****

**ОТЧЕТ**

**по производственной практике**

**студента группы ИС-20-3к**

**специальности 6В06107- «Информационные системы»**

**Стратинский Аркадий**

**Ф.И.О. студента**

**Место прохождения практики:**

**ИП «Электромен»**

**название организации**

**Руководители практики:**

**к.т.н. ассоц.профессор факультета “ИТ”,**

**Алимжанова Л.М.**

**Алматы 2020 г.**

|  |  |
| --- | --- |
| **Дата** | **Количество работы** |
|  | Знакомства с компанией и коллективам. Ознакомление с техникой безопасности. |
|  | Определение рабочего места и дальнейшего фронта работы. |
|  | Установка необходимых программных обеспечений |
|  | Начало написания программы в Delphi |
|  | Завершение написания программы в Delphi |
|  | Отчет о проделоной работе и его сдача руководству. |

**Содержание отчета по преддипломной практике**

**по специальности «Информатика»**

|  |  |  |
| --- | --- | --- |
|  | Введение (цель, задачи практики)………………………………………… | 4 |
| 1 | Общая характеристика **ИП «Электромен»**……………………… | 5 |
| 2 | Роль и функции «IT-специалиста»………………………………………. | 6 |
| 3 | Характеристика автоматизированной системы управления....................... | 9 |
| 4 | Создание автоматизированной системы управления в организации................................................................................................ | 12 |
|  | Заключение………………………………………………………………….. | 33 |
|  | Список литературы…………………………………………………………. | 34 |

**ВВЕДЕНИЕ**

Преддипломная практика является составной частью программы подготовки студентов. Основным содержанием практики является выполнение практических учебных, творческих заданий, соответствующих характеру будущей профессиональной деятельности, подготовка выпускной квалификационной работе.

Цель преддипломной практики:

закрепление теоретических знаний и сбор материала для выполнения выпускной квалификационной работы.

Задачи преддипломной практики:

1. Закрепление знаний и навыков, полученных при теоретическом обучении.

2. Овладение опытом работы с конкретными проектными материалами.

3. Сбор необходимых материалов и документов для выполнения выпускной квалификационной работы.

4. Формирование профессиональной позиции обучающегося, его мировоззрения, стиля поведения, освоение профессиональной этики.

5. Приобретение навыков корпоративной работы в составе группы дизайнеров и других специалистов.

Процесс прохождения преддипломной практики направлен на формирование следующих компетенций:

* формирует навыки использовать нормативные правовые документы в своей деятельности;
* стремится к саморазвитию, повышению своей квалификации и мастерства;
* умеет критически оценивать свои достоинства и недостатки, наметить пути и выбрать средства самосовершенствования;
* осознает социальную значимость своей будущей профессии, обладает высокой мотивацией к выполнению профессиональной деятельности;
* способен анализировать и определять требования к проекту;
* способен составлять подробную спецификацию требований к проекту;
* способен синтезировать набор возможных решений задачи или подходов к выполнению проектов;
* способен научно обосновать свои предложения.
  1. **Общая характеристика ИП «Электромен»**

Основной вид деятельности ИП «Электромен» - производство электромонтажных работ. Дата образования организации 6 июня 2014год.

*Электромонтажные работы* являются одной из основных составляющих проведения каких-либо значительных ремонтных процедур.

Любое место, в котором будут проводиться масштабные операции по реконструкции, будь то производственные, офисные, другие нежилые или жилые помещения, подобные действия не будут исключением.

***Виды деятельности, которыми занимается организация:***

1. Штробление. Применимо для бетонных и кирпичных стен.
2. Укладка проводки за подвесными потолками**.**
3. Электромонтаж проводов в стенах из гипсокартона**.** Используется при наличии данных перегородок.
4. Монтаж проводки через скрытие в полах**.** Выполняется до проведения работ по стяжке.
5. Применение металлорукавов и кабель-каналов**.** Выполняется в зданиях с деревянной основой.

Тот или иной метод имеет свои особенности, которые повлияют на составление плана электросети. Главным образом это отразиться на количестве затрачиваемых материалов для осуществления электромонтажа.

Кроме этого, при планировании выполнения подобных операций, следует учитывать возможные нагрузки на сеть, а также количество необходимых розеток, выключателей, электрощитов. После того, как план будет составлен, весь рабочий процесс электриков переходит в практическую часть.

