***23. Конфигурационное управление***

Конфигурационное управление (Configuration Management) является процессом управления и контролем конфигурацией продукта или системы на протяжении всего жизненного цикла. Оно включает в себя идентификацию конфигурационных элементов, управление изменениями, версионирование, учет и аудит конфигураций. Вот некоторые ключевые аспекты конфигурационного управления:

1. Идентификация конфигурационных элементов: Определите конфигурационные элементы (КЭ), которые составляют систему или продукт.
2. Управление изменениями: Разработайте процесс управления изменениями, который обеспечит контролируемые и документированные изменения в конфигурационных элементах.
3. Версионирование и учет: Ведите учет версий конфигурационных элементов и документации. Каждая версия должна быть уникально идентифицирована и должна содержать информацию о внесенных изменениях, дате, авторе и других релевантных атрибутах.
4. Контроль и аудит: Проводите контроль и аудит конфигураций, чтобы убедиться в их соответствии установленным требованиям и стандартам.

Внедрение конфигурационного управления способствует лучшей организации и контролю изменений в системе или продукте, уменьшению рисков и конфликтов, повышениюэффективности работы и обеспечению качества и надежности.

***24. Управление изменениями.***

Управление изменениями (Change Management) — это процесс планирования, внедрения и контроля изменений в организации с целью эффективного управления переменами и минимизации негативных воздействий на бизнес-процессы и сотрудников. Вот некоторые ключевые аспекты управления изменениями:

1. Идентификация изменений: Определите изменения, которые могут возникнуть в организации.
2. Оценка воздействия: Оцените потенциальное воздействие каждого изменения на организацию.
3. Планирование изменений: Разработайте детальный план внедрения изменений. Определите этапы, ответственных лиц, ресурсы и сроки для каждого изменения.
4. Коммуникация и обучение: Обеспечьте эффективную коммуникацию о предстоящих изменениях.
5. Реализация изменений: Выполняйте план изменений, следуя определенным этапам и используя необходимые ресурсы.
6. Оценка и корректировка: Оцените внесенные изменения и внесите в них правки, если это требуется.

***25. Менеджмент контроля программных продуктов.***

Менеджмент контроля качества программных продуктов (Software Quality Control Management) является важной составляющей процесса разработки программного обеспечения. Он включает в себя набор практик, процедур и инструментов, направленных на обеспечение высокого уровня качества программного продукта. Вот некоторые ключевые аспекты менеджмента контроля качества программных продуктов:

1. Определение критериев качества: Определите критерии, по которым будет оцениваться качество программного продукта.
2. Планирование контроля качества: Разработайте план контроля качества, который определит, какие процессы и методы будут использоваться для проверки соответствия программного продукта установленным критериям качества.
3. Тестирование программного продукта: Выполнение функциональных тестов, интеграционное тестирование, нагрузочное тестирование, тестирование безопасности и другие виды тестов, которые помогут выявить ошибки и дефекты в программном продукте.
4. Управление дефектами: Установите процесс управления дефектами, который позволит отслеживать и исправлять обнаруженные ошибки и дефекты.
5. Анализ и отчетность: Проводите анализ результатов контроля качества и подготавливайте отчеты о выполненных тестах, обнаруженных дефектах, процессе исправления и общем качестве программного продукта.

***26. Формирование отчётности по информационно-технологическому проекту***

Формирование отчетности по информационно-технологическому проекту является частью процесса управления проектом. Отчетность позволяет оценить прогресс, результаты и эффективность проекта, а также предоставляет информацию заинтересованным сторонам.

шаги для формирования отчетности по информационно-технологическому проекту:

1. Определение цели отчетности: Определите, какую информацию вы хотите предоставить в отчетности и кому она будет предназначена.
2. Составление плана отчетности: Создайте план, который указывает, какие отчеты будут представлены, их формат и частоту.
3. Определение содержание отчетов: Разработайте шаблоны отчетов, которые включают информацию о текущем состоянии проекта, достигнутых результатах, бюджете, ресурсах, рисках и проблемах.
4. Сбор данных: Систематически собирайте данные, необходимые для отчетности.
5. Анализ и интерпретация данные: Проводите анализ данных, чтобы оценить прогресс проекта, выявить тренды и проблемы, идентифицировать успешные аспекты и области для улучшения. Объясните причины отклонений от плана и предложите соответствующие рекомендации.
6. Подготовление отчетов: Отформатируйте отчеты в соответствии с выбранными шаблонами и стандартами.

***27. Создание демонстрационного материала по информационно-технологическому проекту.***

Создание демонстрационного материала по ИТ-проекту необходимо для презентации проекта и привлечения внимания заинтересованных лиц.

