OSS Business Case Builder

Version 1.3

PASSIONATE ABOUT OSS

SHARING OUR PASSION
FOR OPERATION AND BUSINESS
SUPPORT SYSTEMS

Contents

1. Introduction

So, you've spotted an opportunity to help your organisation meet strategic objectives through the implementation of an OSS (Operational Support System) or BSS (Business Support System) project and feel compelled to make it happen. [Note that this document will refer to these systems as the singular OSS to represent the plural OSS/BSS for simplicity of language].

Can you clearly articulate the problem/s you will solve or the opportunity/ies you will seize?

The OSS Business Case is the fulcrum between business benefits (project sponsors) and technological change (project implementers). The business case must quantify the growth, transformation and critical advantages that will be derived by the organisation. It also presents how to compare, then leverage the best of the people, process and technical alternatives that you've analysed.

After identifying the objective, the biggest difficulty in gaining sponsor buy-in is identifying the benefits. Unfortunately, many of the benefits of an OSS can appear intangible (ie indirect Return on Investment [ROI] rather than direct ROI). This makes quantification and justification of the business case difficult!

The primary aim of this template is to provide various approaches to justify the outlay of resources on your next OctopOSS[1]. **Section 3.4 below provides a list of business case justification factors that you can build your case around**. Executive management sets the tone of the business and key themes. Your business case must unify your idea with the organisation's key themes and/or objectives.

The Business Case must be easily understandable and be able to promote the concept throughout your organisation, articulating the project's merits to senior management, stakeholders, users, customers, partners, integrators and beyond. Chances are that you're part of your organisation's operations, IT or OSS business unit, but please don't underestimate the reach that your OSS has beyond your business unit. Consider that each impacted department can have its own culture and tolerance to change.

OSS Business Cases have traditionally been inward facing in that they are produced by technically-focussed Business Units to meet their own needs and the outcome becomes technology-driven. In the future, OSS business cases should be driven by the needs of other business units as much as the OSS business unit.

For example, say your marketing department has identified an opportunity to derive corporate advantage through an ability to offer bundled services. The OSS must undergo enhancement to support this initiative and the new Business Case must be built around the marketing department's objectives / benefits.

OSS Business Cases are often full of an overkill of facts and figures, but fail to provide an action plan for how the objective will be achieved. Your business case on the other hand, will be highly action-oriented.

[1] I refer to an OSS project as an OctopOSS because no matter how many tentacles you have under control, there is always another one ready to whack you on the back of the head.

2. Business Case Template

The most important parts of this document appear in the business case justification sections, where we help you to build a compelling business justification to start your OSS project. However, before that we'll start by providing a framework in which to develop that justification.

2.1. Framework

In the unlikely event that your organisation doesn't have its own business case template, then the ***"Developing a Business Case: Expert Solutions to Everyday Challenges"*** by Harvard Business Review Press provides a useful framework as follows:

- **Step 1: Define the opportunity**. Describe the situation and the business objectives that your proposal will affect. (refer to section 3)
- **Step 2: Identify the alternatives**. Brainstorm multiple approaches and then choose three to four to analyze.
- **Step 3: Gather data and estimate time frame**. Obtain information about each alternative and estimate how long each option will take to implement.
- **Step 4: Analyze the alternatives**. Analyze how your options will affect the business objectives you've defined.
- **Step 5: Make a choice and assess the risk**. Make a recommendation based on your analyses, and consider how you will mitigate any risks associated with your recommendation.
- **Step 6: Create a plan for implementing your idea**. Identify, at a high level, how you will achieve your goals and who will be accountable for each milestone. Spell out when you expect to see benefits.
- **Step 7: Communicate your case**. Create a document or a presentation, or both, to sell your recommendation to decision makers.

There are many other great resources to help you build a Business Case if the framework above from HBR Press doesn't fulfil all of your needs.

For example, there is one item that I think is missing from HBRs framework:

- **Step 0: Objective / Intent**. Start with a simple statement of what you're aiming to achieve, but more importantly, WHY you're trying to achieve it. This should be only a sentence or two long and becomes the guiding vision for the project.

2.2. Additional Notes

As noted, your organisation probably already has its own business case template, with its own specific format. This OSS Business Case Builder is not attempting to be THE template, but helps to add strength to any framework your own OSS Business Case must build upon.

The OSS justifications (section 3.4) will often need to be translated into quantitative data to add further strength to your story. We'll discuss numerical analysis later, but you will need to determine the level of complexity / granularity of your data gathering exercise.

If in doubt, err on the side of simplicity and keep supporting details for appendices.

3. Step 1. Define the Opportunity

3.1. Don't forget Step 0. Define the Objective / Intent

Describe the 1-2 sentence statement that concisely defines what you're aiming to achieve and why.

3.2. Business Alignment

The Business Case must clearly align the technical solution with desired business outcomes, so the following questions will help to identify what is most important:

1. Do you have a clear understanding of what your corporate vision and competitive advantage (VCA) is
2. In what areas do your current management tools influence that VCA, either positively or negatively
3. What areas of the business are most reliant on OSS tools
4. What areas of the business consume the most work-hours with OSS or similar tools
5. In what areas will OSS efficiency improvements deliver the greatest outcomes (eg faster service activations, faster revenue turn-on, reduced fault rectification times, reduction in head-count, etc)
6. What additional functionality would deliver the greatest outcomes (eg sales target identification, fault trend analysis / predictors, network capacity planning simulators, etc)

> The questions above should consider not just the current state but the desired future state, particularly if your OctopOSS project will be large enough to have direct impacts on the future state.