При проведении **электромонтажных работ** на производственных объектах, необходимы: схемы, планы и чертежи помещений и всех предполагаемых приборов, которые в них будут размещаться (различные механизмы и электродвигатели, датчики, счетчики). Вся проводка и электросистема распределяется соответственно, в определенных местах. Вся система затем подключается к панели управления и распределительному электрошкафу, с помощью которых и производится электропитание всей линии электричества. Конечно, можно провести такие**электромонтажные работы** и кустарным методом, не зная и не придерживаясь никаких особых правил и техник. Но такие действия чреваты авариями, несчастными случаями, поломкой оборудования и пожарами.

**2. Роль и функции «IT-специалиста»**

Отрасль IT трудно назвать новой. Первый пик рынка информационных технологий пришёлся ещё на 80-е – правда, в Америке. В Казахстане кибернетика тогда ещё была «продажной девкой империализма» и только начинала стыдливо использоваться в военном деле. Но буквально сразу же после перестройки положение IT в нашей стране резко изменилось. Программисты, как ранее физики, стали модным трендом на международном рынке труда – люди, получившие эту специальность, высоко ценятся за рубежом.

Сегодня отрасль IT – одна из самых перспективных и быстроразвивающихся, не только в Казахстане, но и во всём мире. Каждый год открывает всё новые и новые горизонты для айтишников. Мгновенное распространение смартфонов, интернет-вещей, дополненная реальность – кажется, будто ничто не может остановить бурное развитие отрасли IT. Даже суровые последствия западных санкций не замедлили развитие информационных технологий в Казахстане.

Каждый день каждый из нас многократно прибегает к продуктам информационных технологий [1]. Когда вы читаете эту статью; когда пишите сообщение на телефоне, и он подсказывает вам верную орфографию; когда пользуетесь банковскими карточками – всякий раз вы используете разработки специалистов IT.

Эта сфера настолько прочно вошла в наш повседневный быт, что теперь времена без интернета и мобильных телефонов кажутся многим глухим средневековьем. Тем не менее, для многих людей старшего поколения аббревиатура IT расшифровывается не только как Information Technologies, но и как Terra Incognita. Вокруг "компьютерщиков", как их ещё называют по старинке многие бухгалтеры или врачи, живёт множество мифов.

***Программист*** – человек, занимающийся непосредственно написанием кодов, «тела» любой компьютерной программы. Любая компьютерная программа – это последовательность действий, записанная по определённым правилам. В некотором смысле программу можно назвать специальной инструкцией, написанной на понятном машинам языке: как и на что реагировать при каких условиях. Работа программиста требует уединения и сосредоточения. По мнению многих программистов, их профессия – творческая, сродни работе писателя, музыканта или архитектора. Как и другие творческие профессии, профессия программиста требует постоянного самосовершенствования. Нельзя оставаться хорошим программистом, не продолжая постоянного изучения своей сферы. Хороший специалист должен постоянно изучать новые языки программирования, искать новые решения, пытаться самостоятельно решать незнакомые, непривычные задания.  
Для трудоустройства программисту, в первую очередь, важно иметь хорошее портфолио – образование в большинстве компаний идёт как вторичное требование. Нередко компании ищут программистов, имеющих опыт в разработке тех или иных типов программ: к примеру, "Программист 1С" или "разработчик мобильных приложений". С точки зрения профессионального профилирования разница между этими профессиями не очень существенна – не больше, чем между менеджерами по продажам в различных отраслях.

***Тестировщик*** проверят созданные программистом программы, ища возможные ошибки и уязвимости. Работа тестировщика требует особенного склада ума – тестировщик должен уметь мыслить нестандартно и обожать эксперименты. Тестировщик должен попробовать предпринять все возможные действия, чтобы найти любые уязвимости: к примеру, если тестируется программа для онлайн-банка, тестировщик должен проверять возможность ошибок при введении особенно длинных или коротких имён пользователей, при оперировании особенно крупными или особенно мелкими суммами, при прерывании действий (допустим, не пропадёт ли сумма со счёта, если во время операции у пользователя банка внезапно прервалось соединение) и так далее. Работа тестировщика также является достаточно уединённой.

***Системный администратор*** – специалист по поддержанию уже созданных информационных систем. В отличие от других специалистов в IT, системные администраторы профессионально занимаются ремонтом компьютеров. Но их основная задача – внедрение и ремонт информационных систем, а не отдельных компьютеров.

В круг типовых задач системного администратора обычно входят такие задачи, как подготовка и сохранение резервных копий данных, установка и настройка необходимых обновлений, установка и настройка нового программного обеспечения, поддержание пользовательских учётных записей, ответственность за информационную безопасность, устранение неполадок. Профессия системного администратора требует готовности к работе в режиме многозадачности.