Шаги для создания демонстрационного материала:

1. Определение цели и аудитории.
2. Создание краткого описания проекта.
3. Визуализация проекта.
4. Проведение демонстрации.
5. Подготовление обучающих материалов.
6. Предоставление контактных данных.
7. Тестирование и сбор обратной связи.

***28. Наиболее сложные особенности в продвижении информационно-технологического проекта.***

Продвижение IT-проектов всегда вызывало трудности, которые обоснованы высокой конкуренцией и высокой стоимостью привлечения ядра целевой аудитории.

***29. Особенности продвижения IT-продукта.***

Основная особенность продвижения IT-продуктов заключается в том, что зависимость подобных услуг/компаний и прочего находится полностью в руках IT-технологий. Что подразумевает под собой необходимость в знании и умении рекламировать или же “пиарить” свой продукт. В отличии от обычных продуктов - IT-продукт будет крайне не выгодно рекламировать с помощью билбордов и рекламных вывесок. Потому любой IT-компании придется либо самой разобраться, что такое “PR-стратегия”, либо найти для её создания специального человека - PR-менеджера, который будет составлять PR-стратегию на основе имеющихся материалов о IT-продукте и мнениях пользователей интернета на специфических форумах, журналах и прочем том, что так или иначе касается специфики продукта.

***30. Olap системы.***

OLAP (Online Analytical Processing) - это система аналитической обработки данных. Она предназначена для подготовки отчетов, построения прогностических сценариев и выполнения статистических расчетов на базе больших информационных массивов, имеющих сложную структуру.

OLAP системы включают ключевые компоненты:

* Базу данных (БД) - источник, из которого берется информационный материал для обработки;
* OLAP сервер - ядро системы, с помощью которого проводится обработка многомерных структур данных, и обеспечивается связь между БД и пользователями систем;
* Приложения для работы пользователей, в которых формируются запросы и визуализируются полученные ответы.

Хранение и обработка данных с применением OLAP систем могут осуществляться:

* Непосредственно на рабочих местах пользователей;
* В форме реляционных баз данных - при совместной работе OLAP систем с ERP или CRM-системами;
* В форме многомерных баз данных на обособленных серверах.

OLAP системы применяются главным образом в сфере принятия стратегических управленческих решений и используются для бюджетирования, построения прогностических моделей, подготовки финансовой отчетности, хранения результатов.

Техники обработки и варианты хранения информации в БД классифицируют системы OLAP на несколько видов:

* системы ROLAP, работающие с реляционными БД, в которых данные сгруппированы в табличной форме. В системах такого типа возможна аналитика информации в виде чисел и текстов;
* MOLAP системы - многомерные системы, в которых данные в ходе обработки структурируются в OLAP кубы на специализированных серверах.
* HOLAP системы - «смешанные» системы, в которых объединены алгоритмы многомерной структуризации данных в форме кубов и размещения их в реляционных таблицах.

***31. Эволюция ERP систем.***

История ERP началась с систем планирования материальных потребностей (MRP) в 1960-х годах, когда J. I. Case, производитель тракторной и строительной техники, работал с IBM над разработкой того, что считается первой системой MRP. Это и было началом развития MRP и ERP систем, подхваченное крупными производителями и приведшее в последствии к стремительному развитию этой отрасли.

Хотя изначально подобные системы были дорогостоящими в создании, требовали команды экспертов для обслуживания и занимали много места, ранние системы MRP позволяли компаниям отслеживать запасы и процессы на производстве. Это помогло производителям успешно управлять закупкой сырья, своевременной доставкой продукции, что неминуемо приводило к совершенствованию процесса планирования производственных циклов.

В 1970-х несколько крупных поставщиков программного обеспечения, включая Oracle и JD Edwards, решили сделать подобное программное обеспечение доступным для широкого круга компаний.

80-е годы ознаменовали собой веху в истории развития ERP-систем, именно тогда появились первые системы планирования производственных ресурсов (MRP II). Эти более сложные решения поддерживали производственные процессы, выходящие за рамки обычной инвентаризации и отслеживания закупок сырья. Системы MRP II позволяли различным подразделениям, вовлеченным в производство, координировать свои действия, и у них появились более продвинутые возможности планирования производства.

К 1990 году исследовательская фирма Gartner ввела термин “планирование ресурсов предприятия". Новое название системы только подтверждало тот факт, что многие предприятия — не только производственные — теперь используют подобную технологию для повышения эффективности всей своей деятельности.

