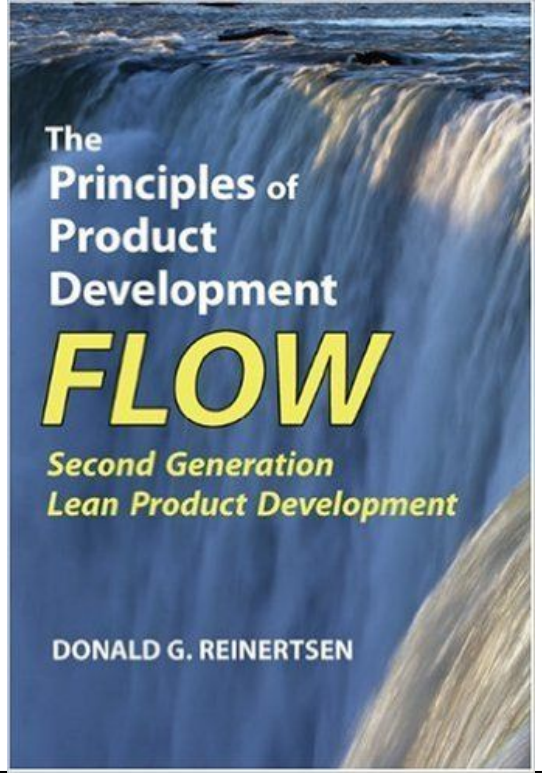


The principles of product development flow

“Second generation lean product management”

Author DonReinertsen

His latest award-winning book, *The Principles of Product Development Flow: Second Generation Lean Product Development,* has been praised as, "… **quite simply the most advanced product development book you can buy.**"

Achieving Flow by emphasising

- Small batch transfers
- Rapid feedback
- Limited work in progress (WIP)

12 critical problems with current product development

- Failure to correctly quantify economics
- Blindness to queues
- Worship of Efficiency
- Hostility to Variability
- Worship of conformance
- Institutionalization of large batch sizes
- Underutilization of Cadence
- Managing timelines instead of queues
- Absence of WIP Constraints
- Inflexibility
- Noneconomic Flow Control
- Centralized Control

Economics

COD (life-cycle profit)

When you ask them how much life-cycle profit will decrease due to a week of delay, they don't know?

Cycle time and percent value added time are only proxy variables!

The unit of measure should be **life-cycle profit impact**, which is the ultimate measure of product development success

Ask 10 people

Ask them to independently estimate what it would cost the company, in pretax profit, if their project were 60 days late to the market.

In the last 20 years the typical range of answers was 50 to 1.

We need Cost of Delay to evaluate

- The Cost of queues
- The Value of excess capacity
- The benefit of smaller batch sizes and
- The value of variability reduction

The value added by an activity is the difference in the price that an economically rational buyer would pay for a product before and after that activity is performed

We can define waste as the failure to optimize our economics

Money we've already spent is a "sunk cost" and should not enter into an economic choice.

Queues

Few developers realize that queues are the single most cause of poor product development performance?

Only 2 percent measure queues! *How big are your queues? What is the cost of these queues?* ***Only 15 percent know their cost of delay!!***

Queues are leading indicators of future cycle time problems

In product development, our greatest waste is not unproductive engineers, but work products sitting idle in process queues.

Inventory in a product development process is not physical objects but information, it is virtually invisible

Inventory is observable through its effects: increased cycle time, delayed feedback, constantly shifting priorities, and status reporting

We need to make quick, forceful interventions when queues get large

Set queue size maximums and frequently monitor queues

Economic waste by queues

- Longer cycle time
- Increased Risk
- More variability
- More overhead
- Lower quality
- Less motivation

Usually, delay cost raises linearly with queue size

Example sources of queues

- Marketing
- Analysis
- Purchasing
- Testing
- Management reviews
- Specialists
- ...