***Системный архитектор*** – это экспертная позиция. Системный архитектор не занимается ни созданием нового программного обеспечения, ни его поддержанием; его работа – это знания. Системный архитектор должен обладать всеми навыками умелого программиста. Его задача на работе – подсказывать наилучшие пути решения, принципы архитектуры той или иной новой программы. *Примерно так это может выглядеть на практике*.  
Допустим, у маркетолога возникает идея: настраивать контекстную рекламу, исходя, в том числе, из запросов близких людей клиентов компании. Первый, с кем нужно будет поговорить маркетологу, – системный архитектор. Исходя из своих знаний, системный архитектор расскажет, из каких именно источников можно будет получать нужные данные, какими способами обеспечить безопасность, а какие инструменты надо будет разработать с нуля. После этого системному архитектору предстоит отстаивать своё решение: обосновывать сроки, размеры бюджета и так далее.

При этом работа системного архитектора не предполагает управленческой нагрузки. Это именно экспертная работа – оценка и отстаивание своего суждения. Если проводить аналогии, системный архитектор чем-то похож на судмедэксперта: он не ведёт следствие, но его суждения совершенно необходимы для успешной работы. Системный архитектор должен обладать глубокой экспертностью в своей сфере.

В отличие от вышеупомянутых IT-специалистов, ***системный аналитик*** имеет достаточно большую управленческую нагрузку. Его задача – составление техзадания для программистов и контроль выполнения работы. *Продолжим предыдущий пример с системным архитектором*.

После того, как системный архитектор составляет основные принципы функционирования новой программы, системный аналитик разделяет эти принципы на отдельные блоки и описывает, какие именно технические ходы нужно предпринять для решения поставленных задач: допустим, использовать конкретный тип запросов с конкретными методами защитной кодировки. После этого системный аналитик распределяет конкретные задачи по подчинённым программистам и контролирует их работу. Если возникают серьёзные трудности, системный аналитик может настоять на перестройке архитектуры системы или изменении цели программы.

Специалист по юзабилити и юзер-экспириенс (UseabIlity и User eXperience) изучает поведение пользователей, их отзывы – и старается разработать наиболее удобный для пользователя (так называемый «user-friendly», «дружелюбный») интерфейс.

Интерфейс – среда взаимодействия между пользователем и программой. Сама по себе программа – это малопонятный неспециалистам набор технического кода. Интерфейс – это то, в чём обычно мы взаимодействуем с программой. К примеру, кнопки «отправить», «входящие» и т.д. – это интерфейс почтовых сервисов. Чем удобнее для пользователей интерфейс, тем с большей вероятностью они будут пользоваться именно им. Неудобный интерфейс гарантированно отпугивает пользователей.

**3. Характеристика автоматизированной системы управления**

MES-системы возникли из потребности оптимизации сложных производственных процессов, таких как производство микроэлектронных интегральных схем, полупроводниковое производство, производственные процессы в фармацевтической, текстильной промышленности, в производстве продуктов питания.

MES-системы представляют собой компьютерно-сетевые архитектуры, способствующие эффективному управлению производственным процессом. MES-системы предоставляют информацию, которая обеспечивает оптимизацию производства продукции от стадии заказа до выпуска конечной продукции. Используя точный сбор текущих данных, MES-система динамически отслеживает весь производственный процесс: инициирует операции, при необходимости оперативно вмешивается в процесс и ведет сбор данных об этом процессе. Функционирование MES-системы обеспечивает оперативное отслеживание постоянно меняющихся условий и устранение лишних затрат. MES-система осуществляет необходимое информационное обеспечение для управления производством в интерактивном режиме.

MES-системы разрабатываются рядом фирм. Фирмы объединены в ассоциацию MESA. Эти фирмы изготовляют различные наборы программных продуктов, в соответствии с требованиями предприятий-заказчиков.

Различают 11 типов блоков, из которых может быть изготовлена конкретная MES система для того или иного производства. Перечислим эти блоки:

Составление расписания операций (Operations/Detail Scheduling) - установление временной последовательности действий для рассматриваемого производственного процесса на данном предприятии при заданных ресурсах.

Составление расписания подразумевает определенную оптимизацию производственного процесса [2].

Распределение ресурсов и слежение за ресурсами (Resource Allocation and Status) - распределение заданий персоналу, машинам и установкам и отслеживание того, что они делают сейчас, и того, что они только что сделали.

Диспетчеризация элементов производства (Dispatching Production Units) - подача команд для транспорта материалов или заказов на определенные участки завода для того, чтобы начать какие-либо процессы или действия.

Управление документооборотом (Document Control) - управление информацией и распределение информации о продуктах, процессах, установках и т.д.

