# Методология разработки и ITIL для DevOps

# Задание 1

Согласно ITIL, какова связь между Данными, Информацией, Знаниями и Мудростью?

**Ответ:**

DIKW - моделей, которые отображают структурные и функциональные связи между данными, информацией, знаниями и мудростью и демонстрирует пути получения ценности в процессе их обработки.



Данные — набор разрозненных фактов, символов (числа, слова, визуальные данные). Находится на дне иерархии и является материалом для обработки, из которого можно получить что-то ценное. Сами по себе данные не несут никакой пользы.

Информация — объединённые по смыслу данные. На этом уровне базовые кирпичики фактов образуют связи. В отличие от данных, информация несёт в себе пользу, т. к. описывает процессы и явления.

Знания — результат фильтрации информации, которая переработана таким образом, что возникает возможность делать выводы.

Мудрость — верхушка пирамиды. На этом этапе обработки данных к знанию добавляется понимание.

# Задание 2

Приведите примеры различных видов изменений (в понимании ITIL) из вашего рабочего опыта.

**Ответ:**

Есть критический важный ресурс, для условности назовем его PR1, на его примере будем рассматривать виды изменений (ITIL) из моего рабочего опыта.

**Standard type:**

1. Изменение конфигурации vps.

2. Изменения прав пользователям/сотрудникам на различных ресурсах.

Как правило изменений данного типа имеют минимальный риск. Запросы таких изменений за рабочий день бывает много.

**Normal type:**

1. На старте проекта отказоустойчивость было реализована уровне контейнеров, на 2-3 vps.   
В ходе развития проекта и увеличение кол-ва сервисов, осуществили переход на оркестратор контейнеров K8S.

2. Следующим шагом ввели отказоустойчивость LB на базе nginx, добавили в работу инфраструктуры проекта vrrp протокол. Реализовали VIP.

В приведенных выше примерах полностью использовали процесс управления изменения (Change Management)

**Emergency:**

До введения отказоустойчивости, для примера LB, при неполадках сети в ДЦ, где расположен ресурс, проект PR1 становился недоступен. Приходилось на скорую руку поднимать 2й LB в другом ДЦ, менять ip, менять конфигурации, firewall правила. Инфраструктура проекта сильно менялась. Такие изменения требовали принятия быстрой оценки, решения, утверждения и реализации.

**Remediation:**

Изменений предназначенные восстановить ресурс, которые перестал правильно функционировать после предыдущих изменений.

К сожалению проекты на большой дистанции подвержены некоторым изменениям, которые нарушают работу ресурса.   
К примеру неправильное значение переменной – пароль для подключения к базе или адрес.

# Задание 3

Как категоризация инцидентов (Incidents) помогает в управлении инцидентами (Incident Management)?

**Ответ:**

Инцидент – внеплановое событие, которое прерывает или уменьшает качество оказываемой it-услуги.

Incidentmanagement (IM) ***-*** процесс, отвечающий за управление жизненным циклом всех инцидентов.

IM гарантирует, что нормальная работа службы будет восстановлена как можно быстрее, а воздействие на бизнес будет сведено к минимуму.

На одном из этапов жизненного цикла инциденту присваивается категория и степень. Степень определяет уровень ущерба, который может нанести инцидент бизнесу/it-услуги. От степени присваивается срочность инциденту - временные рамки, в которые инцидент должен быть устранен.

В зависимости от того, какую область ИТ или бизнеса затрагивает инцидент, например, сеть, оборудование и т. д., инциденту можно присвоить категорию и соответствующую подкатегорию.

После присвоения инциденту категории и степени приоритета он автоматически переадресуется специалисту, обладающему необходимыми знаниями и навыками для решения данного события.

# Задание 4

Зачем нужен мониторинг? В своём ответе постарайтесь придерживаться идей ITIL.

**Ответ:**

Событие – любое изменение состояния нашей конфигурационной единицы или услуги.

Для того чтобы быть эффективным, система должна знать состояние инфраструктуры и ее компонентов, а также отслеживать любые отклонения от нормальной работы. Управление событиями реализовывается с помощью инструментов мониторинга.

Благодаря инструментам мониторинга и интеграции с разными сервисами, мониторинг позволяет сообщить об изменениях или отклонениях статусов конфигурационных единиц/услуг, что позволяет команде своевременно предпринять необходимые действия. Это увеличивает эффективность и позволяет задействовать персонал на новом уровне.

# Задание 5

Перечислите ключевые метрики качества и эффективности работы Технической поддержки?

**Ответ:**

**KPI -** метрика, которая используется для управления процессом, ИТ-сервисом или активностью

Среднее время первого ответа; среднее время ответа; общее количество заявок; количество просроченных заявок; доля заявок; закрытых в ходе первого обращения; доля решенных заявок; среднее время обработки заявки; лучшие специалисты; индекс лояльности; оценка удовлетворенности клиентов.

# Задание 6

Приведите пример любого процесса (Process) с которым вы часто сталкивайтесь и опишите для него следующие параметры:

* Входные и выходные данные;
* Цель;
* Уровень по CMMI Модели зрелости и обоснование Вашего решения;

**Ответ:**

Одним из часто сталкиваемых процессов у меня на практике – это репликация и резервирование базы данных.

**Цель репликации**: при сбое в работе мастер базы быстрое переключение на реплику.

**Цель резервирование**: возможность восстановить данные при их потере.

**Триггер процесса** – запрос от команды разработчиков

**К водным данным** этого процесса можно отнести параметры авторизации в базе:   
адрес базы (dns/ip, port), пользователя и пароль с администраторскими правами.

**На выходе получаем** инстанцию реплики и архив базы данных.

Процесс репликации и резервирование по уровню CMMI Модели зрелости **– 5 Optimized.**

Имеются строгие правила и подробная документация. Инстанции (мастер и реплика) и резервирование – мониторится. Процесс резервирование автоматизирован. В установленное время запускает автоматически процесс. После его завершения приходит отчет и результаты архивирование (архив данных). Процесс репликации автоматизирован – кластер базы данных. Кластер сам определяет мастера. Во время перебоев в работе мастер инстанции деградирует текущего мастер инстанцию и возводит до уровня мастера следующую инстанцию реплики. Система оповещает о сбоях в работе мастера, а также о переключении/повышении на новую инстанцию базы.

# Задание 7

Опишите основные принципы Agile Manifesto и объясните задачу, которую они помогают решить, будучи примененными к разработке ПО;

# **Люди и взаимодействие** важнее процессов и инструментов.

# **Работающий продукт** важнее исчерпывающей документации.

# **Сотрудничество с заказчиком** важнее согласования условий контракта.

# **Готовность к изменениям** важнее следования плану.

# То есть, не отрицая важности того, что справа, мы все-таки больше ценим то, что слева.

# Задание 8

Опишите основные Достоинства, Недостатки и Критерии применимости следующих моделей разработки:

* Waterfall
* Iterative
* Spiral
* V-Model

**Ответ:**

**Waterfall –**  каскадный метод разработки продукта.

Отсутствие итераций, разработка это один большой цикл. Выявление и исправление ошибок происходит на этапе тестирования. Клиент не участвует в создании продукта после формирования ТЗ. Вся разработка идет строго по ТЗ. Следующий этап не начинается, пока не закончится предыдущий.

Использование waterfall:

Простые проекты малого или среднего размера; проекты, требующие более строгого контроля; проекты, которые должны соответствовать многочисленным правилам и нормам

Основным требованиям является понимание финального состояния продукта. Используя waterfall даст большое и подробное ТЗ, строгое планирование этапов проектирования, бюджет и сроки проектирования.

**V-Model –** линейная модель проектирования продукта, у которой на каждой стадии есть активное тестирование.

Она так же как waterfall сильно завязана на ТЗ и анализе, отсутствует гибкость, процесс разработки – большой цикл, клиент знакомится только с финальной версией продукта, но в отличии от предыдущего метода, у v-model на каждом этапе есть активное тестирование, которое влечет за собой подорожание разработки, но повышает качество и минимизирует риски.

Проекты, в которых недопустимы сбои и простои, где главной ценностью является качество и гарантия работы продукта – авиация, медицина, космонавтика, армия и т.д.

**Spiral -** модели разработки все идет по спирали: жизненный цикл продукта закручен в нее и разделен на фазы. Прохождение каждого витка дает инкремент.

D этой модели появляется вовлеченность клиентов. Они могут быть задействованы на этапах изучения и рассмотрения каждого цикла.

Типичная спиральная итерация длится около 6 месяцев и начинается с 4 важных действий - тщательного планирования, анализа рисков, создания прототипов и оценки ранее выполненной части. Повторяющиеся спиральные циклы серьезно увеличивают сроки проекта.

**Interative –** это проектирование продукта, которое осуществляется небольшими этапами, в ходе которых ведется анализ полученных промежуточных результатов, выдвигаются новые требования и корректируются предыдущие этапы работы

Появляется гибкость разработки, на каждом этапе можно вносить корректировки. Клиент вовлечен в разработку в достаточной мере. Итерационная модель жизненного цикла не требует для начала полной спецификации требований

Такой подход используется для крупных и критически важных проектов, которые предпочтительно состоят из слабо связанных частей, таких как микросервисы.

# Задание 9

Для приведённых ниже проектов, что бы вы выбрали: Scrum или Kanban?

**Подготовка к свадьбе: scrum –** повторяемые спринты  **Работа отдела технической поддержки: kanban** – не прерывный процесс

**Разработка коробочного версии системы учёта пользовательских заявок с дальнейшей поддержкой этого продукта**: команда разработки системы учета – scrum, команда поддержки – Kanban

**Косметический ремонт ванной комнаты**: scrum – повторяемые спринты

**Startup проек** – scrum/kanban

# Задание 10

Какие принципы Lean поддерживает Kanban?

**Ответ:**

Manage Flow - управление поток производства/разработки

Limit Work In Progress – кол-во одновременных задач на исполнение

Visualize value creation chain - визуализация цепочки произведенной работы, получение ценностей

Continuous Improvement – постоянные улучшения

Make Process Explicit - явность рабочего процесса

Pull instead of Push - берем задачи в зависимости от нагрузки, а не получаем их

* Top of Form