Именно тогда ERP - системы обрели свою нынешнюю идентичность: единую базу данных для сбора информации во всей компании. ERP-системы поддерживали и другие бизнес-функции, такие как бухгалтерия, продажи, инженерия и управление человеческими ресурсами, чтобы служить единым источником точных данных для всех сотрудников.

ERP-системы продолжали развиваться на протяжении 90-х годов. Одним из крупных прорывов стало появление облачной ERP-системы, впервые предложенной NetSuite в 1998 году. С помощью облачной ERP, рассматриваемой в глобальном смысле как усовершенствование локальных систем. Предприятия получили возможность получать доступ к критически важным бизнес-данным через Интернет с любого устройства, имеющего доступ к глобальной сети. Облачные решения означали, что компаниям больше не нужно было покупать и обслуживать аппаратное обеспечение, что уменьшало потребность в количестве ИТ-сотрудников и облегчало внедрение ERP системы.

Облачная модель сделала ERP-системы, когда-то ограниченные в пределах конкретного предприятия, доступными для небольших компаний, которым не хватало капитала для запуска и поддержки ресурсоемкого локального решения. Малые и средние предприятия разных отраслей смогли пользоваться теми же преимуществами, что и их более крупные коллеги, включая автоматизацию процессов, повышение эффективности и актуальности данных.

Сегодня ведущие ERP-системы представляют собой обширные хранилища информации, способные генерировать отчеты, которые могут осветить эффективность каждого аспекта бизнеса, от продаж и маркетинга до разработки продукта и управления персоналом и операциями. Существует бесчисленное множество приложений, предназначенных для различных отраслей промышленности, бизнес-моделей и задач, и ERP работает как командный центр для того, что может являться обширной сетью программного обеспечения.

***32. АСОИУ для управления IT-проектами.***

АСОИУ (Автоматизированная система управления информационными управленческими проектами) представляет собой комплекс программных средств и методов, предназначенных для автоматизации процессов управления IT-проектами. Она помогает организовать работу команды, планировать и контролировать выполнение задач, управлять ресурсами и обеспечивать эффективное взаимодействие между участниками проекта.

В состав АСОИУ для управления IT-проектами могут входить следующие модули и функциональные возможности:

1. Управление проектами: инструменты для создания и отслеживания графиков работ, управления ресурсами и бюджетом проекта, учета рисков и принятия решений.
2. Управление задачами: система разделения проекта на отдельные задачи, назначение ответственных исполнителей, контроль выполнения задач и отслеживание прогресса.
3. Управление коммуникациями: инструменты для обмена сообщениями, документами и информацией между участниками проекта, ведение обсуждений, система уведомлений и оповещений.
4. Управление документацией: возможность создания и хранения проектной документации, контроль версий, доступ к документам для соответствующих участников проекта.
5. Управление ресурсами: планирование и распределение ресурсов проекта (людских, материальных, финансовых и т. д.), мониторинг затрат и оценка эффективности их использования.
6. Отчетность и аналитика: генерация отчетов о ходе выполнения проекта, анализ данных, оценка результативности, идентификация проблем и поиск улучшений.
7. Интеграция с другими системами: возможность взаимодействия с другими информационными системами, такими как системы учета, системы контроля версий, CRM-системы и др.

Выбор конкретной АСОИУ зависит от требований и потребностей вашей организации или команды. Некоторые из популярных АСОИУ, которые можно использовать для управления IT-проектами, включают в себя Jira, Microsoft Project, Asana, Trello, Basecamp и многие другие.

***33. Технологии разработки IT-проектов в организации.***

Технологии разработки IT-проектов в организации могут варьироваться в зависимости от конкретных требований проекта, доступных ресурсов и предпочтений команды разработчиков. Однако, ниже приведены некоторые популярные технологии и подходы, широко применяемые при разработке IT-проектов:

1. Agile-методологии разработки: Agile является гибким подходом к разработке, который позволяет командам быстро реагировать на изменения в требованиях проекта. Некоторые из популярных Agile-методологий включают Scrum, Kanban и Extreme Programming (XP). Они предлагают итеративный подход, разбивая проект на короткие циклы разработки (спринты) и акцентируя внимание на частой коммуникации и быстром достижении рабочих результатов.
2. DevOps: DevOps объединяет разработку и операционное обслуживание проекта для достижения ускоренного развертывания и повышения стабильности системы. Этот подход включает автоматизацию процессов развертывания, непрерывную интеграцию и развертывание (CI/CD), а также активное взаимодействие между разработчиками и системными администраторами.
3. Контейнеризация: Контейнеризация позволяет упаковывать приложения и их зависимости в изолированные контейнеры, что обеспечивает портативность и согласованность работы приложений в различных средах. Наиболее популярной платформой для контейнеризации является Docker.
4. Облачные технологии: Использование облачных платформ, таких как Amazon Web Services (AWS), Microsoft Azure или Google Cloud Platform, позволяет разработчикам создавать, развертывать и масштабировать приложения без необходимости владения и поддержки собственной инфраструктуры. Облачные технологии обеспечивают гибкость, масштабируемость и доступность для разработки и развертывания IT-проектов.
5. Микросервисная архитектура: Микросервисы являются архитектурным подходом, в котором приложение разбивается на небольшие, независимые и самодостаточные сервисы, взаимодействующие между собой через API. Это позволяет более гибко масштабировать и модифицировать отдельные компоненты системы.
6. Использование открытых и популярных языков программирования и фреймворков: Выбор языка программирования и фреймворков зависит от требований проекта и предпочтений команды разработчиков. Некоторые из популярных языков программирования включают JavaScript, Python, Java, C# и Ruby. Они имеют широкую поддержку сообщества разработчиков и множество готовых фреймворков и библиотек.

Важно отметить, что выбор технологий и подходов к разработке IT-проектов может быть индивидуален для каждой организации и проекта. Он должен учитывать требования бизнеса, существующую инфраструктуру, опыт и навыки команды разработчиков.

***34. Эволюция систем ERP.***

История ERP началась с систем планирования материальных потребностей (MRP) в 1960-х годах, когда J. I. Case, производитель тракторной и строительной техники, работал с IBM над разработкой того, что считается первой системой MRP. Это и было началом развития MRP и ERP систем, подхваченное крупными производителями и приведшее в последствии к стремительному развитию этой отрасли.

Хотя изначально подобные системы были дорогостоящими в создании, требовали команды экспертов для обслуживания и занимали много места, ранние системы MRP позволяли компаниям отслеживать запасы и процессы на производстве. Это помогло производителям успешно управлять закупкой сырья, своевременной доставкой продукции, что неминуемо приводило к совершенствованию процесса планирования производственных циклов.

В 1970-х несколько крупных поставщиков программного обеспечения, включая Oracle и JD Edwards, решили сделать подобное программное обеспечение доступным для широкого круга компаний.

80-е годы ознаменовали собой веху в истории развития ERP-систем, именно тогда появились первые системы планирования производственных ресурсов (MRP II). Эти более сложные решения поддерживали производственные процессы, выходящие за рамки обычной инвентаризации и отслеживания закупок сырья. Системы MRP II позволяли различным подразделениям, вовлеченным в производство, координировать свои действия, и у них появились более продвинутые возможности планирования производства.

К 1990 году исследовательская фирма Gartner ввела термин “планирование ресурсов предприятия". Новое название системы только подтверждало тот факт, что многие предприятия — не только производственные — теперь используют подобную технологию для повышения эффективности всей своей деятельности.

Именно тогда ERP - системы обрели свою нынешнюю идентичность: единую базу данных для сбора информации во всей компании. ERP-системы поддерживали и другие бизнес-функции, такие как бухгалтерия, продажи, инженерия и управление человеческими ресурсами, чтобы служить единым источником точных данных для всех сотрудников.

ERP-системы продолжали развиваться на протяжении 90-х годов. Одним из крупных прорывов стало появление облачной ERP-системы, впервые предложенной NetSuite в 1998 году. С помощью облачной ERP, рассматриваемой в глобальном смысле как усовершенствование локальных систем. Предприятия получили возможность получать доступ к критически важным бизнес-данным через Интернет с любого устройства, имеющего доступ к глобальной сети. Облачные решения означали, что компаниям больше не нужно было покупать и обслуживать аппаратное обеспечение, что уменьшало потребность в количестве ИТ-сотрудников и облегчало внедрение ERP системы.

Облачная модель сделала ERP-системы, когда-то ограниченные в пределах конкретного предприятия, доступными для небольших компаний, которым не хватало капитала для запуска и поддержки ресурсоемкого локального решения. Малые и средние предприятия разных отраслей смогли пользоваться теми же преимуществами, что и их более крупные коллеги, включая автоматизацию процессов, повышение эффективности и актуальности данных.

Сегодня ведущие ERP-системы представляют собой обширные хранилища информации, способные генерировать отчеты, которые могут осветить эффективность каждого аспекта бизнеса, от продаж и маркетинга до разработки продукта и управления персоналом и операциями. Существует бесчисленное множество приложений, предназначенных для различных отраслей промышленности, бизнес-моделей и задач, и ERP работает как командный центр для того, что может являться обширной сетью программного обеспечения.

***35. Особенности менеджмента в малом бизнесе.***

Менеджмент в малом бизнесе имеет свои особенности, связанные с ограниченными ресурсами, небольшой командой и гибкостью. Вот некоторые особенности, которые можно выделить:

1. Мультифункциональность: В малом бизнесе менеджеры часто вынуждены выполнять множество различных задач и ролей. Например, предприниматель может быть и руководителем, и финансистом, и маркетологом. Успешные менеджеры в малом бизнесе должны быть готовыми к такой мультифункциональности и уметь эффективно управлять разными аспектами бизнеса.
2. Гибкость и быстрая реакция на изменения: В отличие от крупных организаций, малые бизнесы могут быстро принимать решения и вносить изменения в свою деятельность. Это позволяет им лучше адаптироваться к изменяющимся условиям рынка, изменениям требований клиентов и конкурентным ситуациям. Гибкость и быстрая реакция на изменения являются ключевыми преимуществами малого бизнеса.
3. Простая и прямая коммуникация: В малых бизнесах команды обычно небольшие, и коммуникация между участниками может быть более прямой и эффективной. Это способствует лучшему пониманию целей, более быстрой передаче информации и принятию оперативных решений. Простая и открытая коммуникация является важным аспектом менеджмента в малом бизнесе.
4. Необходимость в эффективном использовании ресурсов: Малым бизнесам обычно не хватает ресурсов, таких как финансы, персонал и время. Это требует от менеджеров эффективного планирования и управления ресурсами, чтобы максимизировать их использование. Они должны быть в состоянии оптимизировать бюджет, распределять задачи с учетом доступных навыков и способностей, а также управлять временем эффективно.
5. Близкое взаимодействие с клиентами: Малые бизнесы, часто, имеют возможность установить более тесные отношения с клиентами и лучше понять их потребности и предпочтения. Менеджеры в малом бизнесе должны активно взаимодействовать с клиентами, обратную связь, чтобы удовлетворить их требования и построить долгосрочные отношения.

Эти особенности менеджмента в малом бизнесе требуют гибкости, предпринимательского мышления и способности быстро принимать решения. Важно также уметь эффективно управлять ресурсами, коммуницировать и строить отношения с клиентами, чтобы обеспечить успех и рост малого бизнеса.

***36. Управление проектами по методике ‘agile’***

Управление проектами по методике Agile (гибкое управление проектами) является популярным подходом к разработке программного обеспечения и других проектов, особенно в быстроразвивающихся и динамичных средах. Agile признает необходимость гибкого и итеративного подхода к управлению проектами, акцентируя внимание на постоянном взаимодействии с заказчиком и быстрой адаптации к изменениям. Вот основные принципы и практики Agile-управления проектами:

1. Команда разработчиков: Agile основан на формировании самоорганизующихся и мультидисциплинарных команд. Команда включает разработчиков, тестировщиков, дизайнеров и других специалистов, работающих в тесном взаимодействии друг с другом.
2. Итеративность и инкрементальность: Разработка проекта ведется через серии коротких циклов, называемых спринтами. Каждый спринт обычно длится от одной до четырех недель и завершается выпуском инкремента рабочего продукта. Это позволяет быстро проверять и тестировать функциональность и получать обратную связь от заказчика.
3. Постоянное взаимодействие с заказчиком: Заказчик активно вовлечен в процесс разработки и имеет возможность предоставлять обратную связь на каждом этапе. Это помогает уточнять требования и приоритеты, а также гарантирует, что конечный продукт соответствует ожиданиям заказчика.
4. Приоритизация и гибкое планирование: В Agile проекты планируются поэтапно и приоритеты устанавливаются на основе ценности, которую каждый функционал или задача приносит бизнесу. Планы могут меняться в ходе проекта, чтобы учесть новые требования и возможности.
5. Короткие сроки и быстрые результаты: Agile ставит акцент на быстрой доставке ценности. Каждый спринт должен завершаться выпуском работающего функционала, который можно представить заказчику или протестировать пользователями.
6. Контроль и адаптация: Agile предусматривает систематический контроль проекта, а также постоянное обучение и адаптацию на основе полученных результатов и обратной связи. Это позволяет реагировать на изменения, улучшать процессы и повышать эффективность.
7. Непрерывная интеграция и развертывание: Agile поощряет автоматизацию процессов разработки, непрерывную интеграцию (CI) и непрерывное развертывание (CD), чтобы сократить время и риски внедрения изменений в проект.

Применение Agile-методологий, таких как Scrum, Kanban или Extreme Programming (XP), позволяет более гибко управлять проектами, повышать прозрачность, эффективность и качество разработки, а также быстро адаптироваться к изменяющимся требованиям и рыночным условиям.