Отслеживание генеалогии продуктов (Product Tracking and Genealogy) - мониторинг прогресса развития создаваемых продуктов и партий, осуществляемый с целью отслеживания полной истории изготавливаемых на заводе продуктов.

Анализ эффективности производства (Performance Analysis) - сравнение результатов завода с метрикми, устанавливаемыми корпорацией, потребителями и организациями, регулирующими рыночные отношения.

Управление персоналом (Labor Management) - управление использованием персонала с точки зрения квалификации, распределения обязанностей и потребностей производства.

Управление оборудованием (Maintenance Management) - планирование и выполнение работ по поддержке работоспособности установок и другого оборудования в соответствии с производственными целями завода.

Управление текущими производственными процессами (Process Management) - управление потоком работ на заводе в соответствии запланированными и текущими процессами.

Управление качеством (Quality Management) - отслеживание и анализ характеристик продуктов и сопоставление их с желаемым идеалом.

Сбор и обработка данных (Data Collection/Acquisition) - мониторинг, обработка и упорядочивание данных о процессах, материалах и операциях, получаемых от персонала, машин и систем управления.

Практика применения MES-систем демонстрирует следующие их достоинства:

уменьшение длительности производственного цикла, в среднем на 45%;

сокращение времени ввода (entry time), на 75% или больше;

уменьшение незавершенного производства (НЗП), на 24%;

уменьшение бумажного документооборота между сменами, в среднем на 61%;

сокращение времени проектирования (lead time), в среднем на 27%;

уменьшение работы с бумажной документацией и потерь на копировальные работы, в среднем на 56%;

уменьшение дефектности производимой продукции, в среднем на 18%.

Перспективы развития MES-систем связаны с общими тенденциями развития компьютерных сетей. Программные продукты MES-систем пишутся на объектно-ориентированных языках. Имеется определенная тенденция к использованию MES-систем в географически распределенных предприятиях (при этом производитель становится как бы ближе к заказчику и к субподрядчикам) на основе использования передовых достижений компьютерно-сетевых технологий.

В США сформирован поддерживаемый государством консорциум Solution for MES-Adaptable Replicable Technology (NIIIP/SMART), который предназначен для разработки стандартов для создания инфраструктуры распределенных предприятий будущего.

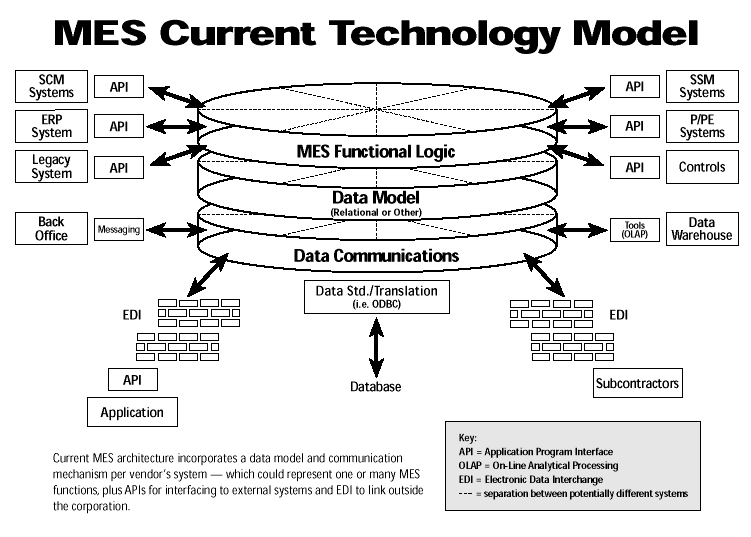


Рисунок 1 - Современная компьютерно-сетевая архитектура MES – системы

Современное микроэлектронное производство представляет собой сложный процесс. На уровне отдельного цеха этот процесс включает в себя обработку партий полупроводниковых пластин на установках, которые расположены на различных участках. Необходима транспортировка партий между участками, необходим учет времен обработки партий на установках, возможных временных задержек обработки, необходим контроль качества партий, исправление дефектов обработки, учет возможных выходов из строя оборудования.

В настоящее время интенсивно развиваются компьютеризированные системы управления производственными процессами. Более того, предприятия, не использующие современные технологии, рискуют оказаться в роли "вечно отсталых" и, в конечном итоге, могут не выдержать рыночной конкуренции.

**4. Создание автоматизированной системы управления в организации**

Охарактеризуем особенности технологического процесса производства партий. Предварительно введем понятия (изделия, партии, технологические маршруты, маршрутные листы), которые потребуются при описании производственного процесса.

*Изделие* - полупроводниковая микросхема, выполняющая заданные разработчиком функции. Каждое изделие характеризуется индивидуальным технологическим маршрутом, который заранее прорабатывается.