Queue size vs cycle time

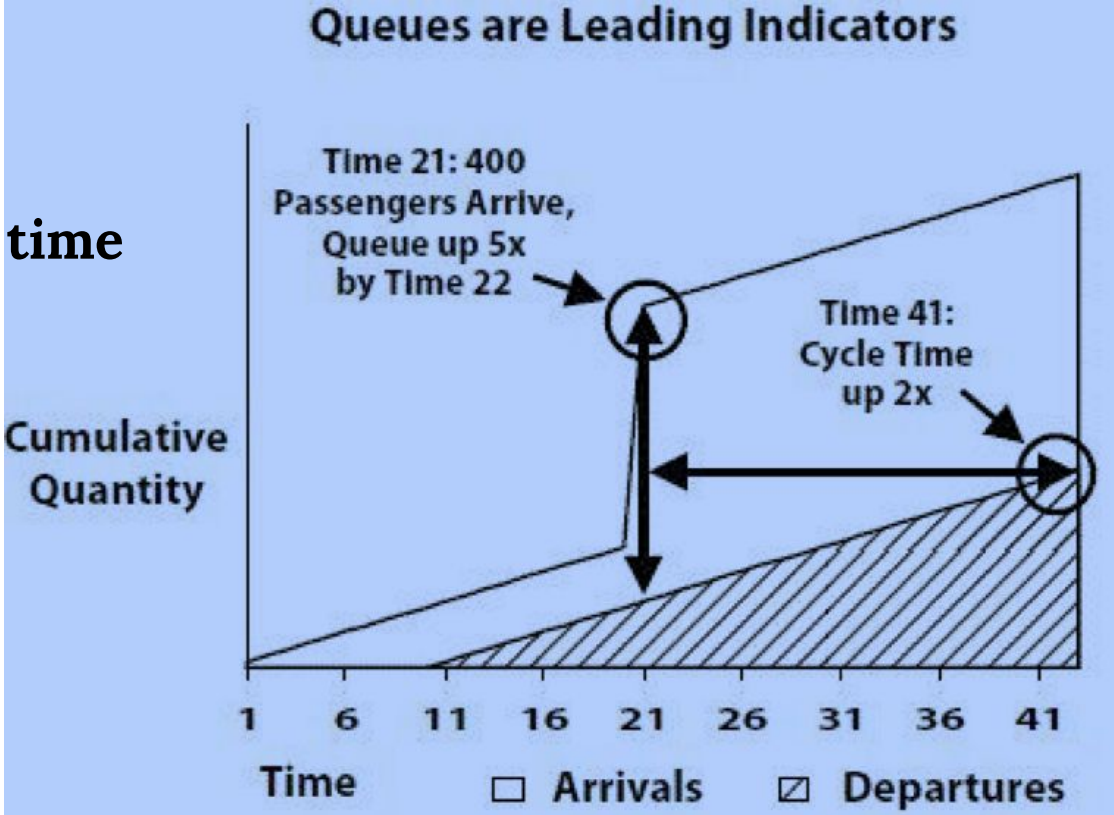


Figure 3-11 If 400 passengers arrive in an immigration area at time 21, then queue size has more than doubled by the next time period. In contrast, the first passenger to report a doubling of cycle time will leave the system at time 41. Queue size detects the problem 20 times faster than cycle time.

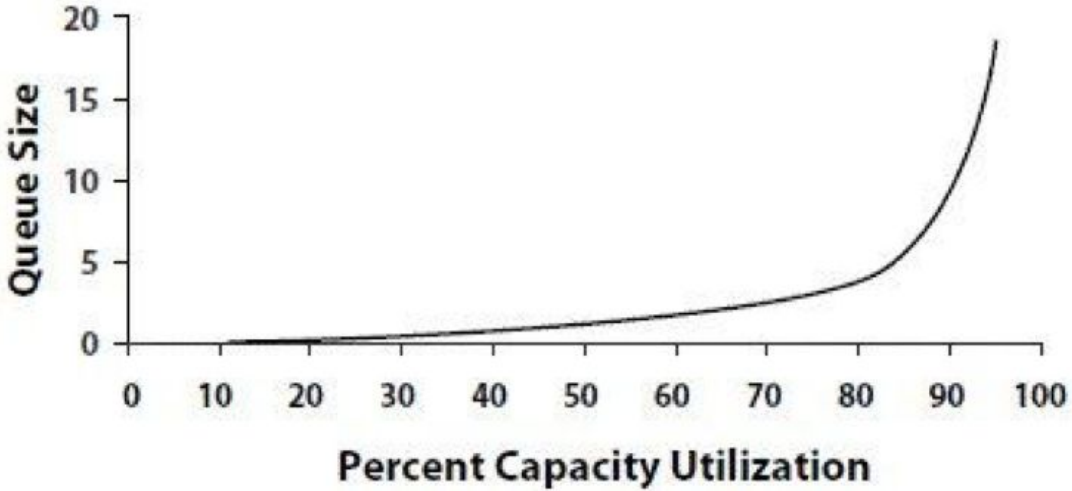
Capacity Utilization

Most of the damage done by a queue is caused by high queue states

High levels of capacity utilization are actually a primary cause of long cycle time

Increasing capacity utilization increases queues exponentially. Increasing variability increases queues linearly.

