ServiceOPS is our own internal operations for delivering our services. This is based on our Enterprise framework simplified version with key steps, processes and guides to enable better engagement with clients, project managers, implementation, delivery and deployment use cases.

Most of these are either BROWN or GREEN FIELD Projects, except when almost 80% of the assets are being reused from earlier projects.

https://www.mckinsey.com/business-functions/operations/how-we-help-clients/service-operations

https://www.referenceforbusiness.com/management/Sc-Str/Service-Operations.html

Although services can have tangible (high goods content) and intangible (low goods content) attributes, services are generally regarded as intangible that is, you can't see, feel, or test a service's performance before purchasing it. Hence, reputation is extremely important. Since services are intangible, it makes sense that they can't be patented. In fact, labor is usually the most important determinant of service organization effectiveness.

https://www.referenceforbusiness.com/management/Sc-Str/Service-Operations.html#ixzz75giJupav

Service operation includes the following processes: **event management, incident management, request fulfillment, problem management, and access management**. Service operation also includes the following functions: service desk, technical management, IT operations management, and application management

https://en.wikiversity.org/wiki/IT_Service_Management/Service_Operation

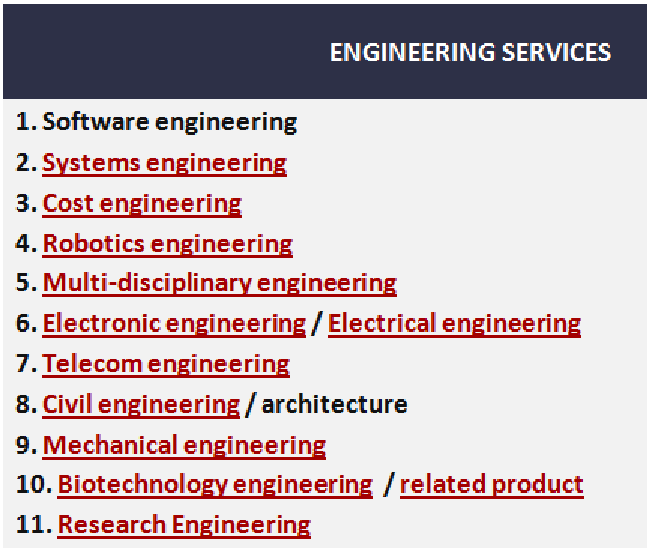
At our startup,

1. ServiceOPS has ORG MODULE (Service).
2. Service has Unique OFFERS (Products and Services)
3. Products and Services has PROCEDURES and PROCESSES (as part of Project charter and Project Plan or client tickets)
4. Each Product and Service Procedure/ Process has STEPS and DETAILS
5. The STEPS generate RESULTS and DATA
6. These RESULTS are tested, verified, re-factored for UAT and final deployment

Based on above we offer our services that enable clarity yet simplicity in the deliverables that we implement. Various tools and forms will be used to note down all the stages

ServiceOPS will look into the Technical part based on project needs. We leverage upon our internal Technical tools, staff and partner best practices to implement the modernization turnkey solutions.

We have about 93 initiatives for Products and there are related projects. A snapshot is shown here



ENGINEERING SERVICES

1. Software engineering
2. Systems engineering
3. Cost engineering
4. Robotics engineering
5. Multi-disciplinary engineering
6. Electronic engineering / Electrical engineering
7. Telecom engineering
8. Civil engineering / architecture
9. Mechanical engineering
10. Biotechnology engineering / related product
11. Research Engineering

To fulfill these 93 initiatives, we have explained the roadmap above; we expect to design more initiatives based on the needs and market demand.

The Operations will monitor, alert, recourse, revamp, throttle, and allocate resources. These are initially start with MANUAL human monitoring and later on automation will be brought in.

As these are mix of BROWN and GREEN field, existing DOMAIN & TECH expertise will be involved that have already worked on such projects and lots of feedbacks from users will allow better features/services to be implemented.

As usual PMI, agile tools, documents and expertise will be used.

TOOLS that will be used will be part of a distributed team and as such most will be Cloud platform DEVOPS based; if not local DEV tools will be used and synched with the main server hosting the production ready sandbox servers.

Our internal tool:

ServicesOps Product features:

ServicesOps: (May be done using Excel, DB (No sql) or Trees)

1. Design Services Tree – at least 3 levels depth and width, flexible to add nodes, dynamically
2. Define requirements tree, see how the hierarchy is done and normalized to fit into the Service Tree with a best value
3. Define Product Tree
4. Define cross-tree references – relation links, dynamically realigned for all 3 trees
5. Provide data value to each node with either a Req, Service and related product
6. Map the Req Tree to Service Tree, see at least one value exist for each Req

7. See how traversing of the Trees is done
8. Once data structures are ready, these trees will be part of an ensemble forest with 3 trees
9. Get sample test requirements, data. Load the trees and give an initial req, see how the search for a unique node is done using some search algorithm (or a DB query)
10. Once search results are returned see if these are real ones that meet the solution
11. Finalize and deploy this as a tool

References: MASTER SERVICES CATALOGUE-CONSULTATION - V1.2.doc

--