3.3. Capturing Initial Requirements

The most important part of OSS requirement capture is to focus on **outcomes / value rather than functionality**. The vendor solutions on the market offer a vast array of functionality, bells and whistles. This makes vendor evaluation difficult as there are no easy "apples-for-apples" comparisons. Vendors tend to focus on the functionality that they can offer in an ever-increasing arms-race of functionality. But your job is to identify the benefits that the solutions offer and prioritise against your organisation's needs.

The aim is to create a guiding vision for what the OSS is to achieve. To borrow from Pareto's 80:20 principle, in most cases the key to capturing a business's OSS requirements is to identify the 20 percent of high-value actions that deliver 80 percent of benefits to the organisation. But more importantly, the requirement capture phase should focus on optimisation of those 20 percent of high-value actions to get your OSS into production faster and cheaper.

We recommend to use two techniques to help identify what's most important on your OSS project:

- The Long-tail Diagram
- The Whale Curve

The **Long-Tail Diagram** below comes from an actual client's functionality usage profile.

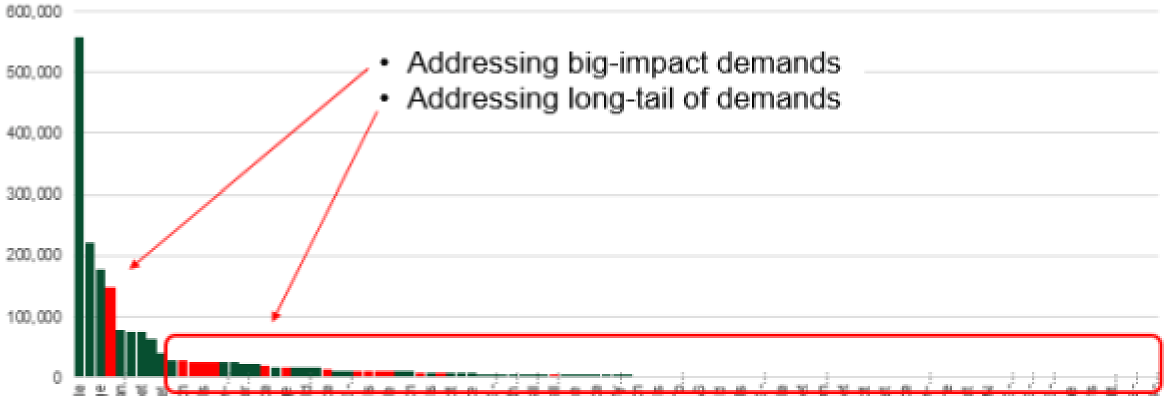


Figure 1 - OSS Functionality Long-Tail Diagram

The x-axis shows a list of functionalities / use-cases. The y-axis shows the number of uses of the function (the y-axis could equally represent usefulness or value or other scaling factor of your choice). In most cases, the big-impact demands on the left side of the graph are going to be important. They're the things that customers need most and use most. Try to not get distracted by all the product "differentiators" out at the far side of the graph as they tend to deliver little business value.

The diagram below shows **The Whale Curve**. It graphs the relative profitability of each product in an organisation's product mix.

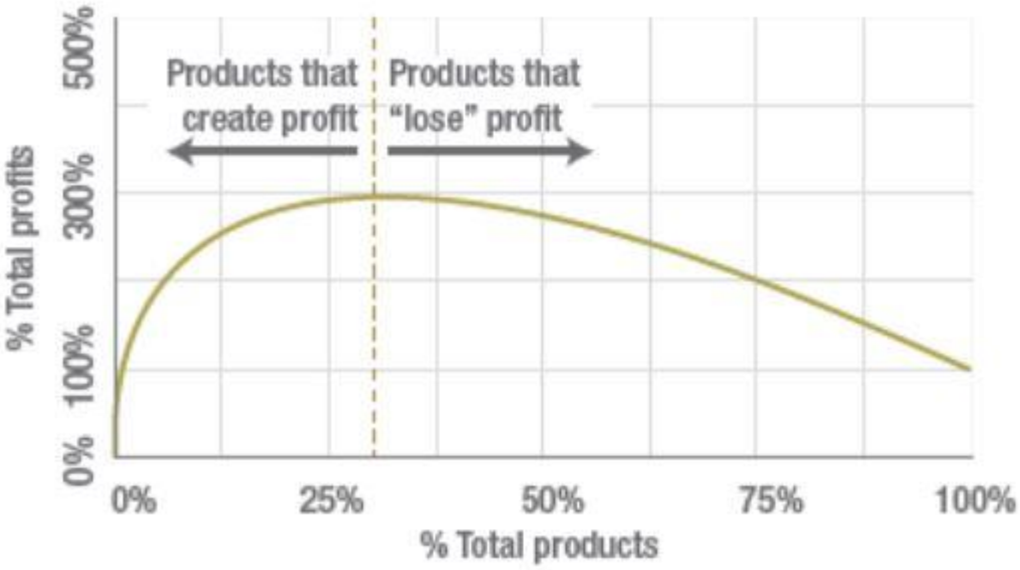


Figure 2 – Whale Curve Diagram From the book, "Waging War on Complexity Costs," by Steven A Wilson and Andrei Perumal.

