**In faster delivery, speed itself is not the goal. But reducing cycle time helps deliver products to market faster, improves productivity, creates faster feedback cycles, and allows us to become more responsive to customers and take advantage of opportunities that may emerge.**

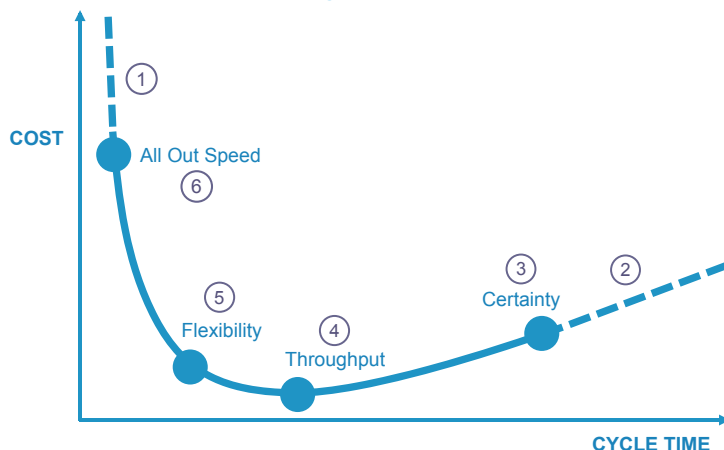
In order to improve flow and reduce cycle time we need to manage queues, visualise and limit work in progress, reduce batch size and overlap different activities such as analysis, development and testing.

As cycle time decreases, cost stays low for a long time or may even decrease further, while the shorter cycle times improve flexibility. But past a certain point choosing which benefit to focus on becomes more important. The focus that is right for your business and the resulting choices that you will make depends on the business need and the customer's view of time.

Most organisations today focus on something they think is terribly important – certainty – which requires detailed plans, exact specifications and big design up-front. This leads to lengthy cycle times and increased costs. The opposite of what they could be doing to achieve faster delivery. To get the full, sustained benefits when using the tools for faster delivery companies should embrace uncertainty and move to become more efficient, faster and more flexible.

## Ideas in practice

### The Faster Delivery Model



The Faster Delivery Model presents a relationship between cycle time and the total cost per development hour. It doesn't capture the full complexity of the system but it is helpful in choosing how to use the tools for faster delivery to optimise for certain benefits like throughput, flexibility or all out speed.

- ① On the top left we can add extra cost but with no appreciable gains.
- ② On the far right extra time means costs begin to creep up.
- ③ We would argue that almost 90% of companies are on the right-hand side, chasing certainty.
- ④ Getting IT to be faster and manage costs is to optimise for throughput, but this requires great deal of up-front knowledge.
- ⑤ The choice to be Agile is to optimise for flexibility.
- ⑥ Only in the most critical and extreme circumstances would we want to use All Out Speed – it's expensive, unsustainable and risky.

*"Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage."*  
The Agile Manifesto

	Throughput	Flexibility	All Out Speed
<b>Type of customer demand</b>	High level of certainty required per batch	Complex and innovative (most software)	Critically high cost of delay
<b>Queues</b>	Control queues to increase capacity	Held outside the team/project	Hold all items which don't require all out speed
<b>Batches</b>	Small batches allow early valuable delivery	Small batches	Generally small batches
<b>Work In Progress</b>	Pull systems and WIP controls can be used to maximise throughput	Pull systems and explicit constraints based on evidence	Limited to one work item to maximise speed
<b>Excess Capacity</b>	100% capacity utilisation does not optimise throughput	Generalised specialists	Lots - need to be able to react when demand appears
<b>Overlapping Processes</b>	Breakdown of dependencies to best allocate 'known' work	High levels of overlapping processes (analysis, development, testing)	Highly desirable, almost no choice