Service Value Proposition:

*A value proposition statement is a promise of value to be delivered by your service. Briefly explain how your service solves your client's problems or improves their situation. Briefly describe the benefits to using your service. You may have already defined your service’s value proposition in the service design phase. Copy it here, it will serve as the service development north star.*

Service Features:

*Briefly describe key service features (2-4 bullets)*

Who can and cannot use this service?

*Describe the clients who can, and cannot use this service. For example, some services are available only to healthcare clients, or students, etc.*

Service Roles:

*Identify who fulfills the following roles for your service. For a comprehensive list of roles and responsibilities, please see* [*The Service Roles and Responsibilities*](https://uit.stanford.edu/sites/default/files/2017/12/01/Service%20Roles%20and%20Responsibilities.pdf) *document:*

* Business Owner:
* Service Owner:
* Service Manager:
* Service Designer:
* Operations Owner
* Finance Consultant:

Security:

* If a Data Risk assessment is required, please document the DRA assessment reference number and clearly indicate the data type for which the service has been approved. (Low, Medium, High Risk Data)
* If a Business Associate Agreement (BAA) has been completed, enter the date of the agreement and reference number

Finance Overview

*Elaborate on the following topics (and others as needed). It is easy to forget why key decisions were made, documenting these decisions will save time and effort in the long run.*

* *Description of the cost components, such as primarily labor, capital costs, etc.*
* *Is this service simply cost recovered, or is the intent to bring in more revenue than just cost? Are you planning a strategy of under-recovering for a specified period of time in order to build market share? Is this a pilot rate?*
* *Was a market analysis completed as a sanity check for your proposed service rate?*

Rates and Relevant Bill codes:

*Document the following information for your service. Your assigned financial analyst and billing consultant can provide you with this information.*

**Rates for (Service Name)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Service Name** | **Monthly****Rate** | **Item Code** | **Charge Type** | **Description** |
|  | $100.00 |  | MRC |  |
|  |  |  |  |  |

Revenue Plan Information (Anchorage on Drupal):

*The financial analyst assigned to assist with your costing and rate development can provide the Revenue Plan information.*

|  |  |
| --- | --- |
| **Revenue Plan**  | **Service Name** |
| Revenue Plan ED |  |
| Revenue Plan Funding |  |
| Revenue Plan Srv Category |  |
| Income PTA |  |

 Request and Order Processes:

*Provide details about the request and order processes, for example, is this a billable service? If so, provide details of the OrderIT order form. If a non-billable service, please provide details of the ServiceNow request catalog.*

Operational Responsibility Table

*The following is an example of a support matrix. Please modify to meet your needs*.

|  |  |
| --- | --- |
| **Task** | **Responsible Group** |
| Code development/enhancement | Emerging Tech |
| CoreOS patching/updating | Automated |
| CoreOS mass urgent security updates | Emerging Tech |
| Core Images mass urgent security updates | Automated |
| Outage notification | Emerging Tech |
| Automation Tooling | Emerging Tech |
| Platform architecture upgrade | Emerging Tech |
| Component updates | Emerging Tech |
| **Change Management** |  |
| Open and close scheduled change requests - only required for Jenkins and GitLab (these are not redundant components) | Emerging Tech |
| **Infrastructure Monitoring, Notifications, Troubleshooting** |  |
| Configure monitoring | Automated |
| Monitor apps and system health | ITOC -> ET |
| Respond to problems as they arise | ITOC -> ET |
| **Application-Level Support** |  |
| Routine code testing, updating, deployment | SWS |
| Outage scheduling | SWS |
| Client Application monitoring | SWS |
| Security and functional updates for client applications | SWS |
|  |  |

Service Support and Escalation:

Please modify the following language to correctly describe support and escalation processes for your service.

* The IT Operation center (ITOC) provides first tier support for (Service Name). The ITOC is available 24x7x365. Requests for support can be submitted via the [Stanford Services & Support](https://stanford.service-now.com/services/) portal, or by calling 5-HELP (725-4356).
* Support issues with a service impact of ‘three’ or ‘four’ will be responded to as per the following incident management chart. These incidents will be responded to during normal business hours. Incidents with a service impact of ‘two’ will be escalated to appropriate second tier support teams as required, including outside of normal business hours. Each Tier Two support team will provide the ITOC with contact information and a schedule indicating staff availability for after hours support.
* Support outside of normal business hours during business critical periods is available for premium clients on a fee basis, and by pre-arrangement. A fee would be assessed only if second tier service is required outside of normal business hours. A scenario in which pre-planned after-hours support might be made available is in which University Human Resources (UHR) requests stand-by support during Open Enrollment. Important – this support level MUST be pre-arranged, or there is no guarantee of availability.

Incident Response Matrix:

*The following is an example of an incident response matrix. Use it as a guide for identifying how the support teams should respond to incidents for key service functionality.*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Incident Description** | **Service Impact** | **Response Urgency** | **Response Time Target** | **Tier 1 Support**24x7x365 | **Tier 2 Support**M-F, 9-5 |
|  |  |  |  |  |  |
| User can't log into their website | 4 - Minor /Localized | 4 - Low | 6 hours | ITOC | SWS |
| Users can't log into any website | 3 - Moderate /Limited | 3 - Medium | 2 hours | ITOC | SWS |
| Some or all files (CSS, images, Javascript) disappear on one website | 4 - Minor /Localized | 4 - Low | 6 hours | ITOC | SWS |
| Some or all files (CSS, images, Javascript) disappear on all websites | 2 - Significant/Large | 3 - Medium | 2 hours | ITOC | SWS |
| One site - WSOD or cannot reach database error | 3 - Moderate/Limited | 3 - Medium | 2 hours | ITOC | SWS |
| All sites - WSOD or cannot reach database error | 2 - Significant/Large | 2 - High | 2 hours | ITOC | SWS |

Next Service Review Date:

*For new services, establishing six and twelve month checks is recommended - not of the entire service, but of targeted processes, such as operational support and provisioning, as well as a check of the service costs. Document your next meeting date.*

* *Processes for review at 6 months*
* *Processes targeted for review at 12 months*

Glossary of Terms (Optional):

The following is optional, however, very useful for those who may not be familiar with the foundational technology of your service. Feel free to add new definitions and remove those not relevant to your service.

|  |  |
| --- | --- |
| *Datadog* | <https://www.datadoghq.com/> Cloud monitoring, presented to a dashboard. |
| *Docker* | <https://www.docker.com/> Docker is a tool that segregates system processes and resources at the kernel level. This allows applications to be run in “containers”. Docker containers are portable |
| *FleetUI* | <https://github.com/coreos/fleet> Application performance analysis and management FleetUI is the user interface for Fleet |
| *Git* | <https://git-scm.com> Git is a [free and open source](https://git-scm.com/about/free-and-open-source) distributed version control system |
| *Jenkins* | [*https://jenkins-ci.org*](https://jenkins-ci.org)Jenkins is an [award-winning](https://wiki.jenkins-ci.org/display/JENKINS/Awards), cross-platform, **continuous integration and continuous delivery** application that increases your productivity. Use Jenkins to **build and test your software projects continuously** making it easier for developers to integrate changes to the project, and making it easier for users to obtain a fresh build. It also allows you to **continuously deliver** your software by providing powerful ways to define your build pipelines and integrating with a large number of testing and deployment technologies. |
| *Newrelic* | <http://newrelic.com/> Monitoring and performance improvement application |
| *Pingdom* | <https://www.pingdom.com> Monitor uptime and website performance, as well as alerting and incident management |
| *Splunk* | Monitoring of client application logs and We currently use cloud based version, not the version offered by ISO |
| *Terraform* | <http://www.terraform.io/intro> Terraform is a tool for building, changing, and versioning infrastructure safely and efficiently. Terraform can manage existing and popular service providers as well as custom in-house solutions. We use it to Builds Amazon environment (moving from m3large to m4) change definitions and do the updates via GitLab, and add machines to VPC.  |
| *EC2* | Amazon’s Elastic Compute Cloud |
| *ELB* | Amazon’s Elastic Load Balancer |
|  |  |
|  |  |
|  |  |