You might be wondering how a profitability graph could ever peak at over 100%. After all, a company can only ever have a profitability of 100% of whatever its profits are, regardless of whether they're large or small. But what this graph shows is that 20-30% of the products (or customers or assets, etc) are generating up to 300% of an organisation's profits, whilst the remainder are actually destroying roughly 200% of profits.

You may want to consider the following business case justifications to make your business case more compelling and build the high-level requirements around that. At a later stage, we'll get to the detailed requirements capture and you may wish to use some of the requirement capture techniques / tools on PassionateAboutOSS.com, that include different options for different scenarios:

1. The traditional requirement capture approach
2. An alternate approach is to capture the required benefits.
3. The Persona / User-story approach
4. The Multi-Filter Approach

It is also important to capture the key business metrics (eg reliability, customer loyalty, number of truck rolls, response time, etc) that relate to the key business objectives and / or requirements (eg number of orders processed per day, number of tickets closed within SLA targets).

Consider these KPIs from the perspective of your sponsor's desired outcomes.

3.4. Business Case Justifications

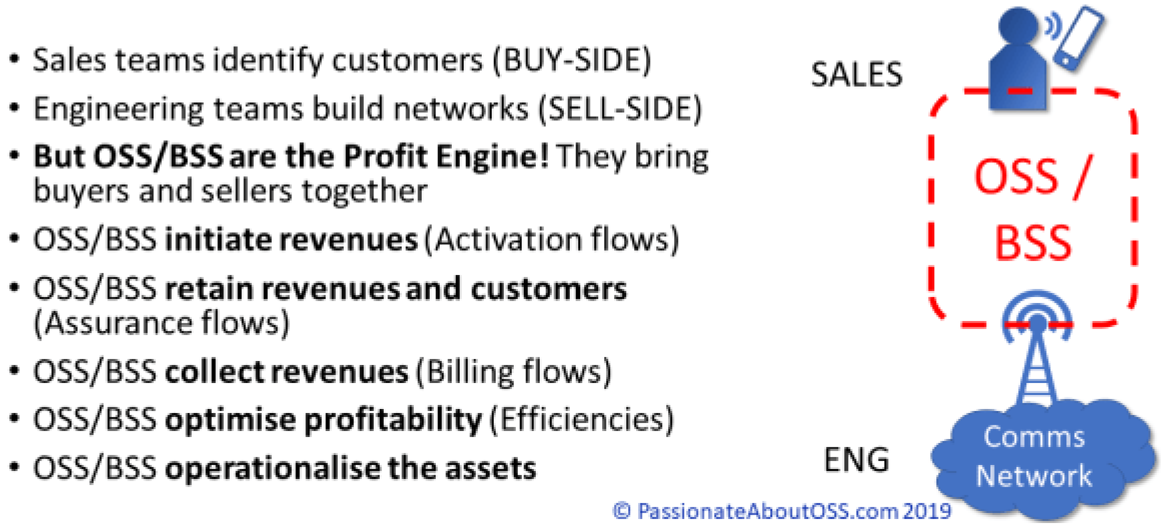
You might be highly technical in nature and orientation. You might have little interest in business cases and/or related justifications. If this is the case, consider this fact - You only get to work on exciting OSS projects if funding is allocated to them. Does that incentivise you to create a more compelling business case?

Your project sponsors might not be as technical as you. Does that incentivise you to create a persuasive argument using benefits rather than features / functions?

As indicated earlier, the real benefits of an OSS project are often intangible. Since the OSS business unit is rarely seen to be generating revenue, it can be difficult to build a Business Case that justifies the large capital outlays and ongoing operational costs.

However, that's a defeatist mind-set. Let's get a little more specific.

As we all know, OSS are the linchpin around which a service provider is built. If the network is ultimately the product for any network operator, then OSS/BSS are the great connectors, connecting customers to that product (ie the marketplace that connects "buyers" with "sellers" [the network]). Both for initial activation, but also ongoing utilisation of network resources.



- Sales teams identify customers (BUY-SIDE)
- Engineering teams build networks (SELL-SIDE)
- **But OSS/BSS are the Profit Engine!** They bring buyers and sellers together
- OSS/BSS **initiate revenues** (Activation flows)
- OSS/BSS **retain revenues and customers** (Assurance flows)
- OSS/BSS **collect revenues** (Billing flows)
- OSS/BSS **optimise profitability** (Efficiencies)
- OSS/BSS **operationalise the assets**

Figure 3 - OSS/BSS are the Profit Engine

However most executive sponsors don’t grasp the full extent of this dependence. There are a few reasons for this:

1. OSS are perceived as not generating any revenue directly (incorrectly I might add!!)
2. OSS projects continually disappoint (on time, cost and/or functionality)
3. The full picture of OSS functionality is hidden (like an iceberg) and is boring to most. The Service provider’s customers certainly have zero interest in the back-end systems / functionality in use
4. Our business cases tend to be built around cost reduction, not business growth. Let’s be honest, OSS cost reductions go in the same excitement bucket as a reduction in paper clips, pens and printer paper
5. The technologies used by OSS seem to be obsoleted even faster than the network technologies they manage
6. They tend to deliver more negative messaging (eg network outages, fall-outs resulting in poor customer experiences, etc), whilst others get the credits for any positive messaging that OSS have facilitated
7. Due to the costs of implementation and change, OSS are seen by many as giant sinkholes of cost centres
8. The list goes on!!

If we accept these statements as fact, why do we even bother working in OSS? Put simply, we don’t just accept them. We must better understand the justifications behind these put-downs above.