*Партия* - группа пластин, помещенных в специальный контейнер, имеющая номер и собственный маршрутный лист. Партия в полупроводниковом производстве обрабатывается и транспортируется в соответствии маршрутным листом. Все множество партий можно разбить на группы по принадлежности к определенному изделию (Рис.4). Принадлежность партии к определенному изделию находит отражение в номере. Номер партии состоит из двух частей: первая часть - номер изделия, вторая - порядковый номер в пределах данного изделия [3].

*Технологический маршрут* - последовательность операций, при выполнении которой на поверхности кремниевой пластины формируется заданная структура. Каждое изделие характеризуется индивидуальным технологическим маршрутом. Прохождение партии по маршруту отслеживается в настоящей системе управления с помощью маршрутного листа.

*Маршрутный лист* - документ, неразрывно связанный с партией. Маршрутный лист представляет собой таблицу со следующими атрибутами: номер операции, название операции, время поступления партии на участок, параметры режима обработки. На маршрутном листе, закрепляемом за партией, ставится номер партии. Каждой партии соответствует свой маршрутный лист, который прикреплен к коробке с пластинами. На протяжении всего периода существования партии маршрутный лист следует за ней. В нем регистрируется прохождение партии по технологическому маршруту. Как правило, маршрутные листы тиражируются типографским методом.

Каждому изделию ставится в соответствие индивидуальный технологический маршрут. В пределах одного изделия все партии имеют одинаковый маршрутный лист. Элементарная единица управления - партия, т.е. система управления отслеживает прохождение каждой партии по маршруту. Партий, представляющих конкретное изделие (изготавливаемых по данному маршруту), может быть много (до 500). Номер партии пластин составляется из номера изделия и номера партии в пределах изделия. Процесс производства партий контролируется посредством регистрации потоков партий в регистрационных журналах, в бланках незавершенного производства и в маршрутных листах. Регистрация необходима для контроля прохождения партий, в частности, при необходимости партию можно быстро найти. Каждый участок имеет свой регистрационный журнал. В нем персонал установок отмечает партии, обработанные на данном участке. По нему считают проработку партий за смену, за месяц. Для "нулевых" партий (запускаемых на начало обработки на входе цеха 201) есть свой регистрационный журнал. Для полностью обработанных партий (на выходе цеха 201) также есть отдельный регистрационный журнал. Маршрутный лист - документ, неразрывно связанный с партией. Каждой партии соответствует свой маршрутный лист, который прикреплен к коробке с пластинами. На протяжении всего периода существования партии маршрутный лист следует за ней. Он представляет собой таблицу, содержащую: номер операции, название, время поступления партии, параметры режима обработки, колонку для росписи или табельного номера. Записи в маршрутные листы вносят операторы участков. Операторы регистрируют выполнение технологических операций: по окончании обработки партии на конкретной операции оператор ставит свой табельный номер и параметры режима обработки [4]. Ни распреды, ни диспетчер, ни начальник смены не вносят никаких записей в маршрутный лист. Бумажный "документооборот" (регистрация в журналах, бланках незавершенного производства и маршрутных листах) до определенной степени избыточен, что обеспечивает надежность контроля производственного процесса. Важное понятие, характеризующее производственный процесс - незавершенное производство (НЗП). Незавершенное производство - это продукция (партии кремниевых пластин), не прошедшая до конца весь технологический цикл. Учитывается НЗП в регистрационных журналах, бланках незавершенного производства и в маршрутных листах. В настоящее время незавершенное производство велико - количество партий НЗП в 4 раза превышает месячную пропускную способность цеха. Наличие большого НЗП приводит к тому, что растет длительность производственного цикла. Большая величина НЗП связана с тем, что административный персонал цеха и завода, желая увеличить загрузку оборудования, вводит в производственный процесс все новые и новые партии. При этом уже обрабатываемые партии скапливаются в "узких местах" производственного процесса, что создает "рецидивы" накопления НЗП. В данное время существует много задержек в транспорте и обработке партий. Эти задержки можно разделить на следующие группы:

обеденные перерывы.

совещания административного персонала цеха.

сдача и приемка смен.

оформлением списка, где какие партии находятся, и поиск партий.

сбои оборудования.

необходимость принимать повторные решения при сбое оборудования.

и т.п.

Первые три группы задержек можно устранить посредством целенаправленной организации рабочего дня персонала (см. ниже).