Queue size vs. capacity utilization



Note: Assumes M/M/1/∞ Queue

Figure 3-2 Queue size increases rapidly with capacity utilization.

Some calculations

https://docs.google.com/spreadsheets/d/1LBrN-P8d7TjGpH2iDlgYdjBjvDw1Ke4E6gpdantRIaE/edit#gid=0

Some diagrams

https://www.actionableagile.com/analytics-demo/

CFD

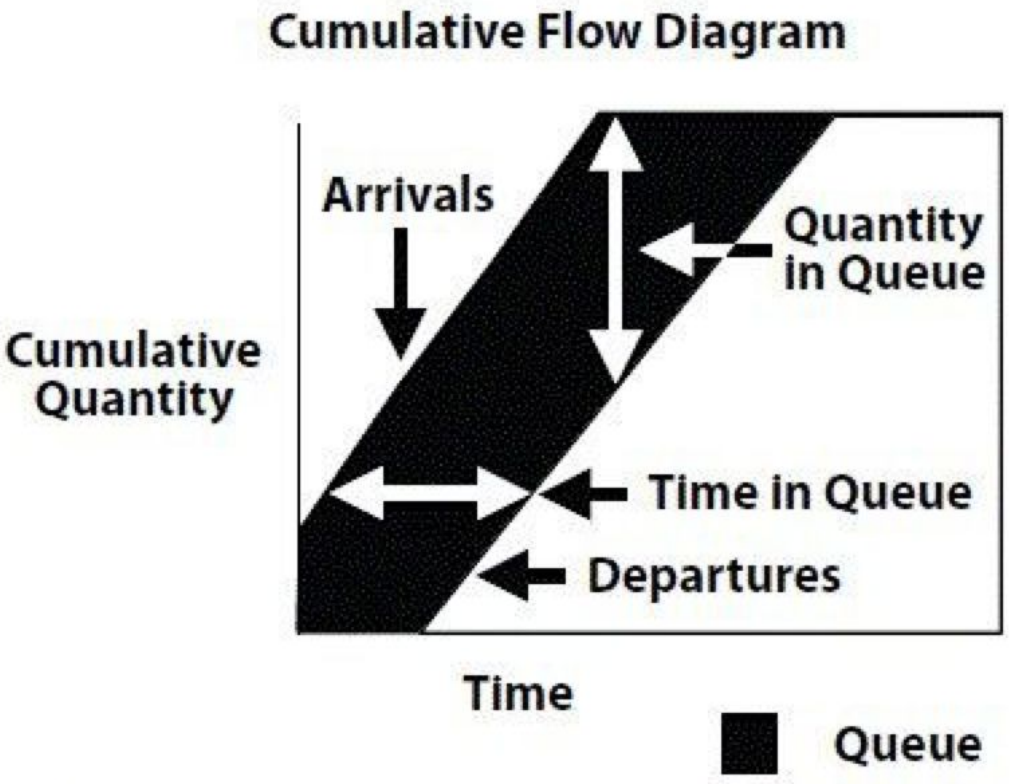


Figure 3-9 The cumulative flow diagram visually depicts what is happening to a queue over time. It is a powerful tool for understanding queueing problems and for diagnosing the causes of queues.

Variability

We cannot eliminate all variability without eliminating all value added!

Minimizing the economic impact of variability is a profound different goal than minimizing variability!

To manage product development effectively, we must recognize that valuable new information is constantly arriving throughout the development cycle!

We must learn to make good economic choices using this emerging information!

We must make resources, people and processes flexible to stay responsive in the presence of variability

But instead we work with specialized resources loaded to high levels of utilization...

Variability and rational risks

Asymmetric payoffs

- The value of a success can be much higher than the cost of a failure

Taking Rational Risks

Choice	Stakes	Payoff	Probability	EMV
A	$15,000	$100,000	50%	$35,000
B	$15,000	$20,000	90%	$3,000
C	$15,000	$16,000	100%	$1,000

EMV=Expected Monetary Value

We cannot maximize economic value by eliminating all choices with uncertain outcomes.

Figure 4-1 Expected monetary value is a function of both probability and payoff-functions. While choice C has the lowest variability, it is not the best economic choice.

Batch size

Reducing batch size is usually the single most cost effective way to reduce queues

Smaller batches have lower risk, are less expensive and produce faster results. They accelerate learning.

5–10 times improvements

Start small and start quickly!

Batch size reduction is one of the cheapest, simplest and most powerful ways to reduce variability and queues

Reducing batch sizes

- Accelerates feedback
- Reduces risk (e.g. Internet and package sizes)
- Reduces variability in flow (otherwise causing periodic overloads)
- Reduces cycle time
- Reduces overhead
- Increases efficiency
- Increase motivation and urgency
- Large batches cause exponential cost and schedule growth
- Large batches lead to even larger batches
- The entire batch is limited by its worst element
- Small batches allows finer tuning of capacity utilization

Reduce project scope

Smaller scope leads to faster development, which shortens time-horizons. Reduced scope and shorter time-horizons combine to reduce risk. Lower risk means we don't need high levels of management involved. This further reduces decision-making delays.

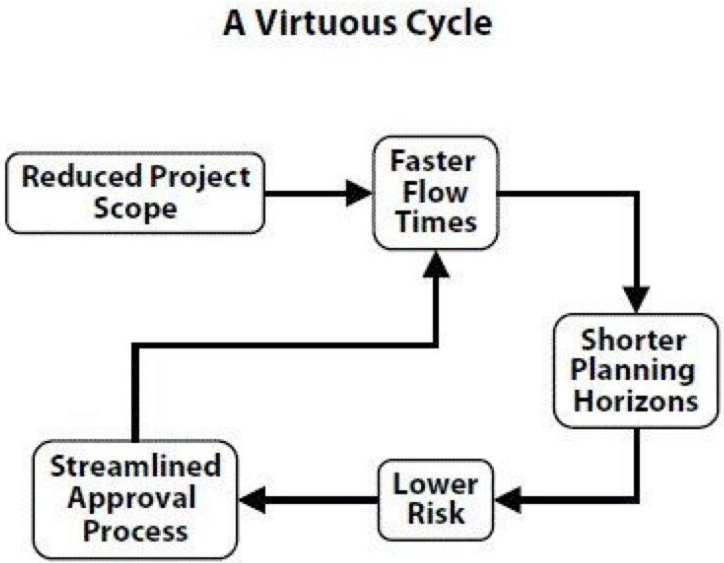


Figure 4-8 By reducing project scope, we trigger a regenerative cycle of faster cycle times and lower risk. This reduced risk leads to less need for oversight.

Batch sizes and queues - cycle time

Reduced cycle time without change to average arrival rate (demand) or average departure rate (capacity).

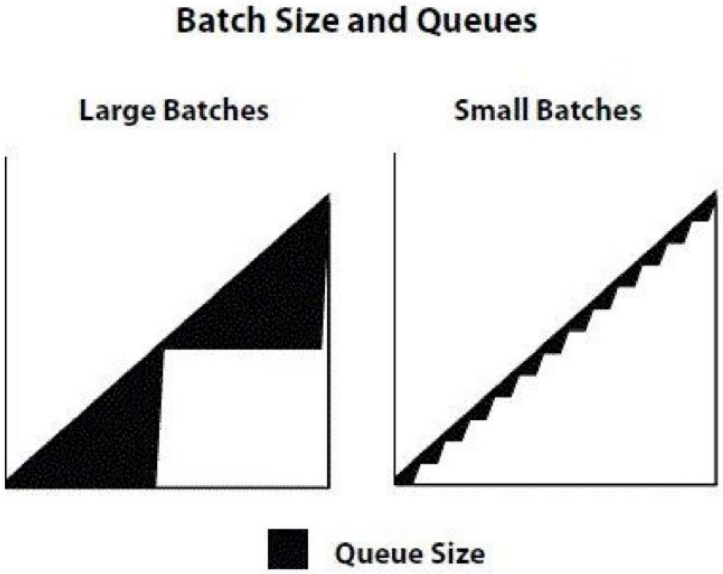


Figure 5-1 The size of the black shaded area indicates the size of the queue. Small batch sizes reduce queues.

Benefits of small batch (testing)

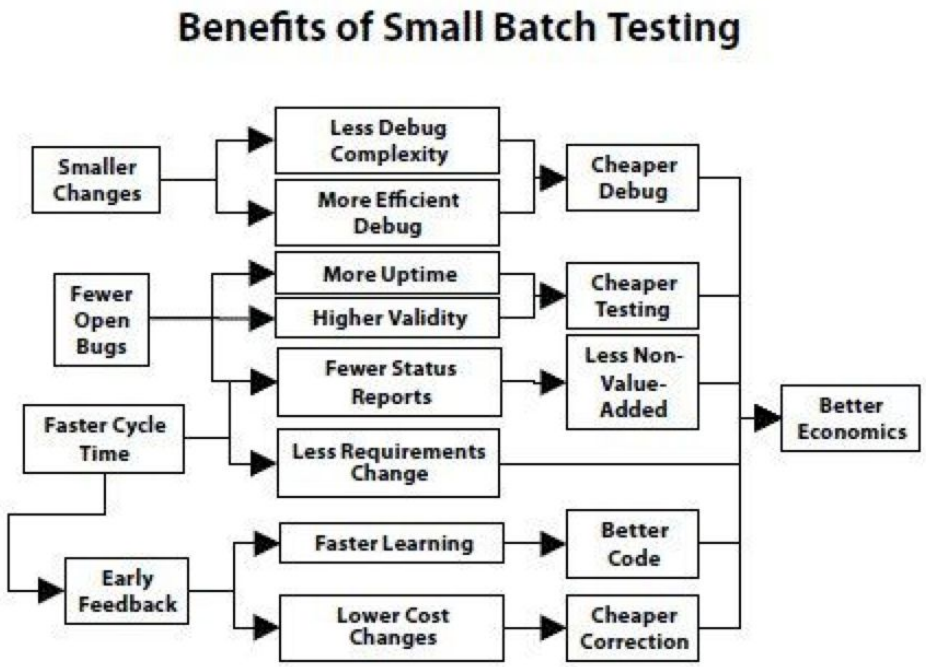


Figure 5-3 Smaller batch sizes produce a wide range of benefits. The impact on overall economics is surprisingly large.

We do not need the perfect answer for the batch size

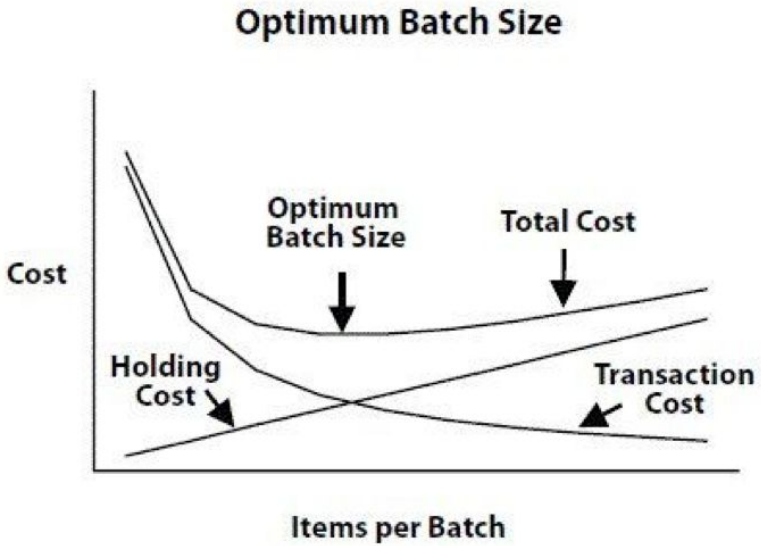


Figure 5-5 Optimum batch size is another U-curve optimization. Higher transaction costs shift optimum batch size higher. Higher holding costs shift it lower.

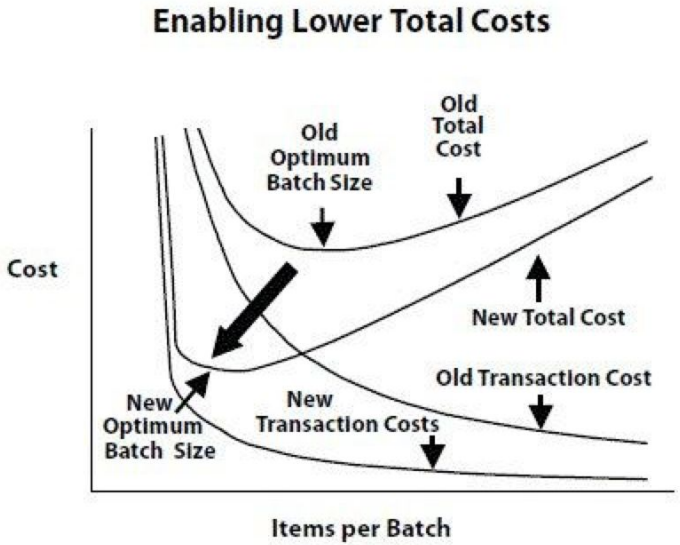


Figure 5-6 By lowering transaction costs, we create a lower optimum batch size. This also creates a new minimum process cost.

Managing batch sizes

- The most important batch is the transport batch (batch that changes the location) - allows overlapping activities and enables faster feedback
- Colocate teams to communicate in small batches
- Short run lengths reduce queues
- Good infrastructure enables small batches
- Sequence first that which adds value most cheaply
- Reduce batch size before you attack bottlenecks (as bottlenecks in product development are stochastic and physically mobile)
- Adjust batch size dynamically to respond to changing economics

Sources of batches

- Project scope
- Project funding (expensive funding rounds increase batches)
- Project phases
- Up front heavy requirements definition
- Project planning (limit detailed planning to a short time horizon)
- Testing
- Capital spending
- Design reviews
- Market research
- Project post mortems

Sequencing

Queue cost is affected by the sequence in wich we handle the jobs in the queue

The goal is to reduce the economic cost of queues, not simply to reduce the size of queues.

Shortest Job First

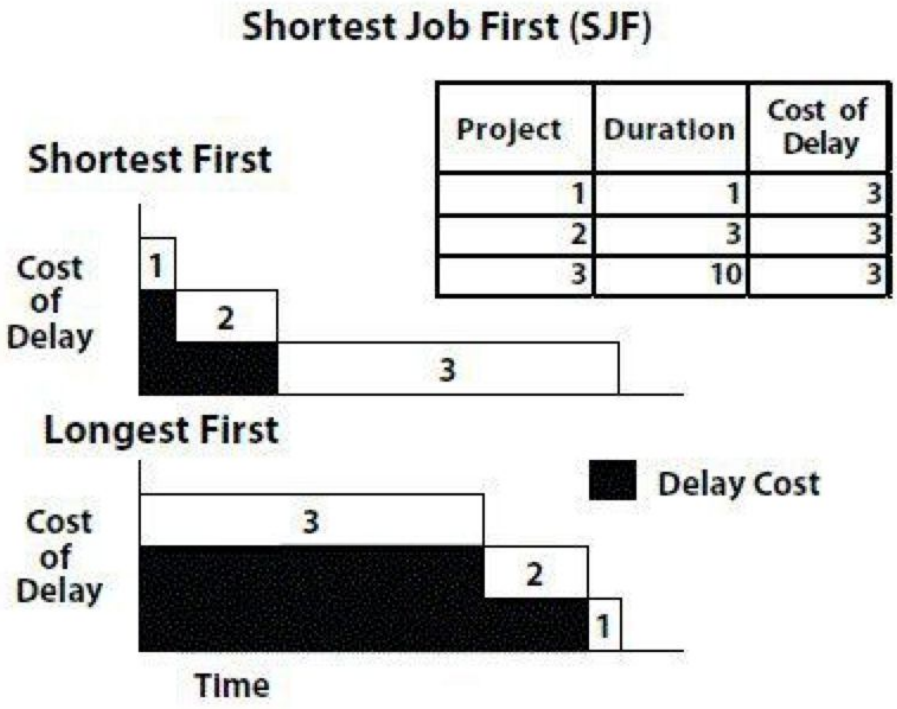


Figure 7-9 When all jobs have the same cost of delay, it is best to service the shortest job first. The black areas represent the total cost of delay associated with different sequencing strategies.

High Delay Cost First

High Delay Cost First (HDCF)

Project	Duration	Cost of Delay
1	3	10
2	3	3
3	3	1

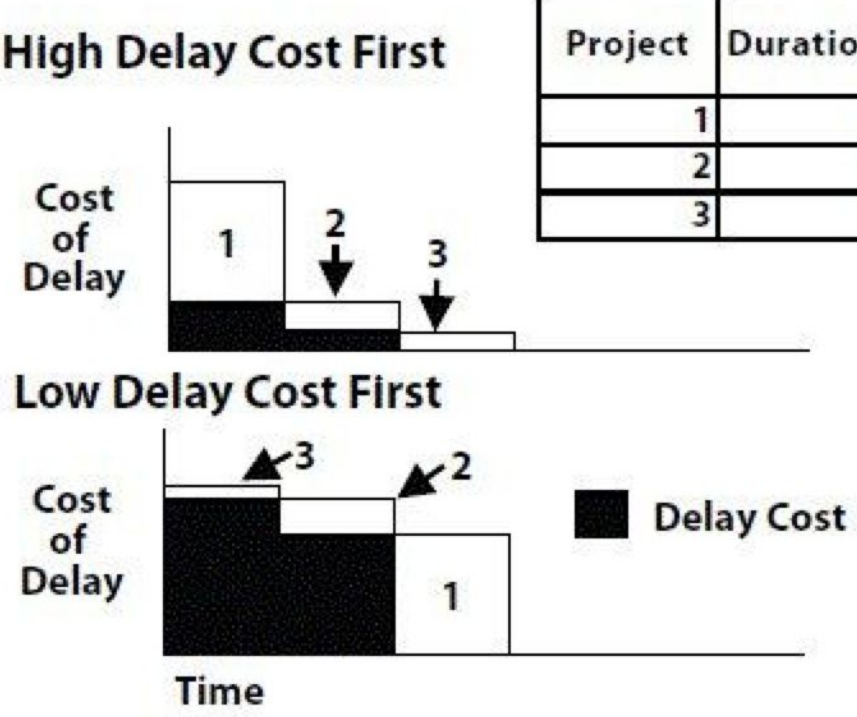


Figure 7-10 When job durations are the same, we should do the high delay cost job first. The black shaded area represents the delay cost for each sequencing strategy.

Weighted Shortest Job First

Weighted Shortest Job First (WSJF)

Project	Duration	Cost of Delay	Weight = COD/ Duration
1	1	10	10
2	3	3	1
3	10	1	0.1

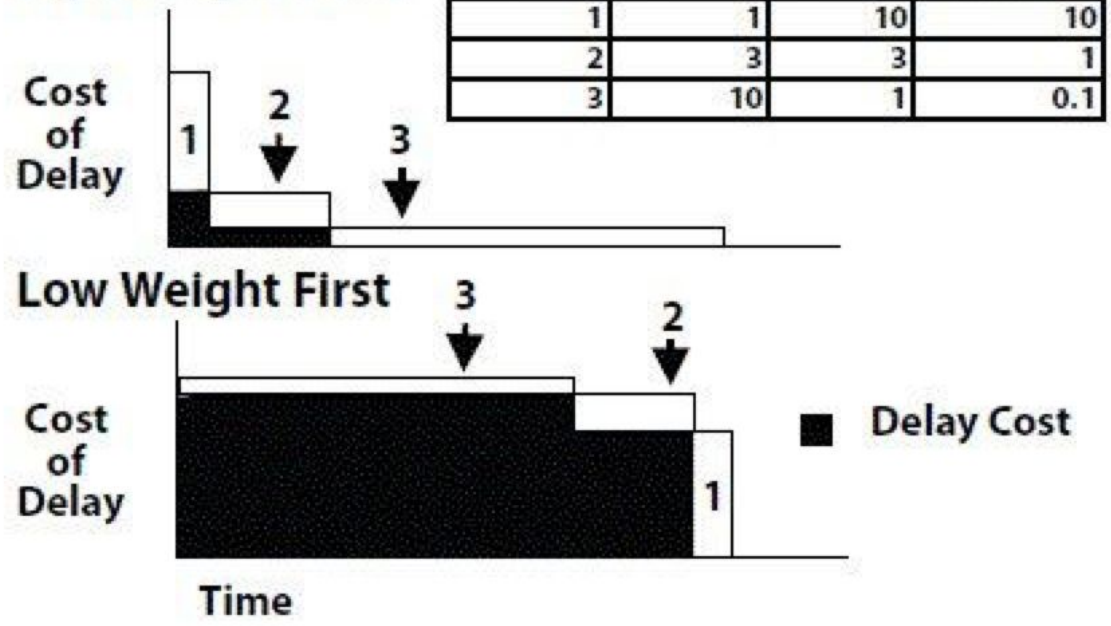


Figure 7-11 When both job durations and delay costs differ, we should prioritize by weighting the job durations. This is done by dividing the job's cost of delay by its duration.

NEXT?

WIP

Control Flow

Fast Feedback

Decentralized control