The power of our persuasion all comes down to the messaging going to project sponsors. Like Steve Jobs, we have just three things (to create the mindset to help build a compelling story for your executive sponsors):

- **Break “the sink-hole perspective” by generating revenues** – via APIs, via platforms, via metrics that show OSS contribution to revenues or anything else that shows real value, not just cost-out, to executive sponsors as well as the customers they’re trying to service. Use metrics to convert all project outcomes into dollar terms.

- **Invert the iceberg** – show just how much capability is hidden under the surface by mapping how dependant the organisation's positive outcomes are on its OSS (OSS as the puppet-master). All the sexy stuff (eg 5G, IoT, network virtualisation, etc) can only be monetised if our OSS pull all the strings to make them happen
- **Simplify** – the massive (and increasing) complexity in our business is boring to non-OSS people and an impediment in our ability to deliver exciting outcomes for everyone else. Just take Google – they have an incredibly complex tech-stack to run… but what does the customer interact with on their search page or via APIs? The complexity is heavily abstracted from the user (or in our case, the sponsor/s).

The business case justifications of OSS tend to fall into four categories:

- **Revenue increase** – the operationalisation and monetisation of an operator's assets
- **Cost reduction** – improving the operational efficiency of the operator
- **Insight generation** – by leveraging the valuable data that an OSS collects
- **Brand value** – this is a catch-all for many different ways an OSS can contribute to (or detract from) an operator's brand. Ce can break this down into:
    - Defence (eg reducing outages that may damage the operator's brand) or
    - Offence (eg faster time to market or activations that give the operator a competitive advantage)
- But there's one other special category that bears consideration – **threat minimisation (or risk avoidance)** – which probably has elements of each of the four points above. It is the insurance policy on people, processes, services and/or technologies that support the network that your OSS is managing. If we take a really macro-view of this, two of the biggest threats facing operators today are disruptive new business models and disruptive new products. Or, you could flip this on its head and see it as **opportunity maximisation** (if you're the one to capitalise on the disruptive opportunity first).

An operator's OSS can have a massive influence on the opportunity maximisation category. If an operator takes months to force urgent changes through its OSS, then it can't respond well to a disruptive threat. Similarly, opportunities / arbitrages only have a short window before the market responds, so a lack of OSS flexibility impacts an operator's ability to maximise opportunity.

Having an OSS with greater agility than competitors can be a more significant, sustainable market advantage than most people in telecommunications realise.

The following table is probably the most important section in this whole document. It contains many examples of business case justifications that you may wish to build your OSS case around (feel free to add your own factors too of course). It will be your ability to build a persuasive case around these examples that will drive project sponsors to invest, or not:

Description	Revenue Increase	Lower OPEX	Brand Value	Metrics that Matter
Faster				
• Operationalisation - Getting new products and network technologies to market faster (ie competitive advantage and faster revenue turn-on)	✓		✓	Time to Revenue, Market share increase
• Getting services turned on more quickly after a customer places an order (faster revenue turn-on and increased customer satisfaction)	✓		✓	Time to Revenue, Market share increase, Volume of Orders Processed
• Faster repair of faults (reduced service breaches as well as associated customer reimbursement and churn reduction)	✓		✓	Reduced refunds, Cash flow reduction avoidance
• Faster correlation of faults and root-cause identification (reduced time to process and suppress secondary alarms)		✓	✓	Reduced resource allocations, Reduced refunds, Market valuation
• Faster project implementations (faster activations and network build-out, which allows faster service turn-on and associated revenues)	✓		✓	Time to Revenue, Market share increase
• Faster identification of market opportunities	✓		✓	Market share increase, positive churn (increased cash flow), Speed of Insight
• Visibility of past / present / future states for faster forensic analysis (eg fault-fix, root-cause, incident handling)		✓	✓	Reduced resource allocations, Reduced refunds, Reduction in revenue loss
More reliable				
• Insurance of brand value			✓	Market share increase, Product price premium, Future earnings / cash flow, Market valuation
• Reduced service breaches and associated customer re-imbursement	✓		✓	Reduced refunds, Cash flow reduction avoidance, Market Valuation
• Mitigates customer churn	✓			Cash flow reduction avoidance
• Reduces maintenance effort		✓		Reduced truck rolls
More flexible				
• Ability to adapt to new business model			✓	Time to Revenue, Market share increase
• Ability to adapt to new services and/or service models (eg service bundling or service buckets)	✓		✓	Time to Revenue, Market share increase

Description	Revenue Increase	Lower OPEX	Brand Value	Metrics that Matter
• Ability to handle corporate change (eg organisational change, process change, etc) faster and cheaper		✓		Time to Revenue, Market share increase
• Ability to adapt to competitive change		✓		Time to Revenue, New Revenue Stream, Market Share Increase
• Ability to adapt to new technology innovations (eg new network technologies, new interface types, etc)		✓		New Revenue Stream, Market Share Increase
• Ability to provide revenue-generating services (eg OSS APIs / microservices)	✓			New Revenue Stream, Market Share Increase
More secure				
• Increases data privacy (ie insurance of brand value)			✓	Market Valuation
• Identifies fraudulent activities (eg reduction in revenue leakage)	✓		✓	Reduced revenue leakage, Increased billable capacity, Reputation loss avoidance
• Increases network / service reliability	✓	✓	✓	Increased billable capacity, Reduced incident-fix resource allocation, Reputation loss avoidance
More automated				
• Automation of high-volume activities (activations)		✓		Reduced resource allocation, Faster revenue turn-on
• Automation of high-volume activities (fault-fix)		✓		Reduced resource allocation, Faster repair time
• Improved consistency and repeatability of tasks	✓	✓	✓	Reduced re-work and fallouts, Efficiency gains from continual process improvement, Improved customer sentiment
• Self-healing networks	✓	✓	✓	Reduced resource allocation, Increased billable time
Improved Data Quality				
• Reduced re-work		✓		Reduced resource allocation (design & audit), Faster revenue turn-on, Reduced truck-rolls
• Less data synchronisation complexity		✓		Reduced resource allocation, Faster repair time
More measurable				

