# Методология разработки и ITIL для DevOps

# Задание 1

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

**Все они объединены в Knowledge Management (Управление знаниями, Менеджмент знаний). Данные – это то, что собирается в процессе мониторинга, анализа логов или инфраструктуры.**

**Информация – добавление к собранным данным контекста из других источников или инструментов.**

**Знания – это структурированная и дополненная информация, сведённая в документацию или базы знаний**

**Мудрость – использование знаний для принятия правильных решений в будущем**

# Задание 2

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

**Стандартные изменения: плановый апгрейд или замена рабочей станции сотруднику, с сохранением профилей и информации.**

**Обычные изменения: установка обновлений ПО на рабочих станциях и серверах.**

**Критические изменения: срочный переход на удалённый режим работы в условиях пандемии при вступлении в силу различных ограничений от регуляторов.**

# Задание 3

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

**Позволяет распределять силы и рассчитывать время устранения инцидента, тем самым определять время простоя, в зависимости от его уровня.**

# Задание 4

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

**Мониторинг позволяет пополнять придерживаться системы Управления знаниями, обнаруживать и прогнозировать инциденты, что позволит уменьшить время отказов.**

# Задание 5

Перечислите ключевые метрики качества и эффективности работы Технической поддержки?

**Для каждого типа события (Инцидент, Проблема, Оповещение, Событие) собираются количество этих событий, повторяемость. Подсчитывается количество по критичности. Соотношение количества событий между собой. Количество и процент перехода событий из одной категории в другую. Оценивается время на устранение и подсчитывается количество превышений обговорённого времени устранения. Стоимость устранения события. Количество и процент ошибочно оценённых по срокам, стоимости и критичности событий. Метрики бывают технологические – чаще всего данные мониторинга, метрики процессов – оценка успешности процесса (CSF), который задаётся метрикой KPI – целевой показатель эффективности и метрики сервисов.**

# 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;

**Восстановление работоспособности рабочей станции. Входные данные: рабочая станция функционирует не ожидаемым образом или не функционирует вообще. Цель устранить неполадки. Подсчитать время и стоимость устранения. Это процесс второго уровня: предсказать его наступление невозможно, но есть документация и рекомендации по диагностике и ремонту различных типов неисправностей. Точное время и стоимость достижения цели указать порой невозможно, потому что при одинаковых симптомах могут быть разные причины.**

# Задание 7

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

**Основные принципы:**

* **Наивысшим приоритетом для нас является удовлетворение потребностей заказчика, благодаря регулярной и ранней поставке ценного программного обеспечения.**
* **Изменение требований приветствуется, даже на поздних стадиях разработки. Agile-процессы позволяют использовать изменения для обеспечения заказчику конкурентного преимущества.**
* **Работающий продукт следует выпускать как можно чаще, с периодичностью от пары недель до пары месяцев.**
* **На протяжении всего проекта разработчики и представители бизнеса должны ежедневно работать вместе.**
* **Над проектом должны работать мотивированные профессионалы. Чтобы работа была сделана, создайте условия, обеспечьте поддержку и полностью доверьтесь им.**
* **Непосредственное общение является наиболее практичным и эффективным способом обмена информацией как с самой командой, так и внутри команды.**
* **Работающий продукт — основной показатель прогресса.**
* **Инвесторы, разработчики и пользователи должны иметь возможность поддерживать постоянный ритм бесконечно. Agile помогает наладить такой устойчивый процесс разработки.**
* **Постоянное внимание к техническому совершенству и качеству проектирования повышает гибкость проекта.**
* **Простота — искусство минимизации лишней работы — крайне необходима.**
* **Наилучшие требования, архитектурные и технические решения рождаются у самоорганизующихся команд.**
* **Команда должна систематически анализировать возможные способы улучшения эффективности и соответственно корректировать стиль своей работы**

**Данные принципы помогают делать заказчика счастливым благодаря регулярным и ранним релизам, команду счастливой благодаря созданию благоприятных условий для работы, а так же совершенствоваться благодаря постоянному анализу и улучшению процесса.**

# Задание 8

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

* Waterfall

**Плюсы: Предсказуемый, линейный, полностью документированный процесс, с чёткими переходами между этапами. Минусы: невозможно изменить требование к проекту, добавить новые или удалить уже неактуальные фичи. Невозможно посмотреть результат в процессе, до полного окончания работ. Тестирование проводится в самом конце разработки, поэтому устранение недостатков будет обходиться очень дорого. Применим для коробочных продуктов с чёткими сроками и требованиями, без применения каких-либо прорывных и экспериментальных технологий**

* Iterative

**Плюсы: возможность добавлять и изменять компоненты по ходу разработки. Нет необходимости в самом начале составлять точные требования к результату. Минусы: требования могут меняться по ходу процесса. Изначально не описан конечный результат, он формируется в процессе работы, при этом фундамент нужно заложить уже в самом начале работы. Применим для продуктов состоящих из множества частей, которые могут появляться или изменяться в процессе жизненного цикла продукта.**

* Spiral

**Плюсы: Постоянная оценка рисков, возможность экспериментировать, необязательно иметь точное представление о результате. Минусы: для оценки рисков требуются высококвалифицированные специалисты, постоянное увеличения времени циклов.**

* V-Model

**Плюсы: постоянный контроль качества на всех этапах, итоговый результат известен с самого начала разработки. Минусы: очень дорогая и время затратная модель разработки, невозможно вносить изменения по ходу разработки.**

# Задание 9

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

* Подготовка к свадьбе;

**Kanban**

* Работа отдела технической поддержки;

**Kanban**

* Разработка коробочного версии системы учёта пользовательских заявок с дальнейшей поддержкой этого продукта;

**Kanban**

* Косметический ремонт ванной комнаты;

**Scrum**

* Startup проект.

**Scrum**

# Задание 10

Какие принципы Lean поддерживает Kanban?

* **Flow**
* **Pull**
* **Perfection**