Остальные группы задержек являются следствием несовершенства системы управления и контроля. Контролировать их людьми очень трудно, поэтому вводится избыточная загрузка цеха партиями, чтобы обеспечить работой все участки в случае сбоев некоторых. Эта загрузка приводит к большому НЗП, что в свою очередь увеличивает длительность цикла. Автоматизированная система управления производственным процессом призвана сократить эти задержки. Электронная система позволит повысить четкость отслеживания партий и планирования работ, в результате этого можно будет уменьшить НЗП, не опасаясь перерывов в работе участков. Уменьшение НЗП в свою очередь приведет к уменьшению длительности производственного цикла.

Оценим количественно задержки в транспорте партий и задержки в управлении транспортом и производством партий.

Задержки в транспорте партий, обусловленные неоптимальностью работы распредов (в расчете на сутки). Пересменка - 2\*20 минут = 40 минут, обеденные перерывы 30\*4 = 2 часа. Составление списка партий, поиск партий (полагаем пока условно, в дальнейшем эти цифры можно будет уточнить) - 2 часа в сутки.

Итого: задержки в транспорте партий составляют примерно 5 часов в сутки.

Устранить эти задержки в электронной системе управления можно следующим образом (для определенности рассматриваем пример подцеха 131): Задержки, связанные с пересменками и обеденными перерывами, можно устранить, вводя дополнительно одного распреда и вводя скользящий график работы. А именно, считаем, что первый распред начинает работу в 7-00, второй - в 7-30, третий - в 8-00. Смена первого, второго и третьего распредов заканчивается в 19-00, 19-30 и 20-00 соответственно. Аналогично смещаются относительно друг друга и обеденные перерывы распредов. В результате, в подцехе все время будут находиться, по крайней мере, два распреда.

Задержки, связанные с составлением списков должны устраниться в электронной системе автоматически, так как ведется учет всех партий. Составление списка (если это потребуется будет сводиться к распечатке соответствующего документа).

Оценим задержки в управлении (в сутки): совещания административного персонала цеха - 2\*30 мин = 1 час, обеденные перерывы - 4\*30мин =2 часа, сдача и приемка смен (полагаем пока условно, в дальнейшем эти цифры можно будет уточнить) - 2\*30 мин = 1 час, сбои оборудования, необходимость принимать повторные решения при сбое оборудования и т.п. (полагаем пока условно, в дальнейшем эти цифры можно будет уточнить) = 1 час.

Итого - 5 часов в сутки не принимается решение о транспорте партий и обработке партий.

Эти задержки можно устранить следующим образом. Сократить или заменить краткой электронной конференцией (или телеконференцией) совещания административного персонала цеха. В крайнем случае, вынести эти совещания за рамки смены, т.е. проводить совещания не во время своей смены, а течение последних 30 минут предыдущей смены. Обеденные перерывы распределить между административным персоналом, так чтобы не было перерывов в управлении транспортом партий.

Наличие электронной системы позволит обеспечить указанное устранение задержек, так как большинство решений о транспорте партий будет приниматься именно электронной системой и нагрузка на административный персонал радикально снизится. Сокращение (или даже полное устранение совещаний) будет возможно, так как многие вопросы могут быть сняты точным контролем и обеспечением информации на рабочие места административного персонала через терминалы компьютерно-сетевой системы управления.

Электронная система обеспечит более строгий учет за транспортировкой и обработкой партий. При этом снимаются проблемы поиска потерявшихся партий, необходимости принятия повторных решений. Система позволит оперативно и гибко реагировать на сбои оборудования, меняя очередность прохождения партий на "узких" участках.

Кроме того, система позволит собирать точную статистику по обработке партий, пропускной способности участков, дефектности; появится возможность прогнозирования, как на краткосрочную, так и на долгосрочную перспективу.

Современное микроэлектронное производство представляет собой сложный процесс. На уровне отдельного цеха этот процесс включает в себя обработку партий полупроводниковых пластин на установках, которые расположены на различных участках. Необходима транспортировка партий между участками, необходим учет времен обработки партий на установках, возможных временных задержек обработки партий. Важно учесть приоритеты изготовления партий. Необходим контроль качества партий, исправление дефектов обработки партий, учет возможных выходов из строя установок, обрабатывающих партии [5].

В этом процессе важно также найти компромисс между максимальной загруженностью установок цеха и минимальным циклом изготовления партий. Влияние загруженности установок на длительность цикла изготовления сводится к следующему: для увеличения загрузки установок нужно запускать в производственный процесс как можно большее число партий, но при этом партии будут скапливаться в "узких местах", в результате чего возникают большие очереди на установках, и длительность цикла изготовления отдельной партии будет возрастать.

Помимо этого, необходимо осуществлять мониторинг качества работы участков, собирать и обрабатывать статистику о процессе производства партий.