Description	Revenue Increase	Lower OPEX	Brand Value	Metrics that Matter
• Increases visibility / tracing / reporting of network or service events, which allows faster identification of revenue-loss situations	✓		✓	Reduced revenue loss, Increased billable capacity, Reputation loss avoidance
• Tracks the status of your assets in near real time (ie more optimal utilisation of available capacity and better capacity planning)? Reduction in network results in	✓	✓		Increased billable capacity, Reduces capital and recurring expense
• Better supply chain planning	✓	✓		New revenue streams, Increased billable capacity, Reduces capital and recurring expense
• Optimal allocation of assets / inventory (and better use of working capital)		✓		Increased billable capacity, Reduces capital and recurring expense
• Minimisation of asset audits		✓		Reduced OPEX
• Easier tracking of the asset list for financial purposes (eg Fixed Asset Registers, depreciation, asset refresh, etc)		✓		Optimise asset allocation, Reduces capital and recurring expense, Better ROIC
• Identification of stranded assets	✓	✓		Optimise asset allocation, Reduces capital and recurring expense, Better ROIC
• Trend analysis to produce better operational insights		✓		Reduced OPEX, Improved efficiency
• Better analysis of routine maintenance to improve network reliability and stock control		✓		Increased billable capacity, Reduced revenue loss, Reduced truck-rolls, Lower MTBF
• Identification of process improvements		✓		Reduced OPEX, Improved efficiency
• Identification of optimal allocation of future investment	✓	✓		Optimise asset allocation, Reduces capital and recurring expense, Better ROIC, Additional revenue streams
• Identification of market opportunities	✓		✓	Market share increase, positive churn (increased cash flow)
• Delivering insights to other departments (eg marketing) that they would not have been otherwise able to identify	✓			Additional revenue streams

Table 1 - Benefits / Justifications for your Project

Notes:

- "Revenue Increase" includes items such as increased market share, revenues from new services, improved service availability, improved market intelligence, increased capital efficiency, etc.
- "Lower Opex" includes items such as reduced operating costs, reduction of the cost of goods sold (eg via streamlined inventory handling), more efficient processing, reduced head-count, simpler availability of data to reduce reporting effort, etc.
- The MTM column (Metrics That Matter) describes the types of metrics for this category that you can build financial benefit models around

3.4.1. Calculating the Metrics that Matter

The main objective of the Metrics that Matter (MTM) is to generate rules of thumb that provide estimated dollar amounts. An example is provided below.

MTM#1	Truck Rolls Per Service Activation
Volume	2,000 activations per day
Rule of Thumb	$1,000 / truck roll

	Current State	Future State
MTM#1	1.57	1.25
Volume	2,000	2,000
Truck rolls /day	3,140	2,500
Rule of Thumb	$1,000	$1,000
Cost / day	$3,140,000	$2,500,000
Savings / day		**$640,000**

Table 2 – Calculating the Metrics that Matter on your Project (Sample)

As shown in Figure 1 above, we want to identify the actions with the largest volumes (left side of the long-tail diagram). These tend to be the metrics that drive the most significant opportunities for transformation.

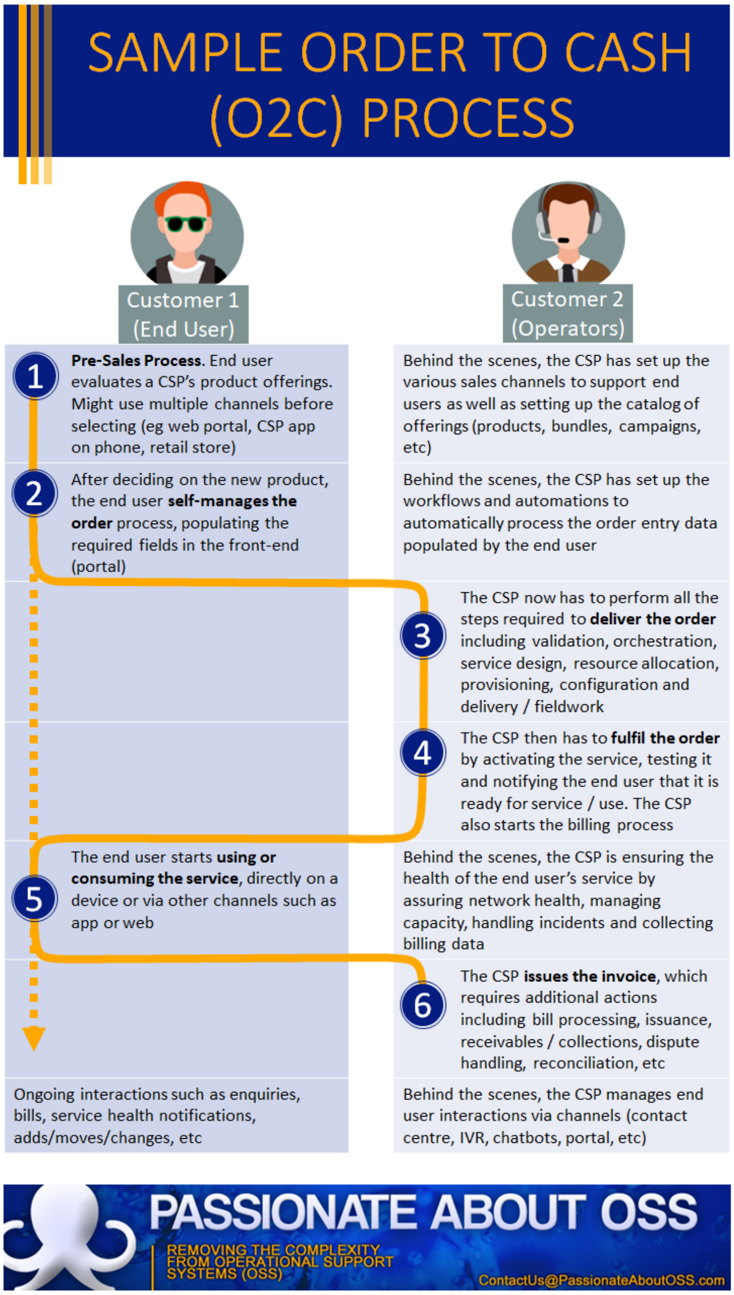
For this example, we identify a cost-benefit of $640,000 per day (which we could then multiply over a longer period of course).

Don't stop at just one MTM. Try to add as many MTMs as possible to justify your business case.

Now, keep that "Future State" column in mind when identifying and evaluating alternatives in the next section. Whilst others might evaluate products based on achieving the highest **functionality density**, you know your business case is more likely to be approved based on **cost-benefit density**.

There’s another consideration when identifying and applying volumes to your metrics. Think in “verticals” as well as “horizontals” in relation to the diagram below. Take the examples below:

- Verticals are volumes that relate to **end-to-end workflows** (ie from step 1 through to the end of the O2C customer journey). For example, a CSP might average 2,000 O2C order completions per day, which flow vertically through this chart.
- Horizontals are volumes that relate to **transactions within a workflow** (ie steps 3, 4 and 6 are done by different groups within the CSP and probably processed with different tools). For example, there might be 7,000 instances of step 3 performed on average each day because this group doesn’t just handle O2C flows, but other processes like changes that aren’t shown on this chart)



SAMPLE ORDER TO CASH (O2C) PROCESS

Customer 1 (End User)	Customer 2 (Operators)
1 **Pre-Sales Process**. End user evaluates a CSP’s product offerings. Might use multiple channels before selecting (eg web portal, CSP app on phone, retail store)	Behind the scenes, the CSP has set up the various sales channels to support end users as well as setting up the catalog of offerings (products, bundles, campaigns, etc)
2 After deciding on the new product, the end user **self-manages the order** process, populating the required fields in the front-end (portal)	Behind the scenes, the CSP has set up the workflows and automations to automatically process the order entry data populated by the end user
	3 The CSP now has to perform all the steps required to **deliver the order** including validation, orchestration, service design, resource allocation, provisioning, configuration and delivery / fieldwork
	4 The CSP then has to **fulfil the order** by activating the service, testing it and notifying the end user that it is ready for service / use. The CSP also starts the billing process
5 The end user starts **using or consuming the service**, directly on a device or via other channels such as app or web	Behind the scenes, the CSP is ensuring the health of the end user’s service by assuring network health, managing capacity, handling incidents and collecting billing data
	6 The CSP **issues the invoice**, which requires additional actions including bill processing, issuance, receivables / collections, dispute handling, reconciliation, etc
Ongoing interactions such as enquiries, bills, service health notifications, adds/moves/changes, etc	Behind the scenes, the CSP manages end user interactions via channels (contact centre, IVR, chatbots, portal, etc)

PASSIONATE ABOUT OSS
REMOVING THE COMPLEXITY FROM OPERATIONAL SUPPORT SYSTEMS (OSS)
ContactUs@PassionateAboutOSS.com

Figure 4 – Simplified Order to Cash (O2C) workflow with hand-offs

4. Step 2: Identify the alternatives

There are two main top-level alternatives here:

- In-house development or
- Vendor or integrator-provided solutions

If going with a vendor-provided solution, the first step is to identify a list of possible suppliers. You may choose to use the supplier list on PassionateAboutOSS.com for this (We'd be delighted to help you apply the Multi-Filter Approach described in Section 3.3 here). Alternatively, you may like to use a more granular process such as an RFI (Request For Information), as described in the vendor selection process and helpful tools on PassionateAboutOSS.com.

5. Steps 3 to 6. Detailed Solution Evaluation

The following steps from the HBR document can be summarised as the Detailed Solution Evaluation:

- Step 3: Gather data and estimate time frame
- Step 4: Analyze the alternatives
- Step 5: Make a choice and assess the risk.
- Step 6: Create a plan for implementing your idea

If you're completing your project using internal resources, then you will control the Detailed Solution Evaluation (DSE). If you're using external suppliers, you will need their support to perform your DSE.

Ideally, you will have already conducted your short-listing of suppliers to limit the number of organisations you need to engage with during the DSE stage.

Most organisations are obliged to seek competitive quotes so you may choose to use an RFP / RFQ (Request for Proposal / Quote) process to collaborate with the shortlisted suppliers on each of the steps of the DSE, including phases / milestones, risk registers, estimated schedules and plans.

Examples of vendor selection processes and tools are provided on PassionateAboutOSS.com to help walk you through steps 3 to 6 in your DSE.

You'll need to develop a more comprehensive list of benefits / requirements / evaluation-criteria for your RFP. You may like to use the various Requirements Capture tools on PassionateAboutOSS.com to help with this.

When preparing your requirements, consider using the "Thrash hard, Thrash early" approach.

Do you know who is impacted by your idea? You may wish to use the Stakeholder Circle tool to identify all impacted stakeholders early in the process to include them in the change that this project will bring about. Organisational change management is one of the biggest stumbling blocks for OSS implementation, but is often overlooked until too late. A perfect technical solution is useless unless the operators are willing and able to use it.

Once you've worked your way through the DSE process described above you may need to provide a summarised / simplified vendor or option analysis in your business case. You may want to re-use a similar evaluation matrix to the one shown in **Error! Reference source not found.**. You may also want to include your more detailed vendor / option analyses as appendices to the Business Case.

When preparing your Business Case, note that the vendor will tend to identify what they need to do, plus a list of assumptions / exclusions. Don't fall for the trap of failing to adequately incorporate what your organisation needs to do as well, including:

- Change Management (refer to the Eight Steps of Change Management on PassionateAboutOSS.com)
- Data Collection (refer to the Seven Steps of Data Migration on PassionateAboutOSS.com)
- Process Re-engineering (and possibly Organisational Restructuring)
- Training
- Testing
- Program Management
- Resource Allocation and Team Building
- Ongoing Operations
- Operational Handover

These aspects can add significant additional effort and cost within your organisation and will almost definitely not be included in the suppliers's proposals.

Ensure your consolidated financials and schedule consider the aspects listed above.

Also ensure that the supplier supports an extensive operational readiness program for all team members who will eventually operate the OSS, especially if this project is introducing a brand new product/s into the OSS.

An OSS can have broad-ranging impacts upon the organisation. The breadth of impact is often not known and many impacted users aren't equipped to handle the change yet. As a result, it is definitely preferred to run a pilot or trial for three main reasons:

- Getting users familiar with the new tools so they can give feedback into the main project via customer requirement refinements
- Testing whether the processes and functionality of the new tools are fit for purpose
- Reducing the change management process via involving users in the transformation and associated approvals

6. Step 7: Communicate your Case

This is where you prepare the business case and persuade the house down by communicating far and wide across the organisation about your OctopOSS. You'll need all the project champions you can get.

Clearly articulate why you have arrived at your conclusion and are promoting the solution that you are. Remember that this is effectively a sales pitch, albeit an internal one. Engage the sponsors and stakeholders in the opportunity and benefits of your project with a compelling story.

With every sales pitch, you need a clear call to action to ask someone to do something specific to assist (eg asking for approval to proceed, access to resources such as headcount, capex, etc)

You may need to develop your story from multiple perspectives as the different stakeholders (sponsors, stakeholders, executives, customers, employees, finance team, etc) will have different driving forces behind giving their tick of approval (or not). Not only that, but they'll have different ways of processing the information you present to them – some fast, some slow; some visual, some audible, some via demos, some via documents.

7. Financial Analysis

Your business case template will undoubtedly include financial analysis. This section provides insights into your internal financial impacts as there will be supplementary costs to the organisation over and above the supplier's quotes.

We've created a blog post called "How Much will your OSS Cost?" on PassionateAboutOSS.com which provides some high-level rules of thumb for estimating the cost of your OctopOSS.

However, you should work closely with your shortlisted vendor and internal cost estimation team to refine these numbers to suit your specific OSS Business Case.

7.1. Costs

7.1.1. Estimation of Costs (CAPEX)

CAPEX (Capital Expenditure) is when an organisation spends money to buy a fixed asset, in this case the supplier's hardware and software costs.

Depending on your organisation's chosen accounting methods, your organisation may choose to capitalise all costs for the project, including the supplier's services costs and perhaps even internal project costs.

As per the call-out box above about OSS estimates, the rules of thumb listed are:

1. OSS Application Licences (x)
2. Third party hardware/software (0.25 – 1x)
3. Project Implementation services (1 – 4x)

Where x is the up-front product license cost.

Note that this estimate will vary between vendor to vendor. Also note that this doesn't include costs relating to turn-on of device interfaces, which can be very costly if management interfaces are not already activated.

The rule of thumb also doesn't include internal project or change management costs if you wish to capitalise these costs.

7.1.2. Estimation of Costs (OPEX)

OPEX (Operating Expenditure) is the ongoing operational costs for running a product or system, in this case the new OSS that you're implementing.

This includes items such as hardware and software license costs, interface license costs, labour for administering and operating the solution, warehousing of spares, travel expenses, leasing expenses, maintenance and repairs, and so forth.

To build a business case, it is recommended that you estimate OPEX before you implement the OctopOSS project and then as per the following section, estimate after implementation of the project.

7.1.3. TCO Model

When comparing pricing options from different vendors, the TCO (Total Cost of Ownership) model is perhaps the more reliable technique. From experience, different vendors will submit different scopes in their offers. One may include third-party software like database licences, whilst others won't. Some will call out exclusions that others include. The TCO model seeks to normalise all the factors, by inserting estimates for any missing factors from each vendor's pricing.

The TCO model also seeks to extend pricing out over a number of years (eg 3-5) to normalise for vendors submitting prices that vary by CAPEX-centric, OPEX-centric or consumption-based pricing models. Contact us at ContactUs@PassionateAboutOSS.com if you'd like assistance preparing your specific TCO comparison model.

7.2. Benefits Analysis

7.2.1. Variation from Current (CAPEX)

Generally, your business case will only present increases in CAPEX unless your project is retiring existing infrastructure. And even then, it is likely that retiring existing infrastructure will reduce OPEX such as removal of maintenance and licenses of the outgoing solution.

7.2.2. Variation from Current (OPEX)

This is generally where you will build the strongest case for your new project when viewed by the project sponsors. If you refer back to the "Lower OPEX" column in Table 1 above, you will identify the various opportunities for OPEX reductions upon which you can build the strength of your business case.

You will need to benchmark your existing situation and estimate the post-project-implementation to provide an analysis.

For example, after implementation of an Alarm Management System at OSS level instead of having one engineer reviewing alarms at each of the five NMS (ie 5 staff per shift), you can reduce this to only 2 staff reviewing all alarms across all platforms. Future implementation of automations, alarm reduction and root-cause analysis engines may allow a further reduction to only 1 staff.

Another example may be the centralised management of spares allows for a 50% reduction in spares on hand, which leads to a capital efficiency improvement. More examples can be identified using the MTM models shown in section 3.4.

7.2.3. Additional Revenues

If you refer back to the "Revenue Increase" column in Table 1 above, you will identify the various opportunities for additional revenues upon which you can build the strength of your business case.

There are some significant opportunities to build additional revenues identified in Table 1. You will need to benchmark your existing situation and estimate the post-project-implementation to provide the detailed financial analysis.

We've created a blog post called “An alternative profit” on PassionateAboutOSS.com that describes how a multi-million dollar OSS had a payback period of only a few months after identifying many unbilled services.

Another example would be in delivering an OSS of such flexibility that your organisation could deliver new services to customers months earlier than competitors, thus taking a significant march on market share. How much might this be worth to your organisation? I'm sure you'll be able to prepare a metric that provides an estimate that makes your project quite compelling.

7.2.4. Brand Value

This is actually one of the most valuable assets of any large organisation. In an increasingly commoditised CSP market, it is one of the few assets that can provide long-term competitive advantage. Brand value is one of the less tangible benefits that can be derived from a new OSS but it can also be one of the most powerful.

If you refer back to the “Brand Value” column in Table 1 above, you will identify the various opportunities for improving brand value and hence opportunities to build the strength of your business case.

There are two aspects of brand value improvement:

- **Brand insurance** – protecting your brand from criticism and customer churn via high levels of service quality and reliability
- **Brand improvement** – increasing the lustre of your brand by consistently bringing attractive products to market earlier than your competitors

There are a number of different ways to calculate brand value and can be determined at a corporate level or at a product/service level. Depending on your solution, it could be either or both.

At a product/service level, brand value can be calculated by comparing the market value of your product with a comparable product that has negligible brand value (ie a commoditised equivalent). By calculating the different price points, you can calculate additional profits earned due to brand image, then calculate present value of the stream of combined additional profits in the future. Before and after estimates will give you an additional brand value derived from the introduction of your OSS project.

At a corporate level there are several alternatives. A method to use here is to measure your organisation's market capitalisation and deduct the value of all assets (except the brand). The remainder can be considered to be brand value. You can attempt to model future market cap after the introduction of your OSS project and indicate increases in brand value.

7.2.5. Intangible Benefits

Despite your best efforts to quantify the benefits in Table 1, there may still be significant benefits that remain intangible but still add weight to your business case. Include these too of course.

7.3. Payback Period

You will have to demonstrate the return for your organisation's time, capital and all associated risks. The payback period of your project is basically the period of time taken to recover the costs of your project (ie the period of time that the accumulated benefits in section 7.2 are required to overtake the costs identified in section 7.1.

The lower the number, the more compelling your business case will be to possible project sponsors, hence the importance of being able to quantify the benefits listed in Table 1.

8. And Finally....

- Keep your OSS Business Case focussed and concise. Don't try to convey too much information to time-poor executives / stakeholders / sponsors. The following is a checklist of items that must stand out clearly above anything else:

1. What is your objective / intent	
2. Why does your organisation need to achieve this objective	
3. Simple financials: a) How much needs to be invested b) What are the earnings (revenues / savings / brand-value) c) How long until profits are derived	
4. What do you need to do to make it happen	
5. What hurdles / risks you expect to face and how you'll overcome them	

  Other information (eg technical solutions, competition, market size, etc) effectively just supports the items above
- Keep all supporting documentation and calculations ready if stakeholders want to delve deeper
- Good luck!!

PASSIONATE ABOUT OSS

SHARING OUR PASSION
FOR OPERATION AND BUSINESS
SUPPORT SYSTEMS

Custom Services

Does your business case justify seeking expert advisors? Do you have a need for specialist implementation skills into a particular area but don't have the time and/or resources for your specific project?

Passionate About OSS offers independent and customized research as well as expert consulting services. We will research, evaluate, and report findings / recommendations based on your unique requirements for various projects such as:

- Vendor Selection
- Market Research
- Contract Advisories
- Product Short-listing
- Competitive Analysis
- Change Management
- Technical and Risk Assessment
- Independent Implementation Advice

For special consultancy or research requests, email us at
ContactUs@PassionateAboutOSS.com