Методология разработки и ITIL для DevOps

Задание 1

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

Пока готовил ответ на этот вопрос нашел потрясающее и определение. «DIKW — информационная иерархия, где каждый уровень добавляет определённые свойства к предыдущему уровню.» Таким образом получается, что данная концепция как последующее понятие взаимодействует с предыдущим:

1. Данные - основа, могут быть самостоятельными;
2. Информация - добавляет деталей к Данным, делая их более конкретными;
3. Знания - добавляют информацию о том, как с этой информацией можно взаимодействовать, как мы получили эту информацию, и как она связана с другой информацией
4. Мудрость - это знание, применяемое в действии. Это суммарная полезность всей иерархии. То есть это то, к чему в конечном итоге приходит субъект после длительной работы с информацией и получения достаточного количества знаний.

Задание 2

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

Standart - Создание новой учетной записи под нового инженера. (Маленькая организация, в которой нет дополнительных разрешений и заявок. Пришел сотрудник - дали доступ)

Emergency - Обрабатывающий сервер перестал отвечать, срочно требуется восстановить к нем доступ и поднять все работающие системы

Normal - Установка дополнительного софта для оптимизации работы (для разработчиков ставил Jenkins, чтобы было проще делать обновления, требовалось разрешение технического директора на установку дополнительных программ)

Remediation - Восстановление к предыдущей версии системы, после обновления, которое по ошибке было несовместимо с другими программами.

Задание 3

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

Это позволяет определить тип инцидента (с чем связан), его переадресация соответствующей технической группе, что позволяет сэкономить время и усилия. А так же позволяет распределить инциденты по приоритетам, и сделать акцент на более актуальные проблемы в текущий момент.

Задание 4

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

Мониторинг необходим для максимизации времени доступности системы. Благодаря мониторингу сможем раньше обнаружить проблему, раньше заняться решением, если проблема критичная, найти узкие места в нашей системе. Принять какие-то действия для недопущения или ускорения решения аналогичных проблем в будущем. Понять более полную картину работы нашей системы, основываясь на метрических данных.

Задание 5

Перечислите ключевые метрики качества и эффективности работы Технической поддержки?

Предполагаю, что для оценки качества работы поддержки стоит использовать:

* Процент обращений, закрытых с превышением сроков, установленных в SLA;
* Оценка качества выполнения обращения;

Предполагаю, что для оценки эффективности работы поддержки стоит использовать:

* MTRS (время восстановления системы);
* MTBF (Время между сбоями);
* MTBF (время между сбоями в системе),

Задание 6

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

Процесс: «открытие входной двери в квартиру»

* Выходными данными являются:
  + Связка ключей;
  + Закрытая дверь;
* Выходными данными являются:
  + Связка ключей;
  + Открытая дверь;
* Целью является вход в квартиру через открытую ключом дверь.
* Уровень по CMMI Модели зрелости и обоснование Вашего решения;

Думаю, что уровень зрелости равен 4 (количественно-управляемый уровень), поскольку риск невысокий, но есть (мы знаем как выглядит наш ключ, но при наличии двух одинаковых ключей отличить их будет сложнее, но на основе опыта делается достаточно просто). Именно на основе большого количества практики мы открываем дверь нужным ключом без всяких проблем.

Не третий уровень, потому что мы знаем цель, мы знаем как ее достичь и что для этого нужно.

Не пятый уровень, потому что не стоит задача постоянного улучшения, потратим на это 3 или 4 секунды - не так важно, как факт открытия двери.

Задание 9

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

* Подготовка к свадьбе;
  + Kanban, потому что подготовка начинается сильно заранее, что позволяет потратить много времени и получить один итоговый результат.
* Работа отдела технической поддержки;
  + Scrum, потому что будет большой потом данных, много задач будет приходить, после решения получать сразу новые, будет как постоянный конвейер
* Разработка коробочного версии системы учёта пользовательских заявок с дальнейшей поддержкой этого продукта;
  + Scrum, потому что после каждого спринта уже будет какой-то результат, от которого можно будет делать небольшие отклонения, то есть вносить коррективы в разработку постепенно, а не делать систему заново с нуля.
* Косметический ремонт ванной комнаты;
  + Здесь решить гораздо сложнее, поскольку к какой-то одной методологии сложно привязаться, поскольку это индивидуально. Можно выбрать Kanban и получить результат, который может отличаться от желаемого, а можно выбрать Scrum и видеть, как происходит модернизация и вносить правки.
* Startup проект.
  + Думаю, что здесь так же, как и в предыдущем пункте сложно выбрать. Но считаю, что более правильным вариантом будет Scrum, поскольку он гибче, а Startup - быстро развивающаяся организация, которой нужно вносить изменения еще до получения конечного продукта.