Bottleneck	
Definition	The bottleneck is the step with the lowest capacity (or highest time per unit) that limits overall throughput. A bottleneck is any step in a process whose limited capacity, long cycle time, or dependency constraints restrict the overall throughput of the entire process. A process can only move as fast as its slowest step.
How to identify	1. **Queue build-up:** where work piles up. Long idle periods, repeated activities 2. **Longest cycle time step:** step taking most time per unit. step dominates total process time. 3. **Highest utilization:** step always busy while others idle. 4. **Data approach:** timestamp each stage and compute average waiting + service time. 5. **Waiting time spikes:** large gaps between activities (seen in timestamps / process mining).
Traditional	Observation, interviews, manual timing
Examples	**1:** if verification team can process 30/day but approval team processes 10/day, approvals are the bottleneck. Training verification team to do 60/day won't increase completed applications. **2:** if requisitions wait 3 days for approval but only 30 minutes for processing → approval is the bottleneck, not procurement staff efficiency. **3:** approval bottlenecks – long idle time. Event data shows prolonged waiting time before approval steps, indicating approval-based bottlenecks. **4:** rework bottlenecks – repeated steps

Capacity bottleneck	**Time bottleneck**	**Resource bottleneck**	**Policy bottleneck**	**Information bottleneck**
Occurs when demand > processing ability.	Occurs when a step takes significantly longer than others.	Occurs due to lack of people, machines, or skills.	Occurs due to rules or approvals.	Occurs when data is unavailable, duplicated, or delayed.
One approval manager handling 200 requests/day while others handle 500.	Heating water in the tea process delays everything else.	Only one trained staff member can operate a machine.	"all requests above x must be approved by director."	Invoice cannot be processed because purchase order data is missing.

Why "improving non-bottleneck doesn't help"?
Because system output is capped by the slowest step. Improving a faster step only increases wip before bottleneck. Because system throughput is constrained by the bottleneck. Improving faster steps only increases work-in-progress before the bottleneck without increasing final output.

What is a bottleneck and how do you identify it?
A bottleneck is the step that limits overall process output due to lowest capacity or highest time per unit. It can be identified by where queues build up, which step has the highest utilization, which stage has the longest service time, and by analyzing timestamps to find where most waiting occurs. A bottleneck is the step with the lowest capacity / highest delay that limits overall throughput (the "constraint"). You identify it by: (1) longest queue/wait time in the map, (2) highest cycle-time contribution, (3) frequent rework loops, (4) resource overutilization (one role overloaded). Example: in procurement, approvals or invoice matching often become bottlenecks because work pauses until someone signs off or reconciles mismatched documents.

Define a bottleneck and explain its impact on throughput.
A bottleneck is the step in a process with the lowest capacity, which limits the overall output of the entire system. No matter how efficient other steps are, throughput cannot exceed the speed of the bottleneck. Bottlenecks cause queues, delays, higher cycle time, and increased costs, as work accumulates before the constrained step. For example, if tea preparation takes 2 minutes but water heating takes 6 minutes, the heater becomes the bottleneck and limits service speed. In bpr, identifying bottlenecks is critical because improving non-bottleneck steps does not increase overall performance.

Identify different types of bottlenecks with examples.
Bottlenecks can be resource-based, process-based, policy-based, or demand-based. Resource bottlenecks occur when limited equipment or staff restrict flow, such as one cashier handling all payments. Process bottlenecks arise from poorly designed sequences, such as unnecessary approvals. Policy bottlenecks result from rules like "manager approval required for every order," even for low-value cases. Demand bottlenecks occur when demand exceeds system capacity, such as peak-hour rushes in cafeterias. Bpr focuses on eliminating or redesigning these constraints rather than merely speeding up surrounding steps.

Identify likely bottlenecks in the longer case and justify using "waiting time" logic.
The longer case shows large delays due to handoffs and approvals, plus waiting around delivery/invoice stages. Bottlenecks are likely: approval queues (manager/finance), reconciliation/invoice confirmation, and supplier coordination. Waiting-time logic: even if active work per step is short, the case duration grows because it sits idle between departments. High handoffs increase coordination delay risk and create stop-start flow, which is classic procurement bottleneck behavior.

Broken processes

Definition	A broken process is not one that is slow once in a while — it is one that systematically fails to meet performance expectations in cost, time, quality, or consistency.
Examples	A process is broken not because people are slow, but because the structure forces waiting, rework, and unnecessary control.

Structural causes	**Policy causes**	**Information causes**	**Capacity causes**	**Behavioral causes**
Too many handoffs, approvals, silos	Outdated rules ("manager must approve everything")	Duplicate data entry, missing data	Bottlenecks, resource overload	Unclear ownership, incentives misaligned

Business process	
Definition	A process is a well-defined, tangible, measurable steps producing consistent output. A business process is a **structured set of activities** that transforms **inputs** into **outputs** to deliver value to a customer (internal or external). Processes are repeatable and cross functional. A "department task" is not a process. A process usually crosses departments.
Examples	**1:** order-to-cash (customer order → payment received). **2:** procure-to-pay (need identified → vendor paid).

Define a business process and give two examples.

A business process is a repeatable set of activities that converts inputs into outputs to deliver value to a customer. Examples include order-to-cash (order to payment) and procure-to-pay (request to vendor payment).

Define a process. Differentiate task vs process.

A process is a structured set of related activities that together transform inputs into outputs to deliver value to a customer. A task is a single unit of work, while a process consists of multiple interconnected tasks working together toward an outcome. Tasks are isolated actions (e.g., "verify invoice"), whereas processes are end-to-end flows (e.g., "procure-to-pay"). A task focuses on what is done, while a process focuses on how value is delivered. Managing tasks improves local efficiency, but managing processes improves overall performance and customer satisfaction.

How to know a process is redundant?

A process becomes redundant when:

- the original constraint no longer exists (technology/society changed)
- the process exists mainly for control/verification that can be achieved differently now
- you see repeated "checking/matching/handovers" because data isn't trusted

Example (Ford accounts payable – lecture reference)

Old: match invoice + PO + receiving documents (tons of mismatches). Reengineered: vendor is paid based on receipt + database match; remove invoice handling. Tech didn't "speed up paperwork" — it removed the need for paperwork.

Business process reengineering (bpr)		
Definition	Business process reengineering is the fundamental rethinking and radical redesign of business processes to achieve dramatic improvements in critical performance measures such as cost, quality, service, and speed. This definition has four key words that matter: fundamental, radical, dramatic, process. Targets when active time < 20% and waiting time > 80%.	
Fundamental	Fundamental means questioning basic assumptions: • Why do we do this step? • Who should do it? • Does it need to exist at all? This is not about improving the step — it is about challenging its existence.	"why do we need invoices?" Instead of "how do we process invoices faster?"
Radical	Radical means root-level change, not small improvements. • Radical ≠ risky for fun • Radical = redesign from scratch	Replacing an approval-based process with rule-based automatic decisions.
Dramatic	Bpr targets order-of-magnitude improvements, not marginal gains. • Not 5–10% better • But 50–80% improvement	Ford reduced accounts payable staff by ~75% after reengineering.
Process	Bpr focuses on end-to-end processes, not departments, tasks, or job titles.	"procure-to-pay" instead of "accounts payable department".
Examples	Ford accounts payable: eliminated invoices altogether. Starbucks mobile ordering: customers perform tasks previously done by staff, reducing wait time and bottlenecks. A procurement process takes 13 days: active work: 2 days, waiting: 11 days. Redesign focus = waiting, not working faster.	

Define bpr and explain its key elements.

Business process reengineering (bpr) is the fundamental rethinking and radical redesign of end-to-end business processes to achieve dramatic improvements in performance such as cost, quality, speed, and customer satisfaction. Its four core elements are: fundamental, meaning questioning why a process exists; radical, meaning starting from a clean slate rather than incremental change; dramatic, aiming for breakthrough improvements rather than small gains; and

process-focused, emphasizing end-to-end workflows instead of individual tasks or departments. Together, these elements distinguish bpr from traditional improvement approaches.

Why must diagnosis precede redesign in bpr?

Diagnosis must precede redesign because you cannot fix what you do not understand. Without proper diagnosis, organizations risk automating or redesigning flawed processes, which only accelerates inefficiency. Bpr emphasizes understanding bottlenecks, root causes, waste, and value flow before proposing solutions. For example, automating invoice processing without questioning why invoices exist may lock inefficiencies into technology. Diagnosis ensures redesign targets the real problems, aligns with customer value, and produces dramatic improvement rather than cosmetic change.

Why does bpr focus more on waiting time than active time?

Because waiting time contributes the majority of lead time without adding value. Reducing waiting through structural redesign yields far greater improvements than speeding up already efficient tasks.

BPR Impact			
Definition	The social, political, economic, and ethical consequences of radical process redesign on employees, management, and organizational structure.		
Social impact	**Political impact**	**Economic impact**	**Ethical considerations**
Job displacement Skill mismatch Stress and uncertainty	Power shifts Loss of managerial authority Internal conflicts	Cost reduction through automation, role elimination Productivity gains and faster cycle times Short-term implementation costs (it, training, restructuring) Risk of service disruption during transition	Fair treatment of employees Transparent decision-making Responsible automation
Examples	Replacing agents with kiosks: improves efficiency ; raises employment and ethical concerns ; requires retraining or redeployment plans. Ford's accounts payable bpr reduced staff by ~75%, lowering operational costs while increasing processing speed.		

Why must organizations consider social and political impacts when implementing bpr?

Because radical process changes can disrupt jobs, power structures, and morale, leading to resistance or backlash if ethical and social concerns are ignored.

Explain the social and ethical impacts of bpr.

Bpr can lead to job displacement, skill mismatches, and increased stress due to role changes and automation. Ethically, organizations must ensure fair treatment of employees through transparent decision-making, retraining, and redeployment. Ignoring social and ethical impacts can cause resistance, low morale, and reputational damage. Responsible bpr balances efficiency gains with employee welfare to ensure sustainable transformation.

Explain social and political implications of bpr.

Bpr has significant social and political implications because it alters jobs, authority, and power structures. Automation may cause job displacement and skill mismatch, increasing stress and insecurity. Politically, managers may lose control, and internal conflicts may arise. Ethical concerns include fair treatment, transparency, and responsible automation. For example, replacing agents with kiosks improves efficiency but requires retraining and redeployment to avoid backlash and morale loss.

Why must organizations consider political impact when implementing bpr?

Bpr alters power structures by removing managerial layers and decision authority. This can lead to internal conflicts, resistance from middle management, and political pushback. If political impacts are ignored, redesign initiatives may be blocked or reversed. Considering political dynamics allows leaders to manage resistance, realign roles, and maintain organizational stability during transformation.

Discuss the broader impacts of bpr.

Bpr impacts organizations beyond efficiency gains. It reshapes culture, roles, governance, and customer experience. Successfully implemented bpr improves speed, quality, and cost while empowering employees through ownership and decision-making. However, it also introduces disruption, requiring ethical consideration and change management. Strategically, bpr enables competitiveness and agility. Societally, it influences employment patterns and skill requirements. Its impact is transformative, not merely operational.

Mitigation strategies	Retraining and redeployment programs. Transparent communication about change. Phased implementation to reduce shock. Ethical guidelines for automation decisions.

State 4 social/political implications of radical bpr and how to reduce resistance.

Radical bpr can cause job redesign/reduction, loss of power (middle managers/approvers), fear and uncertainty, and union/employee pushback; it can also shift accountability and expose poor performance. Your notes explicitly link deeper redesign with stronger political/social impact. Reduce resistance through: top leadership sponsorship, clear communication of "why," employee involvement, training/reskilling, and transparent metrics so change feels fair rather than political.

BPR Resistance	
Definition	Resistance to change is the natural human reaction to perceived threats caused by organizational change. Resistance is normal, not irrational. Resistance is rational behavior when employees perceive risk to job security, status, or competence.
Reasons	Employees identify strongly with their job titles and expertise. Bpr collapses roles (e.g., "case manager"), threatening identity. **Example**: an accounts clerk resists bpr because "invoice matching" defined their expertise. When invoices are eliminated, their professional value feels erased.
	Some employees benefit from process complexity (gatekeepers, approvers). Bpr removes informal influence, not just formal authority. **Example**: a senior clerk delays files to gain importance. End-to-end digital workflow removes their leverage.

Fear of job loss	**Loss of power and control**	**Fear of incompetence**	**Cultural inertia**	**Uncertainty and ambiguity**	**Loss of identity**	**Loss of informal power**	**Trust deficit**
Automation Role elimination Downsizing (common in bpr)	Fewer approvals Flattened hierarchy Decision automation	New technology New skills required Unlearning old methods	"this is how we do things here" Comfort with routine Habitual workflows	Unclear future roles Incomplete communication Lack of trust	Employees define themselves by specialized tasks eliminated by bpr	Individuals lose influence gained through approvals and bottlenecks	Employees don't believe management's promises
Ford reduced accounts payable staff by ~75%.	Managers often resist because: "if the process decides, what is my role?"	People fear being exposed as "not good enough".	"we've always done it this way."	"what happens to me after this?"	Invoice clerks lose role when invoices are removed	Gatekeepers removed by straight-through processing	"they say no layoffs now, but later they will"

Identify and explain sources of resistance in bpr initiatives.

Resistance arises due to fear of job loss, loss of authority, skill gaps, cultural inertia, and uncertainty about future roles. These concerns make employees defensive even when redesign improves efficiency. It arises from fear of job loss, loss of power, skill gaps, cultural inertia, uncertainty, and loss of identity. Employees may resist automation that eliminates roles, while managers resist flattened hierarchies that reduce authority. Informal power holders (gatekeepers) oppose streamlined workflows. Poor communication and lack of trust intensify resistance. For example, invoice clerks resist invoice-less processing because it removes their expertise. Resistance is rational, rooted in perceived threats, not emotional weakness.

Why is resistance to bpr considered rational rather than emotional?

Resistance to bpr is rational because it stems from real threats perceived by employees, such as job loss, loss of authority, skill obsolescence, and uncertainty about future roles. Bpr often eliminates tasks, layers, and positions, directly affecting livelihoods and professional identity. Employees act defensively to protect themselves, not to oppose efficiency. Recognizing resistance as rational allows management to address concerns through communication, retraining, and role clarity rather than dismissing resistance as negativity.

Explain cultural inertia as a source of resistance with an example.

Cultural inertia refers to the tendency of organizations to continue established routines simply because "this is how we do things." Long-standing habits, norms, and informal rules create comfort and predictability. Bpr challenges these norms by introducing radical change, which feels disruptive. For example, employees may resist self-service systems because they are accustomed to paper approvals. Cultural inertia slows adoption even when redesign clearly improves performance.

How does fear of incompetence contribute to resistance in bpr?

Fear of incompetence arises when bpr introduces new technologies, roles, or skills that employees are unfamiliar with. Workers worry about being unable to perform, losing confidence, or being judged as inadequate. For example, staff accustomed to manual processing may resist automation because it exposes skill gaps. This fear leads to avoidance, passive resistance, or rejection of change unless proper training and support are provided.

How can management reduce resistance to bpr?

Management can reduce resistance by transparent communication, involving employees early, providing retraining, and clearly defining future roles. Explaining the purpose of change, showing personal benefits, and offering skill development reduces fear. Assigning process ownership and creating trust through consistency also help. Resistance decreases when employees feel respected, informed, and prepared for redesigned roles.

Why do bpr initiatives fail?
Bpr initiatives fail when organizations focus on technology instead of people, ignore culture, underestimate resistance, or lack leadership support. Common failures include technology-driven redesign, poor communication, absence of performance metrics, and resistance from middle management. Automating a bad process only makes inefficiency faster. Without change management and employee buy-in, redesigned processes collapse. Failure is rarely technical; it is organizational and behavioral.

Case	
Definition	A case is one complete execution of a process from start to end. One end-to-end flow = one **case**.
Examples	One purchase request, one tea order, one student registration
It has	case id, start/end timestamps, activities executed

Change management	
Background	Bpr is not just process change — it is organizational change. When you redesign a process, you also: change roles, remove jobs or authority, shift decision power, alter identity and routines. This makes bpr emotionally and politically disruptive.
Definition	Change management is the structured approach used to help individuals, teams, and organizations transition from the current state (as-is) to the future state (to-be) while minimizing resistance and performance loss.
Bpr	Why bpr needs stronger change management than incremental change: bpr is radical, bpr breaks long-standing rules, bpr removes middle layers, bpr questions "how we've always done things"

Why is change management more critical in bpr than in incremental improvement initiatives?
Because bpr introduces radical, organization-wide changes that disrupt existing roles, authority, and routines. Without managing human reactions and resistance, even technically sound redesigns can fail.

Why is change management critical in bpr?
Change management is critical in bpr because reengineering introduces radical changes to processes, roles, and power structures. Without managing human reactions, even technically sound redesigns fail. Employees fear job loss, loss of authority, and skill obsolescence, leading to resistance. Effective change management uses communication, participation, training, and leadership support to reduce uncertainty and build trust. For example, retraining staff before automation reduces fear and increases acceptance. Bpr succeeds only when behavioral change accompanies process change.

Constants		
Definition	things that must remain true (core purpose), while everything else is negotiable.	
BPR	BPR is hard because humans cling to old process logic. Real redesign starts when you say: "What must stay?", "What can be removed?"	
Examples	Boarding pass problem • Purpose: verify right passenger enters plane • Old assumption: paper boarding pass required • New: digital verification / biometrics / QR	Baggage • Purpose: enforce weight allowance • Old: agent weighs and checks • New: self-service kiosk shifts responsibility (outsourcing labor)

Cost–benefit / cost–complexity matrix		
Definition	**Low cost, high benefit** → priority, best candidates, quick wins, should be prioritized immediately **High cost, high benefit** → strategic decision, strategic initiatives, large bpr projects, require senior management approval	**Low cost, low benefit** → minor fixes, small improvements, acceptable to do, no strategic impact **High cost, low benefit** → avoid, generally avoided. Poor use of organizational resources
Key point	Not every process improvement is worth doing. Big ideas without measurable benefit are **waste**, not innovation.	

Explain the cost–benefit (or cost–complexity) matrix and how it guides priorities.
Your lecture method: ceo sets a target (e.g., 5% cost reduction) then departments inventory processes and estimate savings, but choose only what is worth the added complexity. A cost–benefit / cost–complexity matrix prioritizes: low cost + high benefit = do first; high cost + low benefit = avoid; high-high = strategic projects; low-low = optional quick wins. This avoids wasting time training people for tiny gains.

How would you defend a redesign proposal to a ceo ?
I'd present: (1) current kpi baseline (cost/txn, cycle time, defect rate), (2) root cause and bottleneck evidence, (3) projected kpi uplift with assumptions, (4) implementation cost (capex/opex), (5) complexity/training needs, (6) risks

(social/political + operational), and (7) payback/roi timeline. This aligns with the "data-driven + business perspective" requirement and avoids proposing technology for its own sake.

Estimating cost reduction potential (but with caution)

For each identified process, departments estimate: how much cost reduction is possible? How much time saving can be achieved? However, a very important bpr principle applies here: just because improvement is possible does not mean it should be done. Why? Because every improvement introduces complexity, such as: training employees, changing roles and responsibilities, time required to implement change, resistance to change. So bpr is not about "fix everything," but about fixing the right things.

Understanding complexity in process improvement

complexity refers to: number of people affected, amount of training required, time needed for change, organizational disruption. A process with high improvement potential but very high complexity may not be worth touching immediately. This is where bpr differs from automation: automation blindly improves tasks, bpr evaluates impact vs effort.

Cost vs time trade-off (decision logic)

Organizations analyze improvement options using cost and time trade-offs:

- High cost, low time → worth doing (quick big wins)
- Low cost, long time → often inefficient, may be skipped
- High time + high effort → needs strong justification

This prevents wasting resources on low-impact changes and aligns effort with value creation

Data driven		
Definition	A data-driven approach in bpr means using actual process data (time, frequency, cost, variation) to: • Identify inefficiencies, • Locate bottlenecks, • Justify redesign decisions, • And evaluate improvements objectively.	Data-driven ≠ technology-driven. Technology is only a tool; data is the basis for decision-making. Processes don't fail the same way every time. Variation across cases is often the biggest problem.
	Understand the problem, measure it, then decide whether technology is needed	

Why is data-driven analysis essential in bpr?

Because bpr involves radical change with high cost and risk. Data-driven analysis ensures redesign decisions are based on actual performance evidence rather than assumptions, reducing failure and resistance.

What does data-driven bpr mean?

Data-driven bpr means redesigning business processes based on factual evidence rather than intuition or hierarchy. Decisions are guided by measurable data such as cycle time, waiting time, defect rates, costs, and customer outcomes. Instead of assuming how a process works, organizations analyze real process behavior using logs, metrics, and performance data. This reduces bias, improves justification for change, and lowers redesign risk. Data-driven bpr ensures that radical changes target actual problems and deliver measurable business value rather than cosmetic or politically motivated improvements.

"just because we can optimize a process doesn't mean we should." Explain with a business example.

Optimization adds implementation cost, training time, risk, and change resistance—so not every 1% gain is worth it. automating a rarely-used internal approval workflow may cost months of development and training, while saving only a few minutes per week. A better decision is to focus on high-volume processes (customer onboarding, billing) where savings multiply. This is a merit-function mindset: pursue changes that materially improve kpi targets and profitability.

Why is "data-driven measurement" essential before redesign (tie to kpis)?

Bpr is risky and expensive; measurement prevents redesign based on opinions. You first quantify baseline kpis: cycle time, defect rate, cost per case, rework %, and waiting time. Then you can prove the bottleneck and estimate savings. This matches your sir's emphasis: "processes, data, reengineering, how to measure." Data also helps defend choices politically ("we changed this because it causes 60% of delay").

Data-driven process engineering

A process can be redesigned by tracing data dependencies instead of departments: the process moves forward only when the required data for the next step exists. This exposes hidden rules (e.g., approvals, checks) as "functions" we optimize. In bpr, data-driven design reduces duplication (same data entered twice), removes unnecessary validations, and makes performance measurable through timestamps and kpis.

Aspect	Opinion-driven redesign	Data-driven redesign
Basis of decisions	Decisions are based on intuition, experience, or seniority, often without factual validation.	Decisions are based on measured evidence, process data, and performance metrics.
Bias & reliability	High risk of personal bias and assumptions, leading to subjective solutions.	Low bias because conclusions are supported by data and analysis.
Risk & outcomes	Higher risk of redesign failure since root causes may be misidentified.	Controlled risk because changes target verified root causes.

Example	"customers complain, so hire more staff immediately."	"data shows 80% of delays occur at approval; adding staff elsewhere won't help."
	"the manager feels this step is unnecessary, so remove it."	"process metrics show this step adds no value and increases cycle time by 30%."

Differentiate opinion-driven and data-driven redesign.
Opinion-driven redesign relies on intuition, seniority, or anecdotal complaints, making it vulnerable to bias and weak justification. Solutions often treat symptoms rather than root causes, leading to high failure risk. Data-driven redesign, in contrast, uses objective evidence such as process metrics, logs, and statistical analysis. It identifies where delays, defects, or waste actually occur, enabling targeted improvements. Data-driven approaches achieve stronger buy-in, clearer accountability, and more reliable outcomes because decisions are supported by measurable facts rather than personal beliefs.

Time data	Frequency data	Cost data	Variation data	Path data
Start time End time Waiting time Cycle time	How often steps occur How often rework happens How often approvals repeat	Labor cost Cost per transaction Cost of delay	Differences across cases Standard deviation Outliers	Which steps occur In what sequence Which steps are skipped or repeated

Why is technology-driven redesign dangerous?
Technology-driven redesign is dangerous because it automates existing inefficiencies instead of questioning whether steps should exist at all. This reinforces outdated rules, increases complexity, and locks in poor process design. As hammer argues, computers amplify flawed processes when used without rethinking work. Technology should enable redesigned processes, not dictate them. Redesign must start with process logic and customer outcomes, not software features.

Explain data-driven decision making from a ceo perspective.
From a ceo perspective, data-driven decision making ensures strategic choices are based on measurable impact rather than assumptions. It enables leaders to allocate resources effectively, reduce risk, and justify transformation initiatives with evidence. Ceos use data to assess customer value, operational efficiency, and financial outcomes, ensuring accountability. This approach improves credibility, accelerates execution, and aligns process redesign with business performance goals such as cost reduction, speed, quality, and growth.

Digital Tech				
	Outsource	In-house	SLA	CDO
Definition	external vendor builds/maintains digital process	company builds its own digital capability	service-level agreement (guarantees uptime/quality)	Chief Digital Officer (responsible for digital transformation)
BPR	"Tech augments your work."			
Examples	Bank saying: "We're an IT company with a banking license." i.e: digital *is* the business, not a side project.			

Division of labour (adam smith pin factory)	
Definition	Division of labour means breaking one job into smaller, specialized tasks so each worker repeats a narrow set of actions. This increases output because workers become faster, switching time reduces, and tools/skills become optimized.
Bpr	It created the idea of a process: work is not "one person does everything" but a structured sequence of steps. However, division of labour also created functional silos, which later become a major source of inefficiency (handoffs, approvals, rework). Division of labour optimizes local tasks; bpr optimizes end-to-end outcomes.
Examples	**1:** one person makes a whole pin slowly vs 10 people each doing one step → total pins skyrocket. **2:** in a bank: one team collects documents, another verifies, another approves, another disburses. Efficiency in each team can exist, but the full customer experience can still be slow due to handoffs.

Why did division of labour increase productivity, and how can it create inefficiency later?
Division of labour increases productivity by specialization, reduced switching time, and standardized tools/skills, so output per hour rises. Over time it can create inefficiency because work is split across departments; the process gains handoffs, waiting time, approvals, and rework. This makes the end-to-end customer experience slow even if each department is efficient.

Bpr initiatives usually start at the top management level. For example, a ceo might say: "next year, we need a 5% cost reduction." This target becomes a strategic goal, not an operational suggestion. Every department is now responsible for aligning itself with this objective. This top-down target setting is important because: it gives direction, it prevents random or isolated improvements, it ensures all departments work toward a common goal

Explain division of labour and its impact on modern organizations.

Division of labour, introduced by adam smith, refers to breaking work into specialized tasks assigned to different workers to improve efficiency and productivity. In modern organizations, this led to functional departments such as hr, finance, and operations. While specialization increases skill efficiency and output, it also creates silos where employees focus only on their own tasks rather than the entire process. This often results in coordination delays, handoffs, and lack of accountability for final outcomes. Thus, division of labour improves productivity but can reduce end-to-end process effectiveness.

End-to-end process thinking	
Definition	End-to-end thinking means evaluating the process as a full chain from trigger to outcome, not isolated steps.
Examples	**1:** student registration end-to-end: admission → fee payment → course enrollment → id card. **2:** banking end-to-end: customer opens account → kyc → verification → activation.

Why is end-to-end thinking essential for bpr?
Because bpr aims for dramatic improvements in overall performance. If you optimize isolated tasks, bottlenecks and handoffs remain and the customer experience stays slow. End-to-end thinking reveals the true delays, rework, and ownership gaps.

Why is end-to-end thinking central to bpr?
End-to-end thinking is central to bpr because it focuses on the complete flow of value from customer request to delivery, rather than isolated tasks or departments. Bpr seeks dramatic performance improvement, which cannot be achieved by optimizing individual steps alone. Viewing the process holistically reveals bottlenecks, redundancies, and non-value-adding activities that are invisible within silos. It also enables clear ownership and accountability. Without end-to-end thinking, organizations risk improving parts of the process while overall performance remains poor.

Explain "end-to-end" thinking with an example.
End-to-end thinking views the process across functions (handoffs), from trigger to outcome. Example: "student fee payment" is not only "accounts office work"; it starts from invoice generation → student notification → payment → verification → receipt → record update. End-to-end exposes delays, rework, and unclear ownership at handoffs, which functional silos hide.

Event		
Definition	An **event** is a recorded activity with: • Case id (process instance) • Activity name • Timestamp • Resource (optional)	**Example (procurement):** • Create purchase requisition • Analyze requisition • Request quotation • Approve quotation
	Each execution = **one case**.	

