| | |
|---|---|
| **Process Redesign** | Identify possibilities for improving the design of a process<br>AS-IS: **<u>Descriptive</u>** modelling of the real world<br>TO-BE: **<u>Prescriptive</u>** modelling of the real world<br>No silver-bullet: requires **<u>creativity</u>**<br>**Redesign heuristics** can be used to generate ideas |

**Process redesign approaches**

| | |
|---|---|
| **Exploitative Redesign (transactional)** | • **<u>Doesn't</u>** put into question the current process structure<br>• Seeks to identify problems and resolve them **<u>incrementally</u>**, one step at a time<br>• Example: **Heuristic redesign** |
| **Explorative Redesign (transformational)** | • Puts into question the **fundamental assumption**s and **principles of the existing process structure**<br>• Aims to **achieve breakthrough innovation**<br>• Example: <u>Business Process Reengineering (BPR</u>) |

**Business Process Reengineering (BPR)**

| | |
|---|---|
| **Transformative:** | Puts into question the fundamental assumptions of the "**as is**" process |
| **Analytical:** | Based on a set of principles that foster:<br>– Outcome-driven processes<br>– Integration of information gathering, work and decisions |

**The Ford Case Study**

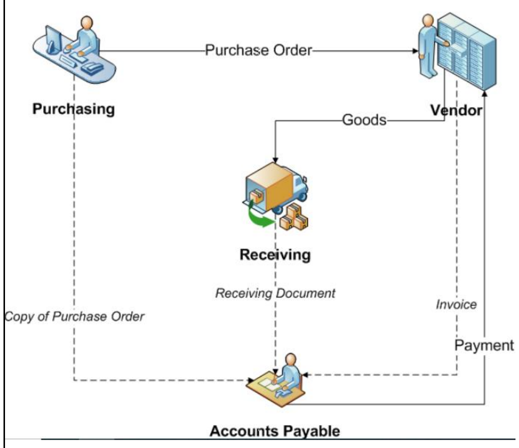
**Ford needed to review its procurement process to:**

**•Do it <u>cheaper</u> (cut costs)**

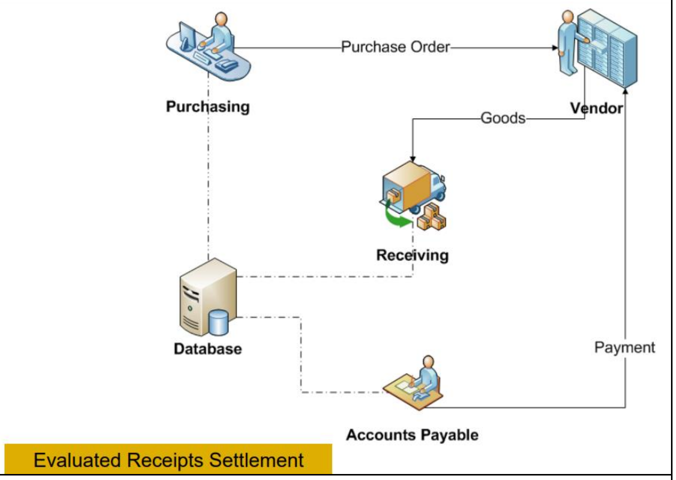
**•Do it <u>faster</u> (reduce turnaround times)**

**•Do it <u>better</u> (reduce error rates)**

How the process worked? ("as is")



Reengineered Process ("to be")



**Some principles of BPR**

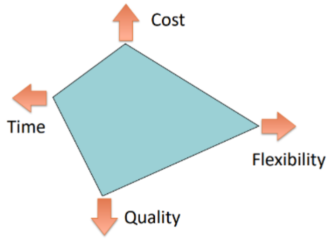
**1. Capture information once and at the source**

**2. Subsume information-processing work into the real work that produces the information**

**3. Have those who use the output of the process drive the process**

**4. Put the decision point where the work is performed, and build control into the process**

**5. Treat geographically dispersed resources as though they were centralized.**

| | |
|---|---|
| **Principle 1** | Capture information once and at the source<br>• **Shared data store** –All process workers access the same data –Don't send around data, share it!<br>• **Self-service** –Customers capture data themselves –Customers perform tasks themselves (e.g. collect documents) |
| **Principle 2** | Subsume information- processing work into the real work<br>• Evaluated receipt settlement: when receiving the products, record the fulfillment of the PO, which triggers payment |
| **Principle 3** | Have those who use the output of the process drive the process<br>• Vendor-managed inventory<br>• Scan-based trading<br>• Push work to the actor that has the incentive to do it |
| **Principle 4** | Put the decision point where the work is performed, and build control into the process<br>• Empower the process workers<br>• Provide process workers with information needed to make decisions themselves<br>• Replace back-and-forth handovers between workers and managers (transportation waste) with well-designed controls |
| **Principle 5** | Treat geographically dispersed resources as though they were centralized.<br>• If same people perform the same function in different locations, integrate and share their work wherever possible<br>• Larger resource pools ◊ less waiting times even with relatively high resource utilization |
| **Self-service-based redesign** | |
| **Principles 1 & 2** | When equipment is needed, site engineer queries the suppliers' catalogue, selects equipment and triggers PO |
| **Principle 3** | Supplier stocks frequently used equipment at construction site, site engineers scan to put them into use |
| **Principle 4** | Site engineer is empowered with the authority to rent the equipment; works engineer performs statistical controls |
| | |
| **Heuristic process redesign** | |
| **Transactional:** | changes the "as is" process incrementally |
| **Inward-looking:** | operates within the scope and context of "as is" process |
| **Analytical:** | based on redesign heuristics that strike tradeoffs between:<br>• Cost • Time • Quality • Flexibility<br>Performance measures: the Devil's Quadrangle<br><br> |

| | |
|---|---|
| **Flexibility** | **Ability** to react to changes in:<br>• Workload<br>• Customer demands and expectations<br>• Resource and business partner availability and performance<br>• **Example**: Following natural disasters (e.g. storms), the number of home insurance claims increases by tenfold<br>To address this surge, flexibility is required at:<br>• **Resource level:** Staff redeployment, faster performance<br>• **Process level:** Performing tasks differently to speed up the front-end<br>• **Management**: Relaxing business rules and controls where possible |
| **Redesign heuristics** | |
| **Task-level** | • Task elimination<br>• Task composition/decomposition<br>• Triage |
| **Flow-level** | • Re-sequencing<br>• Parallelism enhancement |
| **Process-level** | • Specialization & standardization<br>• Resource optimization<br>• Communication optimization<br>• Automation |
| **Task-level** | |
| **H1. Task elimination (T+, C+/-, Q-)** | Consider trade-off between the cost of the check and the cost of not doing it<br>**Examples:**<br>• **Procure-to-pay process:** some types of employees are empowered to trigger isolated purchases below $500 without supervisor approval<br>• **Order-to-cash process:** invoices from trusted suppliers under $1000 are not checked on a one-by-one basis<br>• **University admission process:** authenticity check is very expensive, yet it leads to only 1% of applications being rejected |
| **H2. Task composition and decomposition** | Composition example: (T+, C+/-, F+)<br>• **Procure-to-pay process:** Merging two checks: “Check necessity of purchase” and “Check budget”<br>Decomposition example: (T-, C+, F-)<br>**• Make-to-order process:** Separate a single thick “prepare quote” task into “prepare bill of materials”, “prepare production plan” and “estimate costs and delivery time” |
| **H3. Triage** | • **Specialize a task:** divide a general task into two or more **alternative** tasks<br>• **Generalize tasks:** integrate two or more alternative tasks into one general task<br>Specialization example: (T+, C+/-, F-)<br>• **Procure-to-pay process:** Separate approvals of small purchases, medium purchases and large purchases<br>Generalization example: (T-, C+/-, F+)<br>• **Make-to-order process:** Integrate quote preparation for two product lines into one single task |

| **Flow-level** | |
|---|---|
| **H4 Re-sequencing** | Re-order tasks according to their cost/effect ratio to minimize over-processing<br>**Examples:**<br>• **Make-to-order process:** If “Prepare production plan” is time-consuming, postpone it until after the quote price has been tentatively accepted by the customer<br>• **Procure-to-pay process:** If “Check necessity of purchase” leads to 20% of knock-outs and “Check budget” leads to 2%, perform “Check necessity of purchase” first<br>• **University admission process:** authenticity check (very slow) leads to 1% of applications being rejected while committee’s check leads to 80% of applications being rejected. Put committee’s check first |
| **H5. Parallelism enhancement (T+,C-)** | Parallelize tasks where possible in order to reduce cycle time<br>**Examples**:<br>**• Procure-to-pay process**: Parallelize “Approve budget” and “Approve necessity of purchase”<br>**• Make-to-order process:** After “Prepare bill of materials”, perform “Prepare production plan” and “Estimate costs” in parallel |
| **H6. Process specialization/standardization** | **Process specialization**<br>• One process is split into multiple ones: by customer class, by geographic location, by time period (winter, summer), etc.<br>• Resources are split accordingly<br>Process standardization<br>• Two processes are integrated • Resources are pooled together<br>Specialization example: (C+/-, Q+/-, F-)<br>• Procure-to-pay process: One process for Direct procurement (e.g. raw materials) and one for Indirect procurement (MRO - Maintenance, Repair and Operations)<br>• **Claims handling process:** One claims handling process for the summer season (stormy season - peak) and one for the winter season (off-peak)<br>Standardization example: (C+, Q+/-, F+0029<br>• Claims handling process: Integrate claims handling for motor insurance across different brands of a group |
| **H7. Resource optimization (T+, C+, F+/-)** | • Use resources of a given type as if they were in one room<br>• Let people do work that they are good at<br>• When allocating work to resources, consider the flexibility in the near future<br>• Avoid setups as much as possible<br>**Resource integration example:**<br>• Claims handling process: Share resources across different types of claims (e.g. motor and personal insurance)<br>**Batching example:**<br>• Claims handling process: Batch all claims for a given geographic area and assign them to the same resources • University admission process: Batch all applications and handle them to the assessment committee |

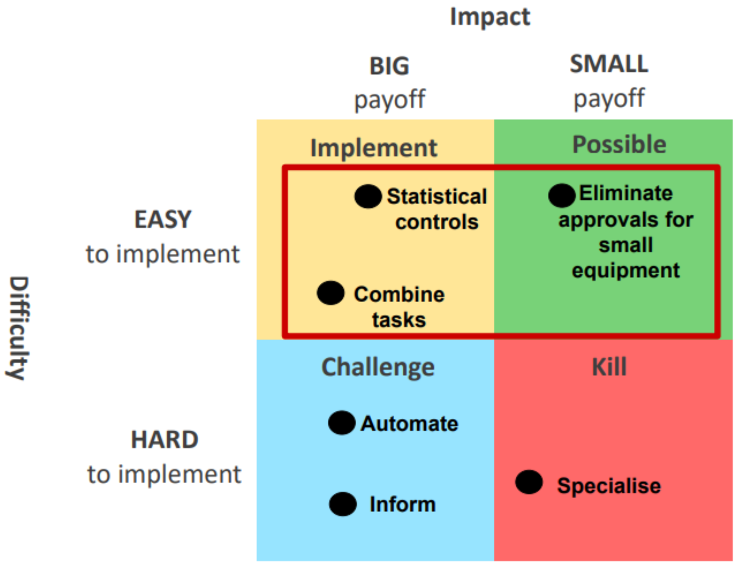
| | |
|---|---|
| **H8. Communication optimization (T+,Q+,C+/-,F-)** | • Automate handling, recording and organization of messages<br>• Monitor customer interactions, record exceptions<br>**<u>Optimize</u>**<br>1. Number of interactions with customers and business partners<br>2. Type of interaction (synchronous vs. asynchronous)<br>3. Timing of interactions |
| **1. Optimize number of interactions** | • Gather sufficient information to get to the next milestone (reduce external interactions) |
| **2. Optimize type of interaction** | • Synchronous interactions effective to resolve minor defects<br>• Asynchronous to notify, inform, resolve major defects, request additional information to reach next milestone |
| **3. Optimize timing of interactions:** | • **Front-loaded process:** bulk of information exchange and processing happens upfront<br>• <u>Complete-kit concept</u><br>• **Back-loaded process**: bulk of information exchange and processing happens downstream<br>• <u>Example: CVS Pharmacy in early 2000s</u> |
| **Complete-Kit Concept:** | "Work should not begin until all pieces necessary to complete the job are available" |
| **Principles for complete-kit process design:** | • Provide complete and easy-to-follow instructions for those who will initiate the process.<br>• If a process cannot start, the client should be notified of all defects that could be reasonably identified at the onset of the process<br>• Consider the tradeoff between "incomplete-kit" process initiation vs. roundtrip to revise and resubmit a request |
| **H9. Automation (T+,C+/-, Q+/-, F-)** | 1. Use data sharing (Intranets, packaged enterprise systems) to:<br>• Increase availability of information to improve visibility and decision making (subject to security/privacy requirements)<br>• Avoid duplicate data entry and transportation<br>2. Use network technology to:<br>• Replace physical flow (e.g. paper documents) with information flow<br>• Enable self-service via e.g. online forms and Web data services<br>3. Use tracking technology to identify and locate materials and resources<br>4. Use business rules technology to automate information processing tasks (including decisions)<br>5. Automate end-to-end processes with a dedicated BPM system or system with process automation functionality |
| | |
| **Two sides of the BPM story** | |
| **Conceptual "to-be" process models** | • are made by domain experts<br>• provide a basis for communication amongst relevant stakeholders<br>• must be understandable<br>• must be intuitive and may leave room for interpretation<br>• contain purely a relevant set of process information |

| | |
|---|---|
| **Executable process models** | • are made by IT experts • provide input to a process enactment system - BPMS • must be machine readable • must be unambiguous and should not contain any uncertainties • contain further details that are only relevant to implementation |
| | |
| **Bridging the gap: A five-step method** | 1. Identify the automation boundaries<br>2. Review manual tasks<br>3. Complete the process model<br>4. Adjust task granularity<br>5. Specify execution properties |
| **1. Identify the automation boundaries** | Principle: not all parts of a process can be automated |
| **2. Review manual tasks** | Principle: if it can't be seen by the BPMS, it doesn't exist. |
| **3. Complete the process model** | **Principle:** exceptions are the rule.<br>• Consider incomplete paths<br>• Rules of thumb<br>**Principle:** no data = no decisions, no tasks handover.<br>• Specify all (electronic) business objects<br>• For each task, determine which business objects it creates, reads, updates, delete (CRUD)<br>• For each decision, determine which objects it needs |
| **4. Adjust task granularity** | Principle: BPMSs add value if they coordinate handovers of work between resources. |
| **5. Specify execution properties** | -> Process variables, messages, signals, errors<br>-> Task and event variables and their mappings to process variables<br>-> Service details<br>-> Code snippets<br>-> Participant assignment rules and user interface structure<br>-> Task, event and sequence flow expressions<br>-> BPMS-specific: work queues, forms, connectors… |
| | |
| **Execution Engine** | • Instantiates executable process models (also called "cases")<br>• Orchestrates distribution of work items to process participants and software services in order to execute a business process from start to end<br>• Logs execution data |
| **Administration & Monitoring Tools** | • To manage automation solutions<br>• To configure access to system components<br>• To monitor participants availability and performance of process cases |
| **External Services** | • Expose a service interface with which the engine can interact<br>• The engine provides the invoked service with the necessary data it will need to perform the activity for a specific case<br>• Examples: rules engine, email or Twitter notification, DB connector, CRM connector… |

| BPMS Landscape | |
|---|---|
| **Big vendors** | • IBM BPM • Oracle BPMS • Microsoft BizTalk, Wf • SAP NetWeaver BPM • Software AG webMethods • Pagaystems PegaRULES |
| **Other closed-source** | • Appian BPMS • BizAgi BPM Suite • Bosch inubit Suite • OpenText BPM • Perceptive BPMONe • Progress Savvion (cloud) • Effektif (cloud) |
| **Commercial open-source** | • Bonita Open Solution • Camunda Fox • Intalio\|BPM • JBoss jBPM • Shark • YAWL |

## Prioritizing redesign options

### PICK chart



Impact

**BIG** payoff

**SMALL** payoff

Difficulty

**EASY** to implement

**HARD** to implement

Implement

● Statistical controls

● Combine tasks

Possible

● Eliminate approvals for small equipment

Challenge

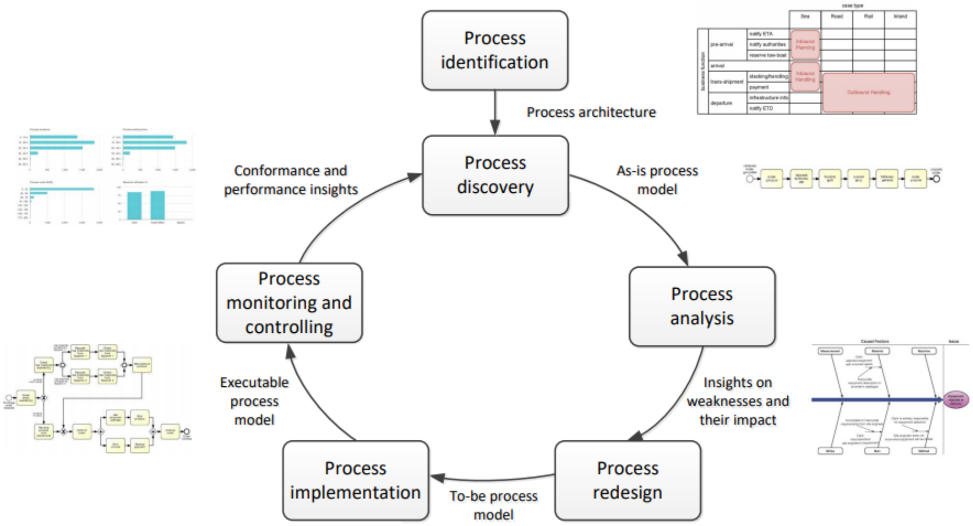
● Automate

● Inform

Kill

● Specialise

## The BPM lifecycle



Process identification

Process architecture

Process discovery

As-is process model

Process analysis

Insights on weaknesses and their impact

Process redesign

To-be process model

Process implementation

Executable process model

Process monitoring and controlling

Conformance and performance insights