Principles of Product Development Flow

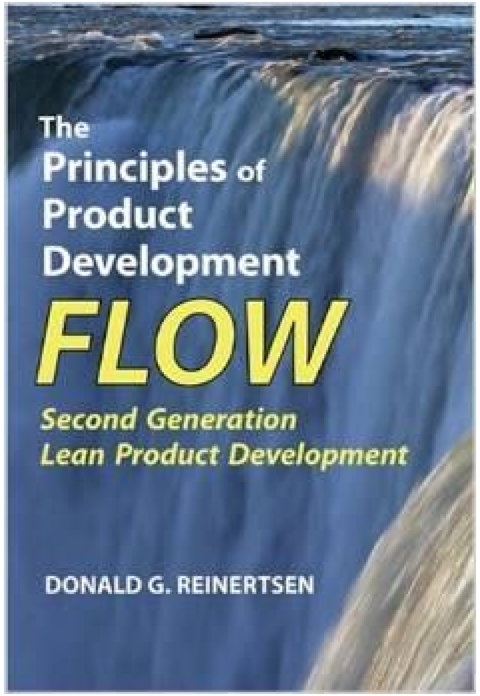
Part 3: Managing Queues

About Me

- Started programming in 1981
- Owner of Enoki Solutions Inc.
  - Consulting and Software Development
- Exposed to several industries
- Running VanDev since Oct 2010

Book:
The Principles of Product Development Flow

- ~$45 on Amazon.ca
- Published in 2009
- Award winning
- Difficult material
- Generally ignored :(
- Awesome IMNSHO



Why?

- Queues cost us money
- Queues are the root causes of many problems
- Analysing queues leads to better techniques for solving our problems

Queue Theory

- Queue Discipline (FIFO)
- M/M/1/∞ (a “normal” queue)
  - Arrival Process Type
  - Service Process Type
  - Server Count
  - Maximum Queue Size
- M - Markov

Q1 The Principle of Invisible Inventory: Product development inventory is physically and financially invisible.

- Very little of product development is visible
  - Bits on a disk are hard to see
- Hard to measure
  - Disks hold more, but are physically smaller today
  - A unit of Design In Progress (DIP) takes more space now than before
  - Measuring MiB doesn’t translate well

Q2 The Principle of Queueing Waste: Queues are the root cause of the majority of economic waste in product development.

- Queues create
  - Longer cycle time
  - Increased risk
  - More variability
  - More overhead
  - Lower Quality
  - Less motivation

Q2 - Longer Cycle Times

- It takes longer to get to the front of the line of a longer line

Q2 - Increased Risk

- Transit time for a request is longer
- During transit we are vulnerable to:
  - customers changing preference
  - competitor product introductions
  - shifts in underlying technology

Q2 - More variability

- Higher levels of utilization amplify variability (more on this later)

Q2 - More overhead

- More work in process -> more reports
- If a task is reported on weekly and takes 5 weeks to transit the queue it will be reported on 7 times (create, 1/wk, complete)

Q2 - Lower quality

- Delays in feedback magnify mistakes and entrench poor decisions

Q2 - Less motivation

- When the next process to take our work is ready for now we feed a sense of urgency
- When the next process to take our work is ready for it weeks from now we feel little value in finishing it now.

Q3 The Principle of Queueing Capacity Utilization: Capacity utilization increases queues exponentially. (For M/M/1/∞)

- ρ = Capacity Utilization
- Queue size $\propto \rho^2/(1-\rho)$
- As $\rho \rightarrow 100\%$, Queue Size $\rightarrow \infty$
- Halving excess capacity doubles queue size
  - 60% to 80% doubles
  - 80% to 90% doubles
  - 90% to 95% doubles

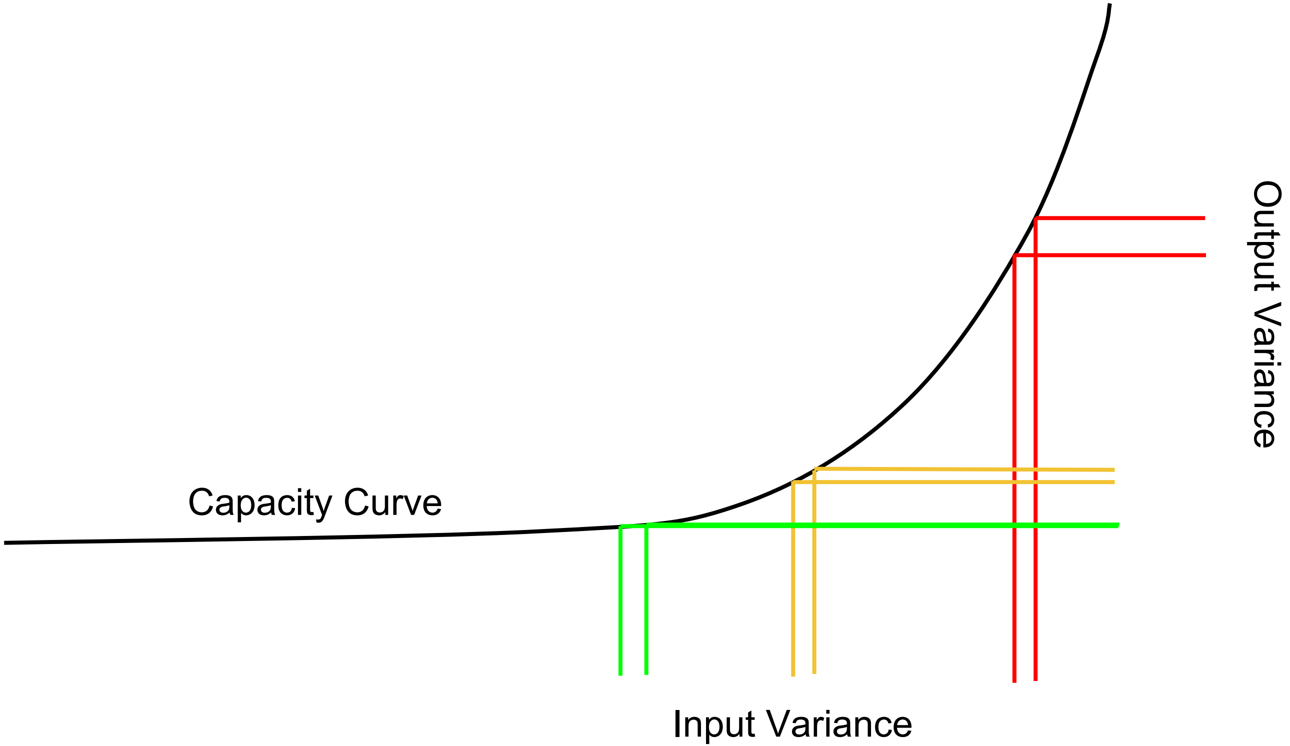
Q4 The Principle of High-Queue States: Most of the damage done by a queue is caused by high-queue states.

- If the item at the front of the queue stalls all items behind it are delayed
- Cost of Delay is magnified by Queue Size
- Ergo: We want to minimize Queue Size

Q5 The Principle of Queueing Variability: Variability increases queues linearly.

- Lots of math here
- Basically: Even if you completely remove service time variability you’d only, at best, half the queue size.
- Lots of our current process improvement attempts to reduce variability!

Q6 The Principle of Variability Amplification: Operating at high levels of capacity utilization increases variability.



Q6 The Principle of Variability Amplification: Operating at high levels of capacity utilization increases variability.

- 72.5% ± 2.5% → 1.38x Output Variance
- 92.5% ± 2.5% → 2.23x Output Variance
- 97.4% ± 2.5% → 56.52x Output Variance

Q7 The Principle of Queueing Structure: Serve pooled demand with reliable high-capacity servers.

- M/M/n/∞ is better than M/M/1/∞
  - One queue feeding multiple checkout lines goes smoother.
- M/M/n/c is even better
  - c is a WIP limit
- Each has trade-offs

Q8 The Principle of Linked Queues: Adjacent queues see arrival of service variability depending on loading.

- If queue A feeds queue B
  - The variability of the service process in A becomes the variability of the arrival process in B
- Smoothing out A smooths out B
- This is why traffic on ramps have lights
- This is also why you must consider more than the bottlenecks!

Q9 The Principle of Queue Size Optimization: Optimum queue size is an economic trade-off.

- Trade-off cost of capacity and cost of delay
- Capacity margin is a better weapon to fight variability

Q10 The Principle of Queueing Discipline: Queue cost is affected by the sequence in which we handle the jobs in the queue.

- We’ve assumed FIFO
- Priority queues by Cost of Delay help
- Making queue size small limits benefit of complex queueing disciplines
  - and limits the debate around them
- In general *simple is better*

Q11 The Subdivided Principle: Use CFDs to monitor queues.

- CFDs capture and display
  - queue size
  - average time in queue
  - arrival rate
  - departure rate
  - reason for queue size (increase in arrival vs decrease in departure)

Q12 Little’s Formula: Wait Time = Queue Size / Processing Rate.

- Use this is estimate the missing variable
  - If you know wait time and processing rate you can approximate queue size.

Q13 The First Queue Size Control Principle: Don’t control capacity utilization, control queue size.

- Stop worrying about idle workers and start limiting your queue size (WIP)
  - Supermarkets figured this out ages ago
  - If the queue for a till reaches 3 or more people they open a new till; i.e. they control queue size!

Q14 The Second Queue Size Control Principle: Don’t control cycle time, control queue size

- Cycle time is a lagging indicator
  - It requires items to exit the queue
- Queue size is immediately visible
  - Add 20 items to the queue
  - Cycle time doesn’t detect the increase until one of them exits
  - Queue size detects the increase immediately

Q15 The Diffusion Principle: Over time, queues will randomly spin seriously out of control and will remain in this state for long periods.

- We suck at understanding randomness
- Given 1000 coin tosses
  - 500 heads and 500 tails is the most likely outcome
  - The odds of that outcome: very small!
- Random Walk
  - Add 1 for heads, Subtract 1 for tails
  - Odds of crossing the zero line many times?

Q16 The Intervention Principle: We cannot rely on randomness to correct a random queue.

- If we are "up" by 10 heads it takes 10 excess tails to get back to zero
  - 1024:1 odds of that happening!
- Randomness does not cancel out.
- We must intervene to fix the problem.

Conclusions

- Queues are bad, but unavoidable
- Make queues visible
- Minimize queue size
  - CoD grows with queue size
  - Large queue size leads to: Longer cycle time, Increased risk, More variability, More overhead, Lower Quality, Less motivation
- Accept lower utilization

Q&A