Factors – critical success						
Definition	**Critical success factors (csfs)** are conditions and practices that must be present for bpr to achieve sustainable and effective transformation.					
Top management commitment	**Clear vision and communication**	**Process ownership**	**Employee involvement**	**Training and skill development**	**Performance measurement and kpis**	**Data-driven justification**
Without senior leadership support, bpr collapses under resistance.	People must understand: Why change is happening What will change How it affects them	Clear accountability across departments.	Involving staff reduces fear and increases acceptance.	Prepares employees for new roles.	Defines success objectively (cycle time, cost, defects, customer satisfaction)	Reduces emotional resistance and political pushback.

Prevents symbolic or cosmetic redesign. Enables post-bpr monitoring and accountability. **Example:** after redesign, ford measured invoice cycle time and staff count to validate improvement.

Identify critical success factors for bpr implementation.
Successful bpr requires strong leadership commitment, clear communication, process ownership, employee involvement, training, and data-driven justification to overcome resistance and ensure sustainable change. Key success factors for bpr include top management commitment, clear vision and communication, process ownership, employee involvement, training, and data-driven justification. Leadership must sponsor change and remove barriers. Employees must understand why change is needed and how it affects them. Process owners ensure accountability across departments. Training prepares staff for new roles. Data reduces political resistance. Together, these factors convert resistance into acceptance and ensure sustainable transformation.

Why is top management commitment critical in bpr?

Top management commitment is critical because bpr challenges existing power structures, roles, and routines. Without senior leadership support, resistance from middle management and employees can derail redesign efforts. Leaders provide authority, resources, and legitimacy to the change, ensuring alignment across departments. Their visible commitment signals that bpr is a strategic priority rather than a temporary initiative, increasing acceptance and follow-through.

Why is data-driven justification essential for bpr success?

Data-driven justification is essential because it grounds redesign decisions in objective evidence rather than opinion. It reduces emotional resistance, weakens political opposition, and builds credibility for radical change. By using metrics such as cycle time, cost, and defect rates, organizations can clearly demonstrate why redesign is necessary and how improvements will be measured, increasing stakeholder buy-in.

Factors – failures			
Technology-driven redesign Lack of change management Ignoring culture	Poor communication No performance measurement Resistance from middle management	Middle management resistance: middle managers lose power when layers and approvals are removed. Fear of loss of authority and relevance	Blocks redesign despite top management support Managers resist automated approvals because decision authority shifts to the process.
Lack of top management commitment Automating instead of redesigning ("technology-driven") Ignoring social and political impact	Poor diagnosis before redesign Unrealistic expectations or timelines Resistance due to fear of job loss Weak communication and training	Most bpr failures are not technical failures, but leadership and change-management failures.	

Key warnings: "don't obliterate driven by technology." Technology **enables**, it does not **justify** redesign.

Why do many bpr initiatives fail despite sound technical design?

Because organizations underestimate human resistance, cultural inertia, and political dynamics, focusing on technology rather than managing behavioral and organizational change.

If this csf is missing...	**This failure occurs...**
Top management commitment	Redesign stalls under resistance
Clear communication	Fear, rumors, and mistrust
Process ownership	Fragmented accountability
Employee involvement	Active resistance
Training	Fear of incompetence
Data-driven justification	Political pushback
Performance metrics	No proof of success

Failure Modes and Effects Analysis				
Definition	FMEA is a structured, step-by-step method to identify: 1. Failure modes 2. Effects 3. Causes & controls		FMEA helps you predict "where can this process break?", "what damage does it cause?", and "what should we fix first before redesigning/automating?"	
Failure modes	ways a step/component/process can fail. The specific failure (e.g., "wrong course eligibility shown", "kiosk printer jam", "payment not recorded").			
Effect	what happens if it fails (impact on customer/business/process). The consequence (e.g., student enrolls wrongly → timetable mess; customer stuck → queue increases).			
Cause	Root reason (e.g., outdated prerequisite data; poor maintenance). why it fails.			
Risk priority	Which failure is most dangerous/likely/hard to detect → fix first. Then you prioritize which failures to fix first, usually by scoring risk (Severity, Occurrence, Detection).			
Current controls	What currently reduces risk (e.g., manual verification, validation rules, logs). what prevents/detects it			
Scoring	S = Severity: How bad is the effect if it happens? O = Occurrence: How often will it happen? D = Detection: How likely you are to catch it before it harms the customer?		Most FMEA uses three scores (1–10). Then: RPN = S × O × D. Higher RPN = higher priority to fix.	
Example	Process step: "Order is paid → customer waits → barista starts later"			
	Failure modes	Effects	Causes	Controls
	- Barista starts late (delay) - Order slip lost / name mismatch - Drink made but customer not notified	- More waiting time (merit function worse) - Customer dissatisfaction (CSAT down) - Queue builds → throughput down	- No real-time order queue visibility - Manual handoff between cashier and barista - Peak rush variability	- Digital order screen showing queue & status - Unique order ID + display board - Auto time-stamps on each step

	You redesign the process to reduce waiting time and variance, but FMEA tells you where the redesign might fail (e.g., screen fails, IDs mismatch, barista ignores queue). So you add controls.

BPR		
	1) BPR is radical change → risk increases	2) FMEA prevents "rework"
	When you "obliterate not automate," you remove steps and responsibilities shift. That creates new failure modes. FMEA is how you anticipate them.	every failure is a rework (repeat process = cost). FMEA reduces failures early → less rework → better cost + productivity.
	3) FMEA supports Six Sigma thinking	4) FMEA is data-driven process engineering
	Six Sigma wants low variation and near-zero defects. FMEA helps you find defect opportunities and reduce them before measuring and controlling.	process = rules on data. FMEA asks: what data/rule failures cause breakdown? (missing prerequisite data, wrong validation, delayed timestamp, sensor errors).
	When you redesign, you might remove steps but also introduce new risks (kiosk down, app crash). FMEA helps you make BPR safe and defensible.	

Fishbone (ishikawa) diagram					
Definition	A fishbone diagram visually categorizes possible causes of a problem to ensure systematic cause identification. It answers: "what could possibly be causing this problem?". (some models also add management or policy.)				
Benefits	Prevents tunnel vision. Encourages team thinking. Supports six sigma measurement later. Converts intuition into structure. It prevents lazy answers like "staff are slow" by forcing systematic cause exploration.				
People	Process	Machines	Materials	Measurements	Environment
Skills, training, coordination, motivation	Sequence, handoffs, rework, rules	Equipment, tools, technology	Inputs, availability, quality	Metrics, data accuracy, kpis	Layout, space, noise, physical constraints
Examples	Problem: Tea delivery delay. "tea takes too long". Causes could be: cups missing (materials), cashier bottleneck (process/people), kettle heating time (machine). Then redesign focuses on the real constraint.				
Staff coordination, speed	Receipt → payment → service order	Slow water heater	Missing cups/spoons	No tracking of service time	Crowded counter

How does a fishbone diagram support process improvement?
It organizes potential causes into structured categories, ensuring that people, process, technology, and environment factors are all examined before selecting improvement actions.

Describe a fishbone diagram and its categories.
A fishbone diagram, also called an ishikawa diagram, is a visual tool used to systematically explore potential causes of a problem. The problem is placed at the "head," while major cause categories form the "bones." Common categories include people, process, machines, materials, measurements, and environment. Each category is expanded with specific causes. For example, late tea delivery may involve slow staff (people), heater issues (machines), poor layout (environment), or missing supplies (materials). Fishbone diagrams help teams see the full picture instead of blaming a single factor.

Using the tea workflow, give 3 root causes and map them to fishbone categories.
From your notes: machine: water heater slow/not ready (creates delay). Materials: missing cups/spoons/sugar/whitener increases time searching. People/process: unclear roles or poor coordination at counter; strict sequence (order→receipt→pay→serve) causes waiting/rework. Fishbone helps categorize causes so improvements are targeted: fix equipment capacity, ensure inventory readiness, and redesign roles/sequence to reduce handoffs and waiting.

Five whys technique	
Definition	The **5 whys** is a questioning method that repeatedly asks "why?" To uncover the root cause of a problem. Problem → why? → why? → why? → why? → why? You stop when: the cause is actionable and further "why" adds no value
Strengths	Simple. Fast. No data tools needed
Limitations	Depends on honesty. Can oversimplify. Not suitable for complex systems alone
Examples	Problem: order delayed. Why? Approval slow. Why? Manager unavailable. Why? Only one approver. Why? Policy requires senior approval. → root cause: policy design

When is 5 whys insufficient?
When problems have multiple interacting causes, in which case tools like fishbone or data analysis are needed.

Flowcharts	
Definition	A flowchart is a visual representation of a process that shows the sequence of activities, decisions, inputs, and outputs using standardized symbols. A flowchart is the visual model of steps + decisions.
Symbols	**Oval (terminator)** → start / end of the process **Rectangle (process)** → an activity or task. Example: "verify documents"

	Diamond (decision) → yes/no or branching logic. Example: "documents complete?" **Parallelogram (input/output)** → data entering or leaving the process. **Arrow (flow line)** → direction of process flow
Strengths	Simple and intuitive ; good for teaching and documentation ; easy to draw quickly in exams ; shows logical order clearly
Limitations	Does **not show who** performs each step ; does not highlight handoffs ; becomes messy for complex processes ; limited for cross-functional analysis. Flowcharts can also become messy for complex processes with many exceptions. They show what happens but not always where time is lost unless extra timing data is added.
What is a flowchart and why is it used in bpr? A flowchart is used to visualize the sequence of steps in a process, helping identify inefficiencies, unnecessary decisions, rework loops, and delays. In bpr, it provides a clear "as-is" understanding before redesign.	
Identify two weaknesses of flowcharts in process analysis. Flowcharts do not show ownership or responsibility and cannot clearly capture handoffs between departments, which limits their usefulness for cross-functional redesign.	

Hammer & champy's philosophy of bpr		
Core belief	Michael hammer and james champy are considered the founders of bpr. Their core belief was that "most organizations are not badly automated — they are badly designed." Meaning: technology is not the problem - old process logic is the problem	
Key ideas	Traditional processes are designed for: • Control • Scarce information • Low-skilled labor • Slow communication	Modern organizations operate in: • Information abundance • Skilled workforce • Fast communication • High customer expectations
Implicit rules	Many processes follow unwritten rules, such as: "we pay only after receiving an invoice", "decisions must go through managers", "customers cannot be trusted to do tasks", bpr breaks these rules.	
Examples	Old rule: "we pay vendors after invoice verification." New rule (ford): "we pay vendors when goods are received."	
According to hammer, why do organizations fail to gain value from it investments? Because organizations use it to automate existing processes instead of redesigning them. This preserves old inefficiencies and only speeds up flawed workflows rather than eliminating unnecessary steps.		
Why does hammer argue that automation often fails? Hammer argues that automation often fails because organizations automate existing inefficient processes instead of redesigning them. Technology is applied to speed up approvals, data entry, and handoffs without questioning whether those steps are needed at all. As a result, inefficiencies, delays, and errors are preserved and even amplified. Automation improves execution speed but does not fix poor process design. Hammer emphasizes that real performance problems stem from outdated rules, fragmented workflows, and functional silos, which cannot be solved by technology alone without fundamental process change.		
Explain how implicit rules hinder process performance. Implicit rules are unwritten assumptions about how work should be done, often inherited from past practices. Examples include rules like "we pay only after receiving an invoice" or "only managers can approve decisions." These rules hinder performance because they are rarely questioned, even when technology and skills have changed. They add unnecessary steps, approvals, and delays, increasing cost and cycle time. Bpr seeks to surface and challenge these hidden rules, replacing them with process designs that reflect current capabilities and customer needs.		

Hammer's paper - **"don't automate, obliterate"**			
Core message	Do not use computers to speed up bad processes. Use computers to **eliminate** bad processes.		
Automate	Taking an existing process, adding it to make it faster, keeping the same logic	Scanning paper forms instead of removing the form.	
	Why it fails? Automation preserves bureaucracy, speeds up errors and locks in bad design. Hammer argues that many it failures occur because companies paved cow paths instead of redesigning roads.		
Obliterate	Question why the process exists, remove unnecessary steps, redesign the flow entirely	Eliminating paper forms using shared databases.	
Example - ford accounts payable	Old process: • Purchase order • Receiving document • Invoice • Three-way matching	New process: • Shared database • Goods receipt triggers payment • No invoice	Result: • 75% reduction in staff • Faster processing • Fewer errors

	• Payment		

Explain "don't automate, obliterate" with an example.

"don't automate, obliterate" means that organizations should eliminate unnecessary processes rather than automate them. Hammer argues that many processes exist only because of old constraints, such as paper-based systems. For example, in ford's accounts payable process, instead of automating invoice matching, ford eliminated invoices entirely by using a shared database and paying suppliers when goods were received. This removed reconciliation, reduced staff requirements, and cut cycle time dramatically. The example shows that redesigning the process yields far greater benefits than automating flawed workflows. It means: don't use technology to speed up a bad process; redesign the process itself, then use tech to enable the new design. Example: if a customer must fill a form, get signatures, and then staff re-enters the same data, "automating" just makes the old steps faster. "obliterate" means remove unnecessary steps (single data capture, fewer handoffs), and then use a shared system so verification happens once and everyone sees it. This matches your sir's warning: don't be "technology-driven" without redesign logic.

Heterogeneous measures (queue time vs CSAT)				
Definition	different nature / different units / different meaning. When you evaluate a process, you might measure:			
	Queue time	Queue length	Cost	CSAT (customer satisfaction)
	minutes (objective, physical)	number of people (objective, physical)	rupees (objective, physical)	rating like 1–5 (looks numeric, but it's based on human opinion)
Why CSAT is "measurable but subjective"	Measurable Because you can assign a number: • "Rate your experience 1 to 5" • "Would you recommend us? 0 to 10" (NPS) • "Satisfied / Neutral / Unsatisfied" (converted to numbers)	Subjective Because the rule inside people's head is different: • One person gives 3/5 if wait is 10 minutes • Another gives 5/5 if staff was kind even with 10 minutes wait • Someone else gives 1/5 because the place was confusing		
	So the measurement output is numeric, but the criteria producing that number is personal. That's exactly what your note says "CSAT measurable, but criteria subjective."			
Example	Starbucks Queue time low ✅ But CSAT low ❌ because customers feel confused (no clear pickup flow). So BPR must include both redesign flow (signage, pickup station, order visibility) not just "increase speed"	NADRA Wait time might improve. But CSAT could drop if staff behavior is rude or instructions unclear. So optimizing only time can still produce a "worse" system overall.		
"Heterogeneous measures are metrics of different types and units (e.g., minutes vs satisfaction ratings). Queue time is objective, while CSAT is numeric but subjective because it's based on human judgment. In BPR we cannot combine them directly; we must normalize them to a common scale and apply weights in a merit function to reflect business priorities. Otherwise we get misleading conclusions."				

The "exam trap": why you can't just add them directly

Imagine you create a merit function like: F = queue_time + CSAT. If queue_time is in minutes and CSAT is 1–5, this is nonsense because:

- 12 minutes + 4.2 stars = *what even is the unit?*
- Also one minute change is huge compared to 0.1 star change, so the formula becomes biased.

Concrete example

Process A:	Process B:	Which is better? If you add raw:
• wait time = 2 min • CSAT = 2.0/5	• wait time = 10 min • CSAT = 4.8/5	• A score = 2 + 2.0 = 4 • B score = 10 + 4.8 = 14.8

You'd conclude A is "better" just because it has lower minutes — but customers are clearly **unhappier** in A. So the trap is: raw addition hides the real tradeoff.

What you're supposed to do instead (in simple words)

Step 1: Put them on the same scale (normalize)

You convert every measure into a comparable scale like 0 to 1 (or 0 to 100). Example:

If wait time ranges 0–20 minutes:

- normalized_wait = wait_time / 20. So 10 minutes becomes 0.5

If CSAT ranges 1–5 and higher is better:

- normalized_csat = (CSAT − 1) / 4. So 4.2 becomes (4.2−1)/4 = 0.8

Now both are "unitless" and comparable.

Step 2: Decide importance (weights)

Because not all measures matter equally. The Merit function might be: $F = w_1(normalized\_wait) + w_2(1 - normalized\_csat)$

Notice we use (1 − csat) because higher CSAT is good, but in a "cost function" lower is better and w_1, w_2 are weights based on business priority. If your goal is "speed matters more than vibe": $w_1 = 0.7$, $w_2 = 0.3$. If your goal is "experience matters more" (like Starbucks premium brand) $w_1 = 0.4$, $w_2 = 0.6$. That's why your teacher talked about: "Is it possible we assign 1 weight to one and 100 to another?". Yes — because weights represent priority.

Idea	
Definition	An idea is a mental model of reality — it's defined by the attributes/elements that make it what it is.
BPR	BPR is full of "ideas" that people memorize but don't apply: "Don't automate, obliterate", "Optimize a merit function", "Processes = rules on data". If you truly understand them, you can look at any messy process (UMS enrollment, Starbucks, NADRA, airport) and rebuild it logically.
Examples	If someone "knows" BPR but still suggests "add one more counter" for every problem, they don't understand the idea. That's a local fix, not reengineering.

Industrial revolution → modern process structure	
Definition	Industrial revolution introduced mechanization, standardization, and large-scale production. Work became repeatable, measurable, and organized into steps.
What it produced in organizations	Standard operating procedures (sops), quality checkpoints, batch processing, hierarchical control structures
Examples	**1:** assembly line manufacturing: steps are sequenced, time-studied, and controlled. **2:** university admissions: many standardized forms, fixed approval steps, defined roles.

Innovation and financial benefit	
Definition	Innovation is not always financially beneficial at the start. Many successful companies do not constantly reinvent everything. Instead, they make incremental improvements, optimize existing processes, introduce controlled innovation.
BPR	This aligns with bpr's idea of strategic change, not reckless innovation
Examples	Apple rarely makes radical changes, it refines existing products gradually.

Merit function	
Definition	A merit function is the score/metric we use to judge whether a process is "better". In bpr we don't improve randomly; we optimize a measurable objective like time, cost, quality, cash flow, customer value, or risk. It is objective function combining measures (often weighted) to evaluate process performance. A merit function is a formula that summarizes how "good" a process is, using measures (cost, time, productivity, satisfaction).
Weighted	If steps are $s_1 \rightarrow s_2 \rightarrow s_3 \ldots$ Each has a measure $m_1, m_2, \ldots$ Then the merit function: $F = \sum(w_i m_i)$. Weights $w_i$represent importance of each step. weights can be 1 vs 100 — because some steps dominate the experience/cost. General form: $F = w_1 m_1 + w_2 m_2 + \cdots$ Where: $m_i$are measures (wait time, cost, defects, CSAT penalty) and $w_i$are weights (importance) **Example:** Starbucks could use: m_1= waiting time, m_2= time inside café, m_3= number of staff (cost proxy)
What is a "merit function" in process improvement? Give 2 examples. A merit function is a scoring/decision criterion used to judge whether a process change is "worth it" using measurable business outcomes (money/time/risk). Your class framing was: use data, tighten kpis, and evaluate improvements financially/operationally. Examples: (1) cost per transaction (rs/order), (2) cycle time (minutes per customer), (3) defect rate (% late deliveries), (4) customer satisfaction (csat/nps). Merit functions prevent "cool ideas" that don't create value.	

Organizational cultural impact of radical change		
Definition	Culture is: shared beliefs, norms, unwritten rules, "how things really work". Bpr directly attacks these unwritten rules.	
Aspect	Old (traditional) culture	Bpr-oriented culture
Control vs trust	Emphasizes tight managerial control, approvals, and supervision to ensure work is done correctly.	Relies on trust and accountability, empowering employees to make decisions within the process.
Hierarchy vs process ownership	Organized around hierarchical reporting lines where responsibility is fragmented across departments.	Organized around end-to-end process ownership, with one owner accountable for outcomes.
Specialization vs multi-skilled roles	Employees perform narrow, specialized tasks repeatedly.	Employees handle broader, multi-skilled roles, often managing work end-to-end.
Example	In a traditional accounts department, multiple clerks handle invoice entry, checking, and approval with heavy supervision.	Under bpr, a single case manager handles the entire payment process digitally, reducing handoffs and approvals.

Bpr	Old belief: “customers cannot be trusted.” New belief: “customers can self-serve using digital tools.” (starbucks mobile ordering is a clear illustration.)

How does bpr challenge organizational culture?
Bpr challenges deeply embedded norms such as hierarchical control, functional silos, and reliance on manual approvals by replacing them with process ownership, trust-based automation, and end-to-end accountability.

Explain the cultural impact of radical redesign.
Radical redesign disrupts existing organizational culture by shifting from control to trust, hierarchy to process ownership, and specialization to multi-skilled roles. Employees must unlearn rigid routines and accept accountability for end-to-end outcomes. Paper-based, rule-driven cultures struggle with digital-first workflows. This culture shock can cause anxiety and conflict but enables agility and customer focus. For example, introducing case managers replaces departmental silos with ownership, requiring a mindset shift from compliance to responsibility.

Outcome orientation	
Definition	Work is organized around **results**, not activities.
Examples	Outcome: “approved loan”. Not: “filled form”, “verified documents”, “manager signed”

Why does bpr focus on outcomes rather than tasks?
Bpr focuses on outcomes rather than tasks because customers value results, not internal activities. Task-based design fragments work into narrow roles, creating handoffs, delays, and unclear accountability. Outcome-oriented processes assign responsibility for the entire result to an individual or team, reducing coordination costs and errors. For example, a case manager handling an application end-to-end delivers faster and higher-quality service than multiple specialists performing isolated tasks. By organizing around outcomes, bpr improves speed, quality, and customer satisfaction simultaneously.

Pareto analysis (80/20 rule)	
Definition	Pareto analysis is based on the principle that a small number of causes account for most of the problem. Often phrased as: 80% of problems come from 20% of causes.
Method	List causes. Measure frequency or impact. Sort from largest to smallest. Focus on top contributors
Bpr	Prioritizes improvement. Avoids spreading effort thin. Supports data-driven decisions
Examples	Out of 100 late orders: 60 due to approvals, 20 due to missing materials, 10 due to staff absence, 10 due to system issues. → focus first on approvals.

How does pareto analysis support decision-making?
It identifies the most impactful causes of inefficiency, allowing organizations to focus resources on areas that yield the highest improvement.

Explain pareto analysis and its use in prioritization.
Pareto analysis is based on the 80/20 principle, which states that a small number of causes usually account for most of the problems. It uses data to rank causes by frequency or impact, often displayed in a pareto chart. In bpr, pareto analysis helps prioritize improvement efforts by identifying high-impact issues first. For example, if 80% of service delays are caused by heater issues and payment delays, fixing these two areas yields maximum improvement. This prevents wasting resources on low-impact problems.

Aspect	Fishbone diagram (ishikawa)	5 whys technique	Pareto analysis (80/20 rule)
Nature	Qualitative and visual	Qualitative and logical	Quantitative and data-driven
Basic structure	Cause-and-effect diagram shaped like a fish skeleton with branches	Linear chain of repeated “why?” Questions	Bar chart ranking causes by frequency or impact
Primary purpose	To explore and organize all possible causes of a problem	To drill down to the root cause of a specific issue	To prioritize the most important causes contributing to a problem
Type of thinking	Divergent (broad exploration)	Convergent (narrowing down)	Selective (focus on the vital few)
Problem complexity	Best for complex problems with many interacting factors	Best for simple or focused problems	Best when many causes exist but resources are limited
Cause categorization	Uses standard categories (people, process, machine, material, environment, measurement)	No categories; follows one cause chain	Causes are grouped and ranked numerically
Data requirement	Optional (can be used with or without data)	No numerical data required	Requires numerical data (counts, cost, defects, delays)
Output	A map of possible root causes by category	A single root cause (or very few)	A ranked list of causes showing highest impact

Decision support	Helps understand where problems originate	Helps decide why a problem happens	Helps decide what to fix first
Time & effort	Moderate to high (brainstorming + structuring)	Low (quick questioning)	Moderate (data collection + analysis)
Risk / limitation	Can become too broad without prioritization	May oversimplify complex problems	Does not explain why causes occur
Typical use order	First step in diagnosis	After fishbone, to validate causes	After causes are identified, to set priorities
Use in bpr / six sigma	Analyze phase: identifying process weaknesses	Root cause validation	Measure/analyze phase: focusing improvement efforts
Example (tea delivery case)	Identifies causes under people (staff speed), machine (heater), materials (cups), process (sequence), environment (layout)	Late tea → why? Heater slow → why? No pre-heating → why? No standard setup	Shows 80% of delays come from heater issues and missing materials

"processizing" an organization	
Definition	Processizing means designing work as **repeatable, documented, measurable processes** rather than informal "people just do it."
Why its done	Predictable quality. Lower dependency on one person, easier training, measurement + improvement
Examples	**1:** helpdesk ticketing system replaces random whatsapp messages. **2:** purchase order workflow replaces "ask the manager verbally."
Risk	Over-processizing can create bureaucracy: too many approvals, too many forms.

Explain "processizing" and one risk of over-processizing.

Processizing means formally defining, structuring, and standardizing work into repeatable processes to improve consistency and efficiency. It helps organizations scale operations and reduce dependency on individuals. However, over-processizing can reduce flexibility and innovation by forcing rigid rules on dynamic situations. Employees may spend more time following procedures than solving real problems. Excessive controls can also slow decision-making and frustrate customers. Therefore, while processizing improves reliability, it must be balanced with adaptability and judgment.

Creating an inventory of processes

Once targets are defined, the organization does something very important: each department creates an inventory of its processes. This means listing every process, such as: procurement, billing, order handling, customer support, approval workflows. These are labeled conceptually as: process 1, process 2, … process n. This step is crucial because you cannot improve what you do not identify. Bpr always begins with process visibility

Process event analysis (operational bpr insight)			
Definition	In process analysis, not all cases complete the full cycle.		
	Some cases: • end early, • get stuck in analysis, • experience long waiting times.	By comparing: • short incomplete cases, • long complete cases.	Organizations can: • identify bottlenecks, • reduce delays, • automate reminders, • simplify communication.
BPR	This is a classic process mining and redesign approach used in bpr to improve flow efficiency.		

Process mapping	
Definition	Process mapping converts invisible work into visible structure. **In real organizations**: work is scattered across people, systems, emails, forms, approvals, delays are hidden as "waiting", responsibility is unclear, no one owns the full process **Process maps**: make handoffs visible, reveal delays and duplication, show where data is re-entered, expose non-value-adding steps, create a shared understanding across stakeholders
Bpr	Only after mapping can redesign begin.
Examples	Before mapping: "student registration is slow." After mapping: you can see: 4 approvals, 2 data re-entries, 1 long waiting queue, unclear ownership

What is process mapping and why is it essential in bpr?

Process mapping is the practice of visually documenting how work flows through a process—from the trigger to the final outcome—by showing activities, decisions, handoffs, and sometimes time/cost data. It is essential in bpr because bpr requires radical redesign, and you cannot redesign what you do not clearly understand. Mapping makes invisible work visible: it exposes delays, rework loops, duplicate steps, unnecessary approvals, and unclear responsibility. It creates a shared "single version of truth" for stakeholders, reducing arguments based on assumptions. Most importantly, mapping

provides a baseline (as-is) so improvements can be measured and justified through data (cycle time, waiting time, defects). Without mapping, redesign becomes guesswork and often fails.

How do process maps help identify bottlenecks?

Process maps help identify bottlenecks by making the process flow visible and showing where work accumulates, loops, or waits. In swimlanes, bottlenecks often appear at steps involving approvals, cross-lane handoffs, or scarce resources. In vsm, bottlenecks become clear when one step has the highest process time or causes the highest waiting time before it. Maps also reveal repeated decision points, rework loops, and unnecessary checks that slow throughput. Once mapped, you can attach metrics (cycle time per step, queue time, defect rate) to confirm the bottleneck with data rather than opinion. This is crucial in bpr because improving non-bottleneck steps doesn't improve overall performance.

Process mining	
Definition	Process mining is a technique that uses event data from information systems to discover, analyze, and improve real process flows. It is discovering/diagnosing processes via event logs with timestamps; variants and happy path matter.
Concept	It answers: what actually happens? How long does it take? Where do cases diverge? Where are delays concentrated?
	Process mining is a data-driven way to discover and evaluate real processes from event logs (what actually happened, not what people think). The minimum event log fields are: case id (process instance), activity name, timestamp (often also resource/department). With timestamps you can compute waiting vs working time, identify the bottleneck step, detect rework loops, and compare as-is vs to-be performance.
Limitations	Requires clean, complete event logs. Cannot explain why behavior occurs (needs interviews). Misses informal work done outside systems. Not suitable for low-volume or non-digital processes. Process mining shows what happens and where delays occur, but not why people behave the way they do.
Examples	Case=order#52, activities: order_placed 10:01 → paid 10:08 → prepared 10:18 → delivered 10:22 ⇒ biggest delay is payment queue (7 min).

What is process mining and what problem does it solve?

Process mining is a data-driven technique that analyzes system event logs to reconstruct how processes actually run in reality. It solves the problem of inaccurate or idealized process maps by revealing real execution paths, variations, bottlenecks, and rework. Unlike interviews or workshops, process mining is based on factual system data, reducing human bias. It exposes hidden delays, rare cases, and compliance issues that traditional mapping misses, making it especially valuable for diagnosing complex, high-volume processes before redesign.

How does process mining help identify bottlenecks?

Process mining identifies bottlenecks by analyzing timestamps in event logs to measure delays between activities. It highlights steps where cases spend excessive waiting time, queue up, or frequently loop back. Unlike assumptions, it shows exactly where congestion occurs and which variants are affected. This allows organizations to distinguish true bottlenecks from perceived ones and focus improvement efforts on constraints that limit throughput and overall performance.

How does process mining complement six sigma?

Process mining complements six sigma by revealing where variation and delays occur, while six sigma explains why they occur using statistical analysis. Mining identifies unstable steps, rework loops, and bottlenecks across variants, providing factual inputs for six sigma analysis. Together, they enable data-driven diagnosis and targeted improvement, combining structural visibility with statistical rigor for stronger process performance.

Define process mining and state 3 insights it provides without naming specific software.

Process mining uses event logs (case id, activity, timestamp, resource) to discover the real as-is process and measure performance. It provides: (1) variants (different paths cases follow), (2) bottlenecks/waiting times (where time accumulates), (3) rework loops (repeated steps), and (4) compliance deviations (skipped approvals). This supports data-driven bpr by showing where delays are structural rather than "people being slow."

Process mining vs data mining

Process includes timestamps + sequence of activities. Data doesn't necessarily include process flow/time sequence.

Aspect	Process mapping	Process mining
Source of information	Based on interviews, workshops, and assumptions about how the process should work.	Based on actual system event logs, showing how the process really runs.
Accuracy & bias	Can be inaccurate and biased due to human memory and perception.	Highly accurate with low bias since it reflects real execution data.
Visibility of variation	Shows a single, idealized flow with limited visibility of exceptions.	Reveals all variations, rework loops, and deviations in the process.
Examples	A team maps the "loan approval process" based on staff descriptions.	System logs show many applications bypass approval or loop multiple times.

	A university draws a registration process assuming one approval path.	Process mining reveals multiple paths and delays during peak regsistration.

Process ownership	
Traditionally	Tasks owned by departments, no one owns the full process, problems fall "between" departments
Definition	One owner responsible for the entire process, accountable for outcomes, not tasks, authority across functional boundaries
Benefits	Clear accountability. Faster decisions, fewer handoffs, better customer experience

Why is process ownership essential in bpr?

Process ownership is essential in bpr because end-to-end processes cut across multiple departments, and without a single owner, accountability becomes fragmented. In functional organizations, each department optimizes its own tasks, while no one is responsible for overall outcomes such as cycle time or customer satisfaction. A process owner is accountable for the complete workflow, ensuring coordination, faster decision-making, and performance improvement across boundaries. This ownership reduces delays, political conflict, and handoff problems, making radical redesign feasible and sustainable.

Who is a process owner?

A process owner is a single person or role accountable for the performance of an entire end-to-end process, not just a task or department. Unlike functional managers, a process owner is responsible for outcomes such as cycle time, cost, quality, and customer satisfaction across departments. Process ownership reduces coordination failures, handoff delays, and blame-shifting. In bpr, assigning a process owner is critical because radical redesign requires clear accountability. For example, a "student registration owner" ensures the full journey works smoothly, not just admissions or accounts.

Aspect	Private sector bpr	Public sector bpr
Primary objective	Profit maximization, cost reduction, speed, and customer satisfaction drive bpr initiatives.	Public accountability, compliance, transparency, and service delivery drive bpr, not profit.
Example	A bank redesigns loan approval to reduce turnaround time and gain market share.	A government office redesigns license issuance to reduce public complaints.
Decision-making speed	Faster decision-making due to fewer approval layers and centralized authority.	Slower decision-making due to multiple approvals, rules, and political oversight.
Cause	Management autonomy and competitive pressure.	Legal constraints, inter-departmental coordination, and political scrutiny.
Risk tolerance	Higher tolerance for risk because failure affects profits, not legitimacy.	Low risk tolerance due to fear of public backlash, audits, and media attention.
Effect	Willingness to attempt radical redesign and experiment.	Preference for minor changes instead of radical redesign.
Incentives & motivation	Strong incentives: bonuses, promotions, market survival.	Weak or no performance-linked incentives; job security dominates motivation.
Example	Employees support bpr if it improves performance bonuses.	Employees resist bpr since efficiency gains do not improve pay or security.
Political influence	Minimal political interference; decisions are business-driven.	High political influence from ministers, unions, and external stakeholders.
Issue	Strategy may prioritize profit over employee well-being.	Redesign decisions may be reversed due to political pressure.
Flexibility of structure	Highly flexible structures allow role redesign and hierarchy flattening.	Rigid structures defined by law, service rules, and civil service regulations.
Effect on bpr	Easier to assign process ownership and remove redundant roles.	Difficult to eliminate roles or merge departments even if inefficient.
Speed of change	Faster implementation due to fewer stakeholders.	Slower implementation due to consensus-building and resistance.
Cultural environment	Performance-oriented culture supports change and accountability.	Stability-oriented culture resists disruption and prefers status quo.

Why public sector bpr often fails?

Public sector bpr frequently fails due to union resistance, political interference, lack of personal incentives, and fear of public criticism. Employees often perceive bpr as a threat to job security rather than an opportunity for improvement. Additionally, rigid laws and accountability requirements limit the ability to redesign processes radically, resulting in partial or symbolic changes instead of true reengineering.

Why is bpr more challenging in the public sector?
Because public sector organizations operate under political scrutiny, rigid rules, and job security guarantees, making radical change difficult despite potential efficiency gains.

Compare public and private sector bpr challenges.
Private sector bpr faces market pressure, enabling faster decisions, higher risk tolerance, and stronger incentives. Public sector bpr is constrained by bureaucracy, political oversight, unions, job security, and public accountability. Political interference and fear of backlash slow change. While private firms redesign for profit and speed, public organizations prioritize transparency and equity. For example, a bank can redesign loan processing quickly, but a government office faces legislative approvals and union resistance. Bpr in the private sector focuses on cost, speed, and customer satisfaction, with stronger incentives and faster decision-making. In contrast, public sector bpr faces political oversight, rigid regulations, limited incentives, and higher resistance to change. Performance metrics are often unclear, and job security reduces urgency. While private firms can redesign aggressively, public organizations must balance efficiency with fairness, transparency, and accountability, making radical redesign more difficult but still necessary.

Root cause analysis (rca)			
Definition	Root cause analysis is a structured approach to identify the underlying causes of a problem rather than treating its symptoms. Bpr cares about causes, not surface issues.		
Symptoms	Late deliveries	Poor scheduling logic	Fixing symptoms leads to temporary relief, not improvement.
	High defects	Inconsistent input quality	
	Long wait times	Approval dependency	
Logic	Problem → why? → why again? → until a controllable cause is found.		
Example	Problem: tea is late. Why? Heater slow. Why? Heater shared. Why? No capacity planning → root cause: lack of equipment planning		

Why is root cause analysis critical before redesign?
Because redesigning based on symptoms risks fixing the wrong issue. Rca ensures that improvements target the true drivers of inefficiency rather than superficial effects.

Explain root cause analysis and its importance in bpr.
Root cause analysis (rca) is a structured approach used to identify the fundamental causes of a problem rather than its symptoms. Tools like fishbone diagrams, 5 whys, and pareto analysis help uncover why a process fails repeatedly. Rca is essential in bpr because redesigning a process without understanding its root problems leads to superficial fixes. For example, hiring more staff may reduce delays temporarily, but if the real issue is poor layout or unclear roles, performance will degrade again. Effective bpr depends on deep diagnosis before redesign.

Sipoc diagram				
Definition	Sipoc stands for: suppliers, inputs, process, outputs, customers. It is a high-level process boundary definition tool. It answers the question: "what is this process, who feeds it, and who receives it?"			
Purpose	Define scope, avoid getting lost in details, align stakeholders, clarify inputs and outputs. Sipoc is often done before detailed mapping.			
It is not	It does not show sequence, timing, decisions. It is strategic, not operational.			
Example	Student registration:			
Supplier	**Input**	**Process**	**Output**	**Customer**
Student	Application, documents	Registration	Enrollment confirmation	Student

Give one limitation of sipoc.
It does not capture sequence, timing, or internal decision logic, so it must be followed by detailed mapping.

What is sipoc and why is it used before process redesign?
Sipoc defines the boundaries of a process by identifying suppliers, inputs, outputs, and customers. It ensures alignment and prevents scope creep before detailed analysis.

Define sipoc and explain its role in scoping.
Sipoc is a high-level process definition tool that stands for suppliers, inputs, process, outputs, customers. Its main role is scoping—it defines the boundaries of a process before detailed mapping begins. Sipoc answers: who provides inputs, what inputs are required, what the process generally does, what outputs are produced, and who receives those outputs. This prevents scope creep (mapping everything in the organization) and ensures stakeholders agree on what is inside vs outside the process. For example, in "student registration," suppliers may include students and finance, inputs may include forms and fee confirmation, outputs may include enrollment confirmation, and customers include the student and university systems. Sipoc is not a sequence map; it's a boundary and stakeholder clarity tool.

Six sigma	
Definition	Six sigma is not a philosophy like bpr. It is a measurement-driven discipline focused on variation control. Six sigma is a data-driven methodology aimed at reducing variation and defects in a process so that outputs

	consistently meet customer requirements. If your process is inconsistent, you get rework, delays, customer dissatisfaction. So BPR cares not only about average speed, but consistency.
Core idea	If variation is reduced, defects automatically reduce. Six sigma is not just math. After measurement, you must go back to the process. It is reducing process variation and defects through measurement and control. You look at histograms of times/defects
Best when	data is captured automatically. you need continuous measurement; manual tracking is too slow/noisy.
	High variance = inconsistent process, Low variance = stable, predictable process

Define six sigma and explain its core objective.

Six sigma is a data-driven process improvement methodology aimed at reducing defects and variation in processes. Its core objective is to achieve consistent, predictable performance by designing processes that operate within tight tolerance limits. By using statistical analysis, six sigma identifies root causes of errors and eliminates them systematically. The ultimate goal is to improve quality, reduce costs, enhance customer satisfaction, and increase profitability by ensuring that outputs meet customer requirements almost all the time.

Why must six sigma analysis return to the process steps?

Six sigma analysis must return to process steps because defects originate from specific activities, handoffs, or decisions within the process. Statistical results alone cannot fix problems unless linked to actual operational causes. By mapping defects back to steps, organizations identify where variation is introduced and implement targeted improvements. This ensures that data-driven insights translate into real process changes rather than remaining abstract measurements.

Explain six sigma results in business terms.

Six sigma results translate into measurable business benefits such as cost reduction, faster cycle times, higher customer satisfaction, and improved profitability. Lower defect rates reduce rework and waste, saving money. Stable processes improve delivery reliability and customer trust. Improved quality leads to fewer complaints and higher retention. By expressing results in financial savings, revenue growth, and risk reduction, six sigma aligns operational improvements with strategic business goals.

Steps	1. Define unit and opportunities per unit (o). 2. Count defects (d) and units (n). 3. Compute dpu = d / n 4. Compute dpo = d / (n×o) 5. Compute dpmo = dpo × 1,000,000 6. Convert to yield ≈ 1 − dpo (for small dpo) 7. Interpret: lower dpmo ⇒ higher sigma ⇒ better process capability.	• N = 200 forms, o = 5 fields each, d = 30 errors. • Dpu = 30 / 200 = 0.15 • Dpo = 30 / (200 × 5) = 0.03 • Dpmo = 0.03 × 1,000,000 = 30,000 dpmo
Delays	Water heating (machine), missing materials, payment & receipt steps, counter congestion, role confusion. The process is failing not because of one big problem, but because small variations accumulate. This cumulative variation pushes delivery beyond 5 minutes.	
Bpr	Where bpr asks: “is this process fundamentally wrong?”, six sigma asks: “why does this process behave inconsistently?”	
Sigma	Sigma (σ) = standard deviation Standard deviation measures spread / variability of data Smaller σ → more consistency Larger σ → more unpredictability	Six sigma does not mean “six steps” or “six tools”. It refers to how far the process mean is from the specification limit, measured in standard deviations.	If average tea delivery time is 5 minutes but Sometimes 3 minutes Sometimes 9 minutes → high variation → unhappy customers → low sigma level
	High variability indicates: lack of standardization, too many exceptions, policy overload, role ambiguity		

Sigma level (capability interpretation)	Sigma level indicates how capable a process is of meeting requirements. Higher sigma = fewer defects.	**Sigma level**	**Defects per million**
		1σ	~690,000
		2σ	~308,000
	Dpmo ≈ 360,000. Sigma ≈ 1.9σ. This is below 2 sigma, meaning: process is highly unreliable. Customers frequently experience failure. The cafeteria cannot meet its service promise consistently. Improvement is mandatory, not optional.	3σ	~66,800
		4σ	~6,210
		5σ	~233
		6σ	~3.4

What does a low sigma level indicate?

A low sigma level indicates a process with high defect rates and significant variation. It reflects poor quality, unstable performance, and frequent customer dissatisfaction. Low sigma processes require constant firefighting, rework, and corrective actions, increasing costs and delays. For example, a 2-sigma process produces thousands of defects per million opportunities. Improving sigma level means reducing variation, improving consistency, and moving toward predictable, high-quality performance.

Mean (μ)	Arithmetic average of data, Shows central tendency	Average tea delivery = 5.68 minutes
Standard deviation (σ)	Measures spread of data around the mean High σ = wide spread, Low σ = tight clustering	Σ ≈ 1.79 minutes This indicates large variation

Explain mean and standard deviation with an example.

The mean is the average value of a dataset and represents central tendency, while standard deviation measures how spread out the data is around the mean. For example, if tea delivery times average 10 minutes, the mean is 10. If deliveries vary between 5 and 20 minutes, the standard deviation is high, indicating inconsistency. A low standard deviation means most deliveries occur close to 10 minutes, showing a stable and reliable process.

Distribution (bell curve intuition)	Most six sigma analysis assumes normal distribution Mean at center, spread defined by σ	If delivery times are widely spread → more late teas.

Kurtosis	Measures peakedness of distribution High kurtosis → data tightly clustered	Low kurtosis = more outliers = worse consistency. Low kurtosis → flatter curve, more extreme values

What is kurtosis and why does it matter in process performance?

Kurtosis describes how "peaked" or "flat" a distribution is—whether values cluster tightly near the mean or have heavy tails/outliers. High kurtosis (tight clustering) suggests consistent service times; low kurtosis suggests more extreme delays (outliers), which harm customer trust. In operations, kurtosis matters because customers remember worst cases: even if the average is fine, frequent long delays create perceived poor quality and increase complaints.

Variation	Variation is the natural fluctuation in process output over time. No real process is perfectly constant. Customers don't experience averages — they experience individual outcomes. A process with a good average but high variation still fails.

What is variation and why is it dangerous?

Variation refers to inconsistency in process outputs, where the same process produces different results over time. It is dangerous because it makes outcomes unpredictable, leading to defects, rework, delays, and customer dissatisfaction. High variation hides underlying process problems and increases operational risk. Even if average performance looks acceptable, large variation can still cause frequent failures. Six sigma focuses on controlling variation because stable processes are essential for quality, reliability, and cost control.

Define six sigma and explain "variation" using standard deviation.

Six sigma is a quality approach focused on reducing defects by reducing variation. Variation is the natural spread in process output (time, cost, errors) and is measured by standard deviation (σ): lower σ means more consistent performance. A high sigma level means the process mean can shift without violating requirements. In services, variation shows up as unpredictable delays: some customers get served fast, others wait too long—six sigma targets making the output stable and within specification.

Common cause variation	**Special cause variation**
Inherent to the process, Always present, Predictable within limits	Unusual, assignable, Comes from specific events
Minor differences in how fast staff pour tea.	Water heater malfunction, missing ingredients, staff absence.

Why does six sigma focus on reducing variation rather than improving averages?

Because customers experience individual outputs, not averages. High variation means inconsistent performance, which leads to defects even if the average output appears acceptable.

Rework	Every failure = rework. Rework means the process runs again → more cost.

Defect	**Opportunities**
A defect occurs when output fails to meet a customer-defined requirement.	An opportunity is a point where a defect can occur.
Tea delivered after 5 minutes = defect. Defect tolerance depends on product. biscuits/cola: near zero defects tolerated, chairs: higher defect tolerance possible. if yield is 99.7% then defect rate is tiny.	Each order = 1 opportunity

• Students not learning • Lack of focus • Zoning out • Incomplete work	Measured using: • Exit tickets • Live quizzes • On-task tracking • Next-day assessments	Opportunities in a classroom process: • Attention (start of class) • Instruction (teaching phase) • Practice (activities) • Exit/retention (end or next day)

Define defect and opportunity.

A defect is any outcome that fails to meet customer requirements or specifications. An opportunity is a single chance for a defect to occur within a process. For example, if a coffee order has three steps—order accuracy, temperature, and delivery time—each step is an opportunity for a defect. Identifying defects and opportunities allows organizations to measure process quality accurately and calculate performance metrics such as dpu and dpmo.

In six sigma terms, what are "defects" and "opportunities"? Give a non-factory example.

A defect is any outcome that fails the requirement (late tea, wrong order, payment not recorded). An opportunity is a point where a defect could occur. Your classroom example defined opportunities as lesson moments where performance can drop (start of class, explanation, practice, closure/recall). Defects there include lack of understanding, zoning out, incomplete work; measurement tools include exit tickets and live quizzes.

Dpu — defects per unit	**Dpmo — defects per million opportunities**

$$\text{Dpu} = \frac{\text{number of defects}}{\text{number of units}}$$	Measures defects relative to total opportunities. $$\text{Dpmo} = \frac{\text{defects}}{\text{units} \times \text{opportunities}} \times 1{,}000{,}000$$
Total orders = 50, late orders = 18 $$\text{Dpu} = \frac{18}{50} = 0.36$$ on average, 36% of units are defective. For every 100 customers: 36 are unhappy. This is very poor performance.	1 opportunity per order $$\text{Dpmo} = \frac{18}{50} \times 1{,}000{,}000 = 360{,}000$$ 360,000 defects per million → extremely inconsistent process.
For example, if 50 defects are found in 100 tea orders, dpu = 0.5. This means each unit has, on average, half a defect. A lower dpu indicates better process quality, while a higher dpu signals significant quality problems that require improvement.	Standardizes performance, allows comparison across industries, used to estimate sigma level. Lower dpmo indicates higher process capability and better quality.
Over one-third of customers receive late service. This inconsistency damages customer trust and increases operational stress. Reducing variation in heating, material availability, and role coordination is essential to meet service promises.	
Why is six sigma insufficient when a process is fundamentally broken? Because six sigma improves consistency within an existing process. If the process structure itself is inefficient or unnecessary, reducing variation will not eliminate waste or delays. In such cases, bpr is required before six sigma can be effective.	
Tea delivery example: 50 orders, 28 late (spec = within 5 mins). Compute dpu and dpmo (1 opportunity per order). Defects = 28, units = 50, opportunities per unit = 1. Dpu = defects/units = 28/50 = 0.56. Dpmo = (defects / (units×opportunities)) × 1,000,000 = (28/(50×1))×1,000,000 = 560,000. Interpretation: defect rate is extremely high; the process is not capable of meeting the 5-minute promise frequently. This matches your class conclusion that the tea process sigma level was very low.	

Aspect	Six sigma	Business process reengineering (bpr)
Primary focus	Variation and defect reduction in an existing process.	Fundamental redesign of an entire process.
Core objective	Make the process consistent, predictable, and stable.	Achieve dramatic improvement in cost, speed, and customer value.
Nature of change	Incremental improvement through continuous refinement.	Radical change through clean-sheet redesign.
Key question asked	“why is this process inconsistent?”	“why does this process exist at all?”
Approach	Improve the current process structure.	Challenge and replace the existing structure.
Primary tools used	Statistical analysis, data measurement, dmaic, control charts.	Process mapping, as-is vs to-be, redesign principles.
Data dependence	Highly data-driven; relies on metrics and calculations.	Uses data for diagnosis but focuses on conceptual redesign.
Risk level	Low risk, controlled and gradual changes.	High risk, but high potential reward.
Time horizon	Ongoing and continuous improvement effort.	Project-based and time-bound initiative.
Typical output	A stable and predictable process with low variation.	A new or completely transformed process.
Best used when	Process exists but is unstable or inconsistent.	Process is fundamentally broken or outdated.
Example	Reducing variation in tea delivery times using mean and standard deviation.	Eliminating invoice processing entirely using shared databases.
Compare six sigma and bpr. Six sigma focuses on improving existing processes by reducing variation and defects through statistical analysis. It is incremental and data-intensive. Bpr, in contrast, involves radical redesign of end-to-end processes to achieve dramatic performance improvements. While six sigma asks “why is the process inconsistent?”, bpr asks “why does this process exist at all?”. Bpr creates new processes, and six sigma later stabilizes and optimizes them.		

Aspect	Process mining	Six sigma	Business process reengineering (bpr)
Core question answered	Where does variation, delay, and deviation occur in the process?	Why does variation and defects occur?	Whether the process should exist in its current form at all.
Primary purpose	Discover the actual process flow and its variants using real execution data.	Analyze and reduce variation and defects to stabilize the process.	Radically redesign or eliminate processes to achieve dramatic improvement.
Nature of approach	Diagnostic and descriptive	Analytical and statistical	Strategic and transformative

Source of insight	System event logs and data traces	Measured process data and statistics	Process logic, value analysis, and business objectives
Type of problems addressed	Hidden loops, rework, bottlenecks, and deviations.	Inconsistency, defects, and instability in outputs.	Structural inefficiencies, unnecessary steps, and outdated rules.
Level of change	Does not change the process; reveals reality.	Improves the existing process incrementally.	Changes the process radically or eliminates it entirely.
Typical output	Visualized real process map with variants and delays.	Stable, predictable process with low defect rate.	New end-to-end process design.
Best used when	The real process behavior is unknown or disputed.	The process exists but is unstable or inconsistent.	The process is fundamentally broken or misaligned with value creation.
Example	Mining reveals approvals loop 3–4 times before completion.	Analysis shows approval delays caused by high variation in decision time.	Redesign removes approvals entirely using shared data.

Aspect	Stable process	Unstable process
Process variants	Has few, well-defined variants that follow a consistent and repeatable path.	Has many variants, with frequent deviations, rework loops, and exceptions.
Cycle time	Predictable and narrow range **of cycle time; tasks complete within expected limits.**	Wide and unpredictable range **of cycle time; same task may take minutes or hours.**
Output quality	Produces consistent quality outputs that meet specifications regularly.	Produces inconsistent outputs, leading to defects and rework.
Level of control	Governed by clear rules, standards, and procedures rather than individuals.	Dependent on individual judgment and personal effort, causing variability.
Reliability	High reliability; customers can trust delivery time and quality.	Low reliability; customers experience delays and quality issues.
Improvement approach	Suitable for six sigma optimization and fine-tuning.	Requires process redesign (bpr) before optimization.
Example	Automated payroll processing that runs the same way every month.	Manual approval process where each manager handles requests differently.

Swimlane diagrams		
Definition	A swim lane diagram is a flowchart divided into horizontal or vertical lanes, where each lane represents a role, person, department, or system. It answers the question: “who does what, and when?”	
Importance	Most inefficiencies come from: handoffs, waiting for another person, approvals across departments, miscommunication. Swim lanes expose responsibility gaps and handoff delays.	
Structure	Each lane = one actor (e.g., customer, agent, system). Activities are placed in the lane of the actor performing them. Arrows crossing lanes = handoffs	
Benefits	When steps move across lanes: • Training cost increases • Coordination effort increases • Waiting time increases • Error probability increases	Swimlane analysis directly supports: • Cost calculations • Capex vs opex justification • Business case for redesign
Why are swimlane diagrams more useful than flowcharts in bpr? Because they explicitly show responsibility, handoffs, and coordination between roles, which are common sources of delay and inefficiency targeted by bpr.		
How do swimlanes help in cost analysis? They help identify how many people are involved, how often handoffs occur, and which roles consume the most time, enabling labor cost and complexity analysis.		
Why are swimlane diagrams superior for cross-functional analysis? Swimlane diagrams are superior because they show both the sequence of steps and the ownership of each step by separating the process into lanes (e.g., sales, finance, operations, customer, system). This makes cross-functional issues visible—especially handoffs, which are a major source of waiting time, miscommunication, rework, and approvals. When arrows cross lanes, you can instantly see coordination costs and delays. Swimlanes also highlight accountability problems: if a process has many lanes involved but no clear owner, it often becomes slow and political. In bpr, redesign often requires combining roles, removing approvals, or shifting tasks to the customer/system—swimlanes make those opportunities easy to identify.		

Value chain analysis (porter)

Definition	Value chain analysis examines how activities within an organization contribute to customer value and competitive advantage.
Bpr	Identifies which activities truly add value. Helps decide what to redesign, outsource, or eliminate. Supports "core vs non-core" decisions

Primary activities	**Support activities**
Inbound logistics Operations Outbound logistics Marketing & sales Service	Infrastructure Hr management Technology Procurement

Examples	Taco bell: core: food preparation, branding. Non-core: raw cooking → outsourced

What is value chain analysis and how does it aid redesign?
Value chain analysis examines how an organization's activities collectively create value for the customer, from inbound logistics to after-sales service. It distinguishes between value-adding and non-value-adding activities across the organization. In bpr, value chain analysis helps identify where value is actually created and where waste exists across departments. For example, repeated data entry across purchasing, receiving, and accounting adds no customer value. Redesign focuses on eliminating such activities and strengthening high-value ones, ensuring that processes are aligned with strategic goals rather than departmental convenience.

Value stream mapping (vsm)	
Definition	**Value stream mapping** is a process visualization tool that shows: all steps in the process, time spent on each step, which steps add value vs do not add value. It answers: **"where is time actually going?"**
Components	Process steps, process time (pt), waiting time (wt), information flow, value-adding vs non-value-adding classification
Examples	Total lead time = 10 days. Actual work time = 4 hours. This reveals massive redesign opportunity.

Why is vsm powerful in bpr?
it quantifies waiting and non-value-adding time, making inefficiencies measurable and justifying radical redesign.

How does vsm help identify bottlenecks?
By showing where waiting accumulates and where cycle time is highest relative to value added.

What is value stream mapping and how does it reveal waste?
Value stream mapping (vsm) is a mapping method that visualizes the entire flow of a process while explicitly separating value-adding time from non-value-adding time (waiting, rework, approvals, movement, duplication). Unlike simple flowcharts, vsm is built to reveal where time actually goes by capturing process time, wait time, and often work-in-progress (wip). It reveals waste because many processes have a small amount of real work time but massive waiting time. For example, an application may take 30 minutes of actual processing but 5 days of waiting for approvals. Vsm makes this visible and quantifiable, helping justify radical redesign (e.g., remove approvals, eliminate duplicate checks, use shared data). It is especially useful for identifying bottlenecks and delay hotspots.

Variant	
Definition	A variant is a unique path a case follows. Two cases can: use the same steps (same variant) or follow different paths (different variants). **Most common variant** = intended process. **Rare variants** = exceptions, breakdowns, policy overload. A variant is a specific pattern of steps (a path). Activity set may be similar, but columns/timestamps differ. Variants represent deviations/paths from expected root. Ideal: fewer variants.
Events	number of activities in that case.
Happy process	Most frequent variant = standard path = "happy process."
Single case	Usually exceptions/outliers (not standard).
Active case	currently being worked on. Like canteen rush: many arrive, then as time ends, fewer come, existing ones complete and leave.
Importance	High number of variants = low standardization. Rare variants often contain: exceptions, rework, special approvals, errors. BPR goal is usually: fewer variants unless exceptions are truly necessary.
Examples	Case a: 6 events → simple, fast. Case b: 18 events → complex, slow. The 18-event case is not "better"; it is more complicated and more delayed. If 608 cases and 608 variants → process is chaotic / inconsistent.

Why are rare variants important in process mining?
Because rare variants often contain rework, extra approvals, or exception handling that significantly increase cycle time and cost, making them key targets for redesign.

Define case and variant with examples.

A case is one complete instance of a process from start to finish, such as a single customer order or insurance claim. For example, one food delivery order from placement to delivery is one case. A variant is a unique path that cases follow through the process. For instance, some orders may include payment failure or manual approval while others do not. Variants show how many different ways a process executes and help identify complexity, deviations, and inefficiencies across cases.

Why are rare variants important?

Rare variants are important because they often represent exceptions, failures, or high-risk scenarios that consume disproportionate time and cost. Although they occur infrequently, they may cause delays, rework, customer complaints, or compliance violations. Ignoring rare variants leads to fragile redesigns that fail under exceptions. Analyzing them helps organizations understand edge cases, reduce firefighting, and design robust processes that handle both common and exceptional situations efficiently.

Compare the 18-event case vs 6-event case in purchasing. What does the difference indicate?

Your analysis: the 18-event case represents a more complete procurement lifecycle and is much longer; the 6-event case is a shorter, minimal/early-stage path. The difference indicates process variability: some cases complete end-to-end, others stop early due to rejection, changing priorities, missing approvals, or incomplete documentation. In bpr terms, variability signals weak standardization and unclear decision rules.

Give 3 recommendations to reduce variation and improve completion rate.

Recommendations from your report: standardize workflow, add automated reminders/escalations for stalled cases, set clear decision rules for quotation approval/rejection, and reduce unnecessary approvals/handovers. These reduce variation by ensuring most cases follow the intended path, reducing rework loops, and preventing "dead cases" that waste effort. Monitoring dashboards focusing on approval delays and invoice stages help continuously control bottlenecks.

Unlearning			
Definition	Unlearning is the deliberate process of letting go of outdated assumptions, habits, and mental models that no longer fit the current environment. Bpr requires unlearning before learning. Innovation fails when people learn new tools but keep old thinking.		
Difficulty	Old methods once worked. Success reinforces habits. People confuse experience with correctness		
Example	Moving from keypad phones to touchscreens required massive unlearning: • Keyboard moved into software • Needed cpu, os, new interaction design • Humans resisted and wanted both keyboard + screen initially		
Keyboard → touchscreen		**Physical ordering → digital ordering**	**Teaching example (class discussion)**
People had to unlearn physical typing logic.		Customers and staff unlearn face-to-face dependency.	Moving from one-way lectures to collaborative knowledge exchange.

Why is unlearning essential for successful BPR?

Because BPR challenges foundational assumptions about how work is done. Without unlearning outdated beliefs, organizations simply automate old practices instead of embracing fundamentally new ways of operating.

What is unlearning and why is it important?

Unlearning is the deliberate abandonment of outdated rules, assumptions, and habits that no longer serve organizational goals. In bpr, unlearning is essential because existing processes are often built on obsolete constraints. Employees must stop following rules like "we pay only after invoices" when new systems allow real-time verification. Without unlearning, organizations merely automate inefficiencies. True redesign requires questioning why steps exist and whether they should exist at all.

Explain unlearning and why it matters in radical redesign. Give an example.

Unlearning is deliberately letting go of old assumptions ("this is how it's always done") so new designs become possible. Your lecture example: shifting mindset beyond norms, like rethinking a keyboard/ui for changing screen sizes—new interaction rules require unlearning old interfaces. In bpr, unlearning removes "sacred rules" (multiple approvals, physical presence) and allows online ordering, self-service, or parallel work. Without unlearning, teams only automate the past.

Flowchart	Swim lane	Sipoc	Value stream mapping
A flowchart shows the sequence and logic of steps and decisions. It answers "what happens next?" But not clearly "who does it."	A swimlane diagram is a flowchart plus responsibility lanes, showing who does what and making handoffs visible—best for cross-functional processes.	Sipoc is a high-level scoping tool; it defines boundaries, inputs/outputs, and stakeholders but does not show sequence or timing.	Vsm focuses on time and waste by separating value-adding vs non-value-adding time and highlighting waiting, delays, and bottlenecks.
sipoc is often used first (scope), then flowchart/swimlane (detail), and vsm for measurement-based waste removal.			

Aspect	Business process reengineering	Six sigma	Total quality management (tqm)
Core idea	Fundamentally rethink and radically redesign processes to achieve dramatic performance improvement.	Use data and statistics to reduce variation and defects in an existing process.	Embed quality as a culture and continuously improve processes through employee involvement.
Primary goal	Radical transformation of how work is done.	Consistency and predictability in outputs.	Continuous quality improvement over time.
Focus area	End-to-end processes that cut across departments (e.g., order-to-cash).	Stability of process performance (variation, defects).	People, culture, and mindset toward quality.
Nature of change	Radical and disruptive; questions whether steps should exist at all.	Incremental and analytical; improves existing steps.	Incremental and cultural; small improvements become a habit.
Time horizon	Short-term, project-based transformation.	Ongoing and continuous improvement cycles.	Long-term and continuous organizational journey.
Type of change	Structural (roles, sequence, ownership, approvals).	Statistical (variation, defect reduction).	Behavioral and cultural (attitudes toward quality).
Use of technology	Technology is an enabler of new process designs (e.g., shared databases removing steps).	Technology is used for measurement, monitoring, and analysis.	Technology supports training, documentation, and feedback systems.
Risk level	High risk, but very high potential payoff.	Low risk, controlled and data-driven.	Low to moderate risk, slow and steady improvement.
Performance metrics	Cycle time, cost, speed, customer satisfaction.	Defects per unit, dpmo, sigma level, variation.	Quality trends, customer satisfaction, employee participation.
Role of employees	Roles may change or be eliminated; resistance is common.	Employees participate as data collectors and problem-solvers.	Employees are central drivers of quality and improvement.
Typical scenario / example	Invoice processing is slow and complex → eliminate invoices entirely using shared data (ford ap case).	Tea delivery times vary widely → analyze mean, standard deviation, and reduce variation.	Customer complaints slowly declining → use training and feedback to steadily improve service quality.
When it is most appropriate	When the process is fundamentally broken and incremental fixes won't help.	When the process exists but is unstable or inconsistent.	When the process is mature and stable but needs long-term quality improvement.
Question it answers	"why does this process exist in this form?"	"why does this process behave inconsistently?"	"how do we sustain quality improvement over time?"
Differentiate bpr and six sigma. Bpr focuses on radical redesign of processes to achieve dramatic improvements, while six sigma focuses on reducing variation and defects through statistical analysis. Bpr changes the process structure, whereas six sigma improves process consistency.			
Bpr redesigns the process itself, six sigma stabilizes the redesigned process, and tqm sustains quality culture over time.			

Aspect	Functional silos	Process-oriented organization
Definition	A functional organization is structured around departments based on specialization (e.g., hr, finance, sales, operations). Each department focuses on its own tasks and responsibilities rather than the complete flow of work.	A process-oriented organization is structured around end-to-end business processes (e.g., order-to-cash, procure-to-pay, hire-to-retire), where work is organized to deliver value from start to finish.
Primary goal	The primary goal is to optimize departmental performance, such as reducing hr costs, speeding up finance approvals, or increasing sales targets—often without regard for overall customer experience.	The primary goal is to optimize overall process performance, ensuring that the entire workflow delivers value efficiently in terms of speed, quality, cost, and customer satisfaction.
Ownership & accountability	Ownership is fragmented across departments. Each department is responsible only for its own tasks, so no single person is accountable for the entire	There is clear end-to-end process ownership. A single process owner is accountable for the entire workflow and its final outcomes, reducing ambiguity and blame-shifting.

	outcome. Problems often fall "between departments."	
Performance measurement	Performance is measured using local kpis, such as departmental efficiency, individual productivity, or internal targets. These metrics can improve locally while the overall process remains slow or broken.	Performance is measured using end-to-end kpis, such as cycle time, defect rates, rework, cost per transaction, and customer satisfaction, ensuring alignment with organizational goals.
Coordination & workflow	Work moves through handoffs and approvals between departments. Each handoff introduces waiting time, coordination overhead, rework, and communication errors.	Work is handled by cross-functional teams with fewer handoffs. Activities are coordinated across roles, reducing delays, duplication, and miscommunication.
Customer perspective	The customer is often viewed as "someone else's responsibility." Each department focuses on its own output rather than the customer's end-to-end experience.	The customer is placed at the center of the process. All activities are designed and evaluated based on how they affect customer value and satisfaction.
Speed & efficiency	Processes are typically slow, as work waits in queues between departments and approvals. Even efficient departments cannot improve overall speed due to bottlenecks elsewhere.	Processes are faster and smoother, as unnecessary steps, approvals, and handoffs are reduced or eliminated through holistic design.
Visibility of the process	No one has full visibility of the entire process. Each department sees only its own part, making it difficult to identify bottlenecks or systemic issues.	The entire process flow is visible, making bottlenecks, delays, and non-value-adding activities easier to identify and redesign.
Change & improvement ability	Improvements are slow and political. Changes require negotiation across departments, and resistance is high because changes affect power and control.	Redesign is easier and more structured because the full process is visible, ownership is clear, and changes are justified using end-to-end performance data.
Suitability for bpr	Poor fit for bpr because functional silos reinforce fragmented thinking and incremental improvements rather than radical change.	Highly suitable for bpr because it aligns with radical redesign, end-to-end thinking, and dramatic performance improvement.
"order fulfillment" touches sales, finance, warehouse, delivery. In silos, each optimizes their own part; the customer experiences delays and errors. Process orientation redesigns the entire flow.		
Why do functional silos create inefficiency? Functional silos create inefficiency because work is fragmented across departments, each optimizing its own goals rather than the overall process. This leads to excessive handoffs, approvals, waiting times, and rework. No single owner is responsible for the complete outcome, so problems fall between departments. Local kpis may show improvement while the overall process remains slow and costly. Additionally, communication gaps and conflicting priorities delay decision-making. As a result, functional silos increase cycle time, reduce responsiveness, and weaken customer experience.		

Aspect	Incremental improvement	Radical redesign (bpr)
Core idea	Focuses on making small, continuous optimizations within the existing process structure. The basic process logic remains unchanged.	Focuses on fundamentally rethinking and redesigning the entire process from scratch, questioning whether steps should exist at all.
Nature of change	Minor adjustments such as reducing errors, shortening individual steps, or improving accuracy.	Structural changes such as removing steps, changing the sequence of activities, redefining roles, or combining tasks.
Magnitude of impact	Produces small, gradual gains (e.g., 5–10% improvement).	Produces dramatic, order-of-magnitude improvements (e.g., 50–80% reduction in time or cost).
Time horizon	Continuous and ongoing, often embedded in daily operations and quality initiatives.	Project-based and time-bound, carried out as a major transformation initiative.
Risk level	Low risk, since changes do not disrupt the overall process design or organizational structure.	Higher risk, as it challenges existing roles, authority, and workflows, but offers much higher potential rewards.
Mindset	Asks: "how can we improve this step?"	Asks: "should this step exist at all?"
Use of technology	Technology is used to automate or speed up existing steps.	Technology is used as an enabler of new process designs, often eliminating steps entirely.
Effect on inefficiencies	Reduces local inefficiencies but often retains handoffs, approvals, and delays.	Eliminates root inefficiencies such as unnecessary approvals, duplication, and waiting.

Organizational disruption	Minimal disruption; existing roles and power structures remain mostly intact.	Significant disruption; roles, responsibilities, and authority may change.
Example	Automating invoice matching to reduce manual effort. Faster data entry, better training, minor automation	Eliminating invoices entirely by using shared databases to trigger payment. Letting customers self-serve, eliminating approval layers

Explain why bpr prefers radical redesign over incremental changes.
Incremental changes improve existing steps but keep the same structure, so core delays like handoffs, approvals, and duplication remain. Bpr seeks dramatic performance improvements by removing non-value steps, combining tasks, redesigning roles, and using shared information so the process itself changes, not just its speed.

Why do incremental improvements sometimes fail to improve overall performance?
Incremental improvements often fail because they optimize individual steps without addressing fundamental process flaws. Improving a non-bottleneck step does not increase overall throughput if the bottleneck remains unchanged. Incremental changes also preserve existing handoffs, approvals, and silos, which continue to cause delays. As a result, local gains do not translate into end-to-end improvement. In complex processes, performance is limited by structure, not small inefficiencies. This is why bpr advocates radical redesign instead of continuous tinkering.

Why do incremental improvements sometimes fail, and when is radical redesign justified?
Incremental improvement fails when the process is fundamentally flawed (too many handoffs, outdated rules, wrong structure). Small fixes reduce seconds, but the system still has "built-in" delays. Radical redesign is justified when performance goals are dramatic (e.g., 50–80% time/cost reduction), when customer needs or tech capabilities have changed, or when the process is broken by policy complexity. Bpr uses clean-sheet thinking: redesign the outcome, then rebuild the process around it.

Real-world bpr example: starbucks
Starbucks is a textbook bpr example. Before: in-person ordering, single physical queue, manual payment. After: mobile ordering, pre-payment, parallel queues, personalized offers. This redesign: eliminated bottlenecks, shifted work to customers, increased speed and accuracy, improved loyalty and revenue. This is radical redesign, not automation — exactly what bpr promotes.

Value-adding (va) steps	Non-value-adding (nva) steps
A step is va if it: • Changes the product/service toward what the customer wants, • Is done right the first time, • Customer would be willing to pay for it. In many processes, **va time < 10%** of total lead time.	A step is nva if it • Is waiting, rework, duplication, unnecessary approvals, reconciliation, searching for info. Nva steps are often "historical leftovers." Bpr questions whether they should exist at all.
Cooking food, diagnosing a patient, shipping the product.	Waiting for signature, entering same data twice, matching an invoice because systems don't talk.

What is the difference between va and nva? Give two examples each.
Va steps directly contribute to what the customer wants and would pay for, like assembling a product or diagnosing a patient. Nva steps add time/cost without improving output, like waiting for approvals or re-entering the same data in multiple systems. Value-adding (va) steps directly change the product/service in a way the customer cares about and would "pay for." Non-value-adding (nva) steps consume time/cost without improving the outcome (often approvals, waiting, re-entry). Va examples: brewing coffee; verifying a payment and issuing receipt. Nva examples: waiting in queue; duplicate form filling / re-typing the same data; unnecessary signatures. Bpr targets removing nva and redesigning the flow so va happens faster with fewer handoffs.

Aspect	Value chain	Process mapping
Level	Strategic level **tool used by top management to understand how the organization creates value and competitive advantage.**	Operational level **tool used by managers and analysts to understand how a specific process actually works.**
Primary focus	Focuses on value creation by identifying activities that add value versus those that do not.	Focuses on flow efficiency, showing how work moves through steps and where delays or bottlenecks exist.
Scope	Organization-wide **view covering primary and support activities (e.g., inbound logistics, operations, marketing).**	Process-specific **view focusing on a single workflow (e.g., order processing, loan approval).**
Decision it supports	Helps answer "what activities should we perform to create value?"	Helps answer "how should this process be executed efficiently?"
Nature of analysis	High-level and conceptual; emphasizes strategic positioning and cost vs differentiation.	Detailed and visual; emphasizes step-by-step execution and handoffs.

Typical output	A breakdown of value-adding vs non-value-adding activities across the organization.	A visual map (flowchart, swimlane, or vsm) showing activities, decisions, and roles.
Time horizon	Long-term; used for strategy formulation and competitive advantage.	Short- to medium-term; used for process improvement and redesign.
Use in bpr	Helps decide which processes are critical and should be reengineered.	Helps redesign how a selected process should be executed (as-is vs to-be).
Example	A university identifies teaching, admissions, and research as core value activities, while administration is support.	Mapping the student registration process to identify delays in fee verification and approvals.
Another example	A manufacturing firm uses value chain analysis to decide whether to outsource logistics.	The same firm maps the order-to-delivery process to reduce cycle time and handoffs.

Aspect	Vertical integration (traditional / hierarchical)	Horizontal integration (process-based)
Basic idea	Work flows up and down the organizational hierarchy, moving through multiple authority levels before decisions are made.	Work flows across functions and departments along the value stream, focusing on completing the process end-to-end.
Flow direction	**Staff → supervisor → manager → director. Decisions travel upward and then downward.**	**Sales ↔ finance ↔ operations ↔ customer. Work moves laterally across departments.**
Primary objective	Emphasizes control, supervision, and compliance with rules and authority.	Emphasizes speed, coordination, and customer value delivery.
Decision making	Decisions are centralized at higher levels, requiring approvals at each managerial layer.	Decisions are decentralized, often handled by cross-functional teams or process owners.
Speed & cycle time	Slower **due to multiple approvals, waiting time, and hierarchical escalation.**	Faster **because work progresses without unnecessary approvals and delays.**
Coordination mechanism	Coordination happens through formal approvals and reporting lines.	Coordination happens through collaboration and shared responsibility across functions.
Accountability	Accountability is fragmented; each level or department is responsible only for its part.	Accountability is end-to-end; one process owner is responsible for the complete outcome.
Customer impact	Customer needs are often secondary to internal control and policy compliance.	Customer needs are central, and the process is designed around customer experience.
Handling of handoffs	Frequent handoffs between levels increase delays and risk of miscommunication.	Fewer handoffs reduce waiting, errors, and rework.
Flexibility & adaptability	Low flexibility; changes require approval across multiple hierarchical levels.	High flexibility; teams can adapt quickly to changes in demand or requirements.
Suitability for bpr	Poor fit for bpr because hierarchy reinforces slow, approval-heavy processes.	Strong fit for bpr because it supports cross-functional redesign and radical improvement.

Lead time	Time from customer request to delivery outcome. Lead time is often "customer visible."
Active time	Time when work is actually being done.
Waiting time	Time when nothing happens: waiting for approval, waiting for data, waiting for availability
Explain waiting time vs active time. Active time is the period when actual work is performed on a task, such as entering data or approving a request. Waiting time is the idle time between activities, often caused by queues, approvals, handoffs, or resource unavailability. In most processes, waiting time far exceeds active time and is the main contributor to long cycle times. Reducing waiting time delivers greater performance improvement than optimizing task execution speed.	
Cost	Total cost per unit or per period (labor, materials, overhead). **Example:** cost per processed invoice.
Waste (process)	Common types: waiting, overproduction, rework/defects, excess movement, excess processing, inventory/wip, unused talent **Example:** repeating kyc verification because data isn't shared.
Efficiency	doing the same work with less waste/resources (time/cost).
Productivity	output per unit input (how much you produce/service per time/person).
Explain efficiency vs productivity using metrics (throughput, cycle time, cost, waste). Efficiency = doing work with minimum waste (time, cost, resources). Productivity = output produced per input (e.g., orders/hour per staff). Metrics: throughput (units/time), cycle time (start-to-finish time), cost per unit, and waste (waiting, rework, motion, overprocessing). A process can be "productive" short-term (people working fast) but inefficient (high rework, high overtime). Measuring these kpis reveals whether improvement is real or cosmetic.	
Efficiency means achieving maximum output with minimum resources. **Inefficiency** exists when processes consume more time, money, or effort than necessary. **Productivity** refers to how much useful output is generated. **Unproductivity**	

indicates wasted effort that does not contribute to organizational goals. In bpr, we are not guessing which process is inefficient. We measure it systematically by analyzing processes and their outcomes	
Why efficiency ≠ productivity? A process can be "efficient" internally but still produce low output. BPR tries to improve outcomes at system level. Sometimes improving efficiency at one step reduces throughput (productivity) due to new bottlenecks. Example: If one cashier becomes super fast at taking orders (efficiency), but only one barista makes drinks, you create a longer queue at drink prep. Productivity of the whole system didn't improve.	
ROI	ROI is the return on investment, but in BPR you start with the merit function (what you're optimizing). Then you translate improvements into financial impact less churn, less rework, more throughput, more customers
Sometimes "customer satisfaction" doesn't directly mean more customers. But it can mean less loss, less churn, repeat business. So you must connect it logically. **Example (Careem):** They paid drivers heavily (loss) to ensure availability (objective: immediate ride matching). That's ROI thinking based on building a system, not instant profit.	
Capex	one-time investment (buy kiosks, build system)
Opex	monthly running cost (maintenance, support, staff cost)
You can tolerate a capex mistake once, but opex keeps eating profit monthly. **Example:** If kiosks reduce staff need by half, but add high monthly support cost, the "new process" might not actually be cheaper.	
Define capex vs opex and why executives worry about opex accuracy. Capex is one-time investment to acquire/build an asset (e.g., kiosk purchase + setup). Opex is recurring monthly running cost (maintenance, licenses, support). Your note: companies are strict that opex must not be misestimated because it directly eats profit month after month. In business evaluation, capex can be justified via roi, but uncontrolled opex creates permanent margin damage—so it's a key merit-function input.	
In a starbucks-like ordering process, propose 2 kpis and explain what they reveal. Two strong kpis: (1) average cycle time from order to pickup; (2) % orders meeting promised time (service-level compliance). Cycle time reveals waiting + bottlenecks (queue, payment, preparation). Service-level compliance shows reliability and variation (important for loyalty). Digitally enabling "order ahead" also ties to engagement/retention kpis like repeat orders or loyalty usage (conversion and retention goals).	
Airport agent → kiosk redesign: write the "before vs after" cost logic and interpret it. Your notes: before, cost per customer is agent payroll cost spread across monthly demand: Before cost ≈ (na × xctc) / (30 × nf × cf). After adding a kiosk, agent service time halves but kiosk adds monthly opex: After ≈ ((na × xctc)/2 + opex) / (30 × nf × cf), plus one-time capex. If after-cost ≤ before-cost and payback covers capex, redesign is justified	

	Meaning	Formula	Bpr use	Example
Cycle time (ct)	Total time from start to finish for one unit. Includes work time + waiting + delays.	Ct = end time − start time	Main "speed" merit function; exposes waiting/approvals.	As-is 12 min → to-be 7 min = 41.7% reduction. From order placed to order received = 12 minutes.
Throughput (tp)	Output rate (how many cases/orders per time). Number of units completed per time period (customers / hour, applications/day).	Tp = units completed / time	Improves capacity without adding resources; bottleneck controls tp.	30 orders/hour → 45 orders/hour = +50%. 120 orders/hour.
Cost per transaction (cpt)	Cost to process one case/order.	Cpt = total process cost / #transactions	Connects redesign to finance (salary + rework + delays).	Rs 120/order → rs 80/order = rs 40 saved each.
Defect rate	% outcomes needing rework/corrections.	Defect% = defects / total × 100	Quality merit function; links to six sigma. (error rate)	18 defects/200 = 9%.
Rework cost	Time/cost spent fixing mistakes.	Rework cost = rework hours × hourly cost	Shows hidden waste; usually caused by handoffs + unclear data. (rework time)	10 hrs/week × rs 500/hr = rs 5,000/week.
First pass yield (fpy)	% cases done right the first time (no rework).	Fpy = good units / total units	Strong quality merit function; improves speed + cost together.	180/200 = 0.90 (90%).

Customer value	"how satisfied is the customer?" (speed, ease, reliability).		Connects process redesign to customer benefit/utility. Utility (proxy)	Waiting time ↓ → utility ↑ (repeat purchase ↑).
Cash flow metric	How fast money is collected/paid.		Shows *business perspective*: process impacts liquidity. (a/r days or payables days)	A/r collection 45 days → 30 days improves cash availability.
Break-even / roi	Is redesign worth it financially?	Roi = (gain − cost)/cost Payback = investment / annual savings	(payback (for redesign investment))	Invest 200k, save 100k/year → payback **2 years**.
Bottleneck utilization	Where time is being lost; "bottleneck = problem in process."		Focus redesign on bottleneck, not random steps. (waiting time share)	If 70% time is waiting for approval → redesign approval logic.

	Problem	Process flow	What was obliterated	Technology used
Banking & financial services (branchless banking)	Traditional banking required customers to visit physical branches for deposits, transfers, and bill payments → long queues, high cost, poor reach.	Physical presence was assumed necessary Manual verification + paperwork High fixed cost (branches, staff)	Physical branch dependency Paper-based verification Manual cash handling workflows	Mobile wallets (easypaisa, jazzcash, m-pesa) Agent networks + sms/ussd Digital kyc
	The entire service delivery model changed, not just faster counters. Banking moved from branch-centric → customer-centric, enabling financial inclusion. Branchless banking did not automate branches; it eliminated the need for branches. Eliminate branches, don't digitize queues.			
Retail / coffee chains (starbucks - type model)	Long queues, order errors, peak-hour congestion, inconsistent service.	Sequential flow: order → pay → wait → serve Cashier as bottleneck Verbal order transmission (high variation)	Physical ordering dependency Verbal communication of orders Manual queue sequencing	Mobile ordering apps Real-time pos + kitchen display Inventory synchronization
	The ordering step moved outside the store, shrinking in-store time and removing the bottleneck. Technology enabled bpr only after the ordering logic itself was redesigned. Remove ordering dependency, don't speed cashiers.			
Manufacturing (tesla giga press / toyota-style logic)	Complex assembly lines with hundreds of parts, welds, inspections.	Too many sub-assemblies Multiple handoffs High defect probability	Dozens of assembly steps Intermediate inspections Excessive part interfaces	Single-piece casting (giga press) Advanced materials & simulation
	Instead of automating welding faster, tesla removed welding altogether for that component. Bpr replaced process complexity with design simplicity. Remove assembly steps, don't robotize welds.			
Education (process, not content)	Students pass exams but don't retain understanding.	One-way lectures Assessment only at end No feedback loop	Pure lecture-based delivery Delayed feedback	Live quizzes Lms dashboards Analytics on engagement
	Learning became a feedback loop, not a content dump. Education bpr redesigns learning flow, not just digitizes notes. Redesign learning loops, don't upload pdfs.			
Healthcare (appointments & diagnostics)	Patients wait hours for consultations and test results.	Sequential approvals Paper files moving physically Repeated data collection	Physical file movement Duplicate patient intake Doctor-dependent scheduling	Electronic health records (ehr) Online appointments Diagnostic integration
	Workflows were redesigned around data availability, not hospital departments. Healthcare bpr follows data flow, not reveal hierarchical flow. Remove file movement, don't computerize it.			

Logistics & e-commerce (order fulfillment)	Slow deliveries, lost orders, poor tracking.	Manual tracking Batch updates Poor visibility	End-of-day reconciliation Manual status updates	Real-time tracking Warehouse automation Integrated erp
	Customers now see the process as it happens, changing expectations and trust. Transparency itself became a process output.			

	What they did	Why it failed	Lesson	Exam line
Government offices (digitized forms, same pain)	Scanned paper forms into pdfs and put them online.	Same approvals Same delays Same silos	Digitization ≠ digitalization ≠ bpr.	They automated paperwork but preserved bureaucracy. Remove approvals, don't scan forms.
Banks installing more counters	Added more teller systems and screens.	Same queue logic, same verification rules, same branch despendency	Capacity increase without redesign shifts the bottleneck, doesn't remove it.	They optimized locally instead of redesigning globally.
Universities using lms only for uploads	Used lms to upload pdfs.	No feedback loops No engagement redesign Learning remained passive	Technology without pedagogical redesign adds no value.	
Erp implementation without process change	Installed erp but kept osld approvals and rules.	Complex screens Resistance Workarounds	Erp exposes bad processes faster—but doesn't fix them.	Erp magnifies process flaws instead of curing them.
Call centers with scripts only	Used crm scripts to speed calls.	Root causes not fixed Repeat calls increased	Front-end automation without backend redesign increases rework.	