Итак, необходима система управления производством партий. Эта система должна обеспечивать принятие решений о транспорте и обработке партий. При этом количество принимаемых решений в единицу времени достаточно велико: оценки для типичного цеха дают величины порядка 108 элементарных принятий решений в месяц. Следовательно, система управления должна быть оперативной. Более того, эта система должна быть достаточно "интеллектуальной", с тем, чтобы учесть указанные выше требования. Оперативность и "интеллектуальность" системы управления подразумевают, что она должна базироваться на современных компьютерно-сетевых технологиях и использовать "интеллектуальные" схемы принятия решения.

В настоящем сообщении предлагаются принципы построения автоматизированной компьютерно-сетевой системы управления транспортом и обработкой партий отдельного цеха.

При разработке этих принципов мы основывались на общих методах и подходах, предложенных в системах ситуационного управления [1-4], и использовали идеи активно развивающегося сейчас за рубежом направлений Manufacturing executing systems (MES-системы) и Системы диспетчерского управления и сбора данных (SCADA-системы).

Общие требования к автоматизированной системе управления производственным процессом состоят в следующем. АСУ ПП должна:

Осуществлять контроль прохождения партий по технологическому маршруту.

Оптимизировать прохождение партий по различным маршрутам, чтобы все они обрабатывались в срок, с наилучшим выходом годных, и предприятие работало с наибольшей производительностью.

Собирать и обрабатывать статистику о качестве работы участков, принимаемых решениях и их результатах, и на основании этого изменять свое поведение и информировать руководство о возможных причинах неоптимальной работы предприятия.

В рамках подготовительного этапа была проведена следующая работа:

Проведен анализ принципов построения аналогичных систем управления (результаты анализа отражены в обзоре литературы).

Проанализирована структура технологического процесса на **ИП «Электромен»**

, сложившаяся к настоящему времени.

Предложена архитектура компьютерно-сетевой АСУ ИП «Электромен»

, предназначенной для управления технологическим процессом производства партий.

Произведен объектно-ориентированный анализ технологического процесса производства партий. В результате анализа разработана объектная модель производственного процесса. Объектная модель представляет собой концептуальную основу разрабатываемой компьютерно-сетевой системы управления производством партий (см. технологическую часть).

На основе проведенного анализа можно рекомендовать следующие общие принципы построения разработки автоматизированной системы управления производством партий (АСУ) ИП «Электромен»

.

Система управления должна включать в себя:

Реляционную базу данных для хранения всей информации о производственном процессе. База данных нужна для того, чтобы информация о состоянии производства была представлена в некотором стандартном формате.

Центральную программу управления. Эта программа должна быть предназначена для:

Принятия решений о транспорте и обработке партий.

Отслеживания и сокращения задержек с целью сокращения производственного цикла.

Сбора статистики (пропускная способность участков, средний объем НЗП, задержки).

Контролирования объема НЗП.

Формирования очереди к потенциально узким участкам.

Своевременного предупреждения "рецидивов" накопления НЗП.

Планирования на краткосрочную и долгосрочную перспективу.

Предполагается, что эта программа управления должна быть расположена на центральном компьютере и помогать принимать решения диспетчеру и персоналу участков.

Локальные терминалы на участках, предназначенные для обмена информацией между персоналом участков и центральным компьютером.

Имитационную программу, предназначенную для анализа производственного процесса и поиска оптимальных режимов управления производством. Системы регистрации и мониторинга производства партий.

Общение программного обеспечения с пользователями должно происходить посредством максимально удобного и интуитивно понятного интерфейса, например, типа MS Windows. Анализ развития компьютерных технологий и опыта зарубежных фирм по разработке MES - и SCADA-систем показывает, что АСУ ПП должна базироваться на компьютерно-сетевых технологиях и использовать методы объектно-ориентированного программирования [6].

Общая схема системы управления представлена на Рис.6. Система предназначена для управления производством партий в отдельном цехе

Центральный блок управления

Имитационная модель предприятия

Реляционная база данных

Программа управления производством

Управляемое производство

Участок №1

Терминал №1

Участок №2 Терминал №2

Участок №N Терминал №N

Установка №1

Установка №2

Установка №3

Группа контроля

Тренинг

Рисунок 2 –Архитектура системы управления производством партий в цехе

Стрелками показаны информационные потоки. Двойными стрелками показаны информационные потоки между программными модулями в центральном блоке управления. Блоки "Тренинг" и "Имитационная модель" не входят в систему управления цехом, а только лишь связаны с ней.

Блоки системы управления описаны ниже.

*Группа контроля* осуществляет общее руководство управлением производством партий в цехе. В группу контроля входит диспетчер и, возможно, другой административный персонал. Группа контроля интерактивно взаимодействует с программой управления производством. Программа осуществляет управление потоками партий в режиме "автопилота". Группа контроля имеет право вмешиваться в режим управления. В случае необходимости группа контроля принимает управление на себя и выполняет управление "в ручном режиме".

*Программа управления производством* предназначена для управления потоками партий в режиме реального времени. Эта программа отслеживает весь производственный процесс.

Предварительно, модули и функции программы управления можно описать следующим образом:

Программа должна содержать *модуль принятия решения*, который обеспечивает решения о транспорте партий и загрузке установок.

Программа должна содержать *модуль* *контроля участков.* Этот модуль отслеживает соответствие между инструкциями программы управления и работой участков.

Программа отслеживает весь процесс производства вплоть до конкретного исполнителя.

В *реляционной базе данных* собирается информация о процессе производства партий. Эта информация используется как в программе управления производством, так и в имитационной модели.

*Имитационная модель* предназначена для исследования функционирования и оптимизации самой системы управления.

Предполагается, что разработанная программа (или ее упрощенная версия) будет использоваться для *тренинга* персонала. Для отдельных цехов блоки "Имитационная модель" и "Тренинг" могут отсутствовать.

*Персонал участка* выполняет операции в соответствии с инструкциями программы и ставит в известность программу о выполненных операциях. Программа регистрации процесса производстваставит обработанную партию в очередь на все установки, соответствующие дальнейшему маршруту партии. Приоритеты рассчитываются только в случае конфликтов. При расстановке приоритетов учитываются мощности установок (скажем, партия с большим приоритетом должна обрабатываться на более мощной установке). Программе известно нормативное время обработки партии, и если потери времени *существенны*, то информация об этом поступает в группу контроля.

На каждом участке есть терминал, через который осуществляется взаимодействие между программой и персоналом участка.

Предполагается, что регистрация выполнения операций осуществляется с помощью внесения оператором участка информации в базу данных.

Как и в имеющейся системе, транспорт партий осуществляется распредами, а анализ забракованных партий - ведущим технологом.

При формировании дисциплины очереди необходимы знания об определенных параметрах партии и производственного процесса. Выделим эти параметры.

Основные параметры (атрибуты партий, атрибуты установок, характеристики производственного процесса), которые требуется учесть при управлении потоками партий:

Время задержки прохождения партией данной операции (отрицательные времена тоже важны), Δ *t*зад;

Априорная важность партии, *Imp* (обычная партия *Imp* = 1, важная партия *Imp* = 2,3,…) априорную важность партии устанавливает группа контроля по заданию руководства;

Процент прохождения маршрута, Pr;

Прогноз следствий невыполнения;

Коэффициент нагрузки, *r* = *I*заявок / *I*обслуживания (*I* - интенсивность). *r* зависит от НЗП. Средняя длина очереди зависит от *r:* *l* = [1 - *r*] -1 Разумно выбирать *r* ≤ 0,5 - 0,6, в этом случае длина очереди *l* = ≤ 2 - 2,5.

Предварительная схема управления потоками партий такова.

Если партия прошла *n*-й микроцикл, то в программу управления производством поступает рапорт о выполнении микроцикла и заявка на выполнение *n* +1-ого микроцикла.

Считаем, что схема имеет вид правил:

Если …., то …., иначе.

Некоторые из правил требуют выполнения определенных расчетов. Схема принятия решения основана на учете указанных выше параметров.

Опишем конкретный вариант упрощенной схемы принятия решения.

*Если* есть свободная установка для выполнения первой операции *n* +1-ого микроцикла, и эта установка одна, *то* заявка подается на эту установку.

Иначе

*Если* свободных устройств несколько, *то* заявка подается на установку с наибольшим приоритетом для данной партии.

Иначе

*Если* все устройства, соответствующие первой операции *n* +1-ого микроциклазаняты, *то* партия подается в очередь на все эти установки.

При возникновении свободного устройства обрабатывается партия с наибольшим текущим приоритетом.

В простейшем случае текущий приоритет есть Δ *t*зад \* *Imp.*

Еще раз подчеркнем, что необходим отдельный анализ по разработке дисциплины очереди. Отметим, что известен ряд методов формирования систем управления, подобных разрабатываемой системе. Одна из серьезных проработок: методы ситуационного управления, описанные выше. Второй пример - схемы классифицирующих систем (Classifier Systems), представляющие собой самообучающиеся системы принятия решения. Эти системы основаны на использовании правил типа "Если …., то …. ". Совокупность таких правил оптимизируется как путем обучения (переоценка приоритетов правил), так и путем эволюции (эволюