STRETCHLEARN · WORKBOOK

Operations Management for Small Business — Workbook

This workbook turns the course into a working set of operations tools for one real business, yours, one you run, or one you are studying closely. Pick a single core process to use as your laboratory throughout, ideally something high-frequency that, if it ran flawlessly every time, would most improve the business. Keep your booking or sales data, a recent week of job records, and any complaint or callback log beside you. Work the sections in order: document the standard work, map and improve the flow, size capacity to your real demand and find the bottleneck, then install quality control and a weekly operating rhythm. The templates are built to be filled with your own numbers and reused.

Operations as a System and Writing SOPs That Stick

Reframe the business as a system, expose where it bleeds time and money, and document the standard work that makes results repeatable.

Exercise: Find Where Your Business Bleeds

Walk one core process in your business and hunt for the seven wastes (the DOWNTIME categories) from the course. Be specific and name the real instances, not the textbook definition.

• Defects and rework: which job or output most often has to be redone, and roughly how often does that happen?

• Waiting: where does work most often stall because the next person, part, or approval is not ready?

• Non-utilized talent: where are skilled or expensive people doing low-value work the system should handle?

• Of all the waste you listed, which single instance costs the most time or money, and would fixing it most improve the business?

Worksheet: SOP Priority Scorecard

List your recurring tasks in the chosen area and score each so you document the right ones first. Fill a row per task; the highest total scores are your first five SOPs.

Task name

Frequency score (1 rare to 5 many times a day)

Cost-of-error score (1 trivial to 5 loses a customer)

Owner-dependence score (1 anyone can do it to 5 only the owner knows)

Total priority score (sum of the three)

Owner who will write the SOP

Target draft date

Exercise: Draft One SOP from a Recording

Take your top-scored task and draft a usable SOP. Record the best performer doing it while they narrate, then write it into the eight-part structure and run the dry-run test from the course.

• Purpose in one sentence: what does this procedure produce and why does it matter?

• Write the numbered steps action-first, one instruction per line, in the exact order performed, at a level a day-one hire could follow.

• What are the specific acceptance criteria, the checkable conditions, that prove the job was done correctly?

• When a different person performed the task from your draft alone, exactly where did they hesitate or guess, and how will you fix those steps?

Checklist: Usable SOP Standards

[ ] Every SOP uses the same eight-part structure so staff know where to look
[ ] Steps are action-first, one instruction per line, with exactly one correct reading
[ ] Each SOP states its acceptance criteria so anyone can check the work is right
[ ] Every SOP carries a named owner and a last-reviewed date
[ ] A second person has performed the task from the document alone and the gaps are closed
[ ] We publish two to three SOPs a week and review a handful each month

Mapping and Improving Your Core Processes

Make the flow visible with SIPOC and swimlanes, quantify the waiting with value-stream timing, and redesign the process for speed and reliability.

Worksheet: SIPOC Frame for Your Core Process

Before mapping steps in detail, frame the whole process on one page. Fill each column for the process you chose as your laboratory.

Suppliers: who feeds this process

Inputs: what they provide to start it

Process: the five to seven high-level steps from start to finish

Outputs: what the process produces

Customers: who receives the output

Process boundaries: where it starts and where it ends

Exercise: Map the Current State Honestly

Draw the real flow as a swimlane diagram, one lane per role, and walk the actual process rather than the manual version. Then annotate it to find your targets.

• How many handoffs (arrows crossing a lane boundary) does the process actually contain, and which

handoff has no clear owner?

• Where does work loop back for rework, and what triggers the loop?

• At each step, who owns it, how long does it typically take, and how often must it be redone?

• Which three steps stand out as having no owner, long waits, or high rework, making them your improvement targets?

Worksheet: Value-Stream Timing and Efficiency

Add time data to your map to separate work from waiting. Fill these from a walk of the real process; the efficiency number is usually a surprise.

Total process time across all steps (hands-on minutes or hours)

Total lead time across all steps (elapsed time including waiting)

Process cycle efficiency = total process time / total lead time (%)

Largest single wait or queue in the process (where and how long)

Each step tagged value-adding, necessary non-value-adding, or pure waste

Estimated time or cost the largest delay adds per job

Exercise: Redesign the Process with ECRS and a PDCA Loop

Design a future state that removes the worst delays using Eliminate, Combine, Rearrange, Simplify, then plan a single tested change rather than a big-bang reorganization.

• Eliminate and Combine: which steps add no value and can be removed, and which are split across people only by accident of how the team grew?

• Rearrange and Simplify: how can you reorder so work never loops back, and which complex step can become a checklist, template, or default?

• What single change will you test first, for how long, and which baseline number (lead time, rework, cost) will tell you if it worked?

• Once a change works, how will you lock it into the SOP and retrain so it does not drift back to the old way?

Capacity Planning and Finding the Bottleneck

Size your operation to real demand with capacity and takt time, find the single constraint limiting throughput, and staff to the workload without over-promising.

Worksheet: Capacity and Takt Time Calculator

Turn your real demand into the pace your operation must hit. Fill each input and compute each output for your chosen unit and period.

Unit of output and period (e.g. jobs per day)

Actual demand for the period from sales or booking data

Available working time per period, net of breaks and setup

Takt time = available working time / demand

Your real average process time to complete one unit

Theoretical capacity (units the period could produce nonstop)

Utilization factor from observation (effective is often 70 to 85% of theoretical)

Effective capacity = theoretical capacity x utilization factor

Exercise: Find and Exploit Your Constraint

Apply the five focusing steps from the theory of constraints to your process. The cheap steps come first; only spend money at the very end.

• Identify: at which step does work pile up in front while the next step starves, making it the constraint?

• Exploit (no spending): how can you stop the constraint sitting idle, stop it doing work a cheaper step could handle, and ensure it never waits?

• Subordinate: how will you pace the rest of the process and size a small buffer to keep the constraint fully fed but not buried?

• Elevate and repeat: if still needed, what investment would add capacity there, and where do you expect the next bottleneck to appear once this one breaks?

Worksheet: Staffing-to-Workload Calculator

Replace headcount guesswork with arithmetic. Fill a block per major step to see where you are genuinely short, where you have slack, and where the real problem is process speed.

Step name

Process time per unit (minutes)

Units demanded per period

Work content per period = process time x units (hours)

Productive hours available per person per period

Required staff = work content / available hours per person

Current staff on this step

Gap and read (short on people, slack, or a process-speed problem)

Checklist: Demand-Matching Standards

[ ] I plan bookings and commitments against effective capacity, not the theoretical number
[ ] I have plotted the shape of demand by hour, day, or season
[ ] Core staff is sized to baseline demand, with flexible capacity layered on for peaks
[ ] Heaviest coverage is scheduled onto known demand peaks, not spread evenly
[ ] I plan productive capacity around 80 to 85% so slack absorbs surprises
[ ] Each period I compare committed versus delivered jobs and feed it back into the plan

Quality Control and the Operating Rhythm

Define what good looks like and catch defects at the source, fix root causes permanently, and install the cadence and KPIs that keep the system running without you.

Worksheet: Acceptance Criteria and Quality Checks

Make quality a yes-or-no check rather than an opinion. Fill this for one key output so the standard and the checks are explicit.

Output being defined

Acceptance criteria: the specific, checkable conditions that make it correct

Poka-yoke: how the common error could be designed to be hard to make

In-process inspection point: where a defect is cheapest to catch and fix

Current first-pass yield = jobs right first time / total jobs (%)

Cost of one defect that reaches the customer (labour, materials, reputation)

Exercise: Run a Root-Cause Analysis on a Recurring Problem

Take a problem that keeps coming back and find its real cause with the Five Whys, then design a countermeasure that changes the system rather than exhorting people.

• State the problem specifically: what goes wrong, where, and how often?

• Run the Five Whys: ask why of each answer until you reach a cause you can actually change. What is that root cause?

• Which fishbone category does the root cause fall under, Method, Machine, Material, People, Measurement, or Environment, and does the data confirm it?

• What countermeasure removes the root cause (make the error impossible or obvious, or build a check into the process), and how and when will you verify the problem actually stopped?

Worksheet: Operations KPI Dashboard Setup

Choose the vital few metrics across the four levers and set where, how often, and by whom they are reviewed. Fill a row per KPI.

KPI name

Lever it reflects (process, capacity, quality, or cost)

How it is calculated

Current baseline value

Target value

Review cadence (daily, weekly, or monthly) and owner

Checklist: Operating Rhythm Standards

[ ] A brief daily huddle covers today's flow, blockers, and what needs attention
[ ] A weekly operations review looks at the KPI trend and open improvement experiments
[ ] A monthly review steps back to capacity, demand shifts, and the next constraint
[ ] We track a vital few KPIs across process, capacity, quality, and cost where the team can see them
[ ] Every meeting has a fixed agenda and is short, ideally standing
[ ] Improvements that work are written into the SOP and the metric is watched to confirm they hold

Your Action Plan

1. Choose one high-frequency core process as your laboratory and pull the data, booking or sales records, a week of job records, and any callback log.
2. Walk that process and list its seven wastes, then name the single instance that costs the most time or money.
3. Score your recurring tasks on frequency, cost-of-error, and owner-dependence, and draft your top SOP from a narrated recording, closing the dry-run gaps.
4. Frame the process with SIPOC and draw the real current-state swimlane map, annotating owner, time, and rework at each step.
5. Add process time and lead time to the map, calculate process cycle efficiency, and identify the largest queue to attack.
6. Redesign the flow with Eliminate, Combine, Rearrange, Simplify, then run one tested PDCA change against a baseline number and lock the win into the SOP.
7. Calculate demand, takt time, and effective capacity, then identify your constraint and exploit it before spending any money.
8. Build the staffing-to-workload numbers and set a schedule that puts heaviest coverage on demand peaks at an 80 to 85% utilization target.
9. Define acceptance criteria for a key output, add a poka-yoke and an in-process check, and start tracking first-pass yield.
10. Run a Five Whys on one recurring problem, fix the root cause in the system, and install a daily huddle, weekly review, and the vital-few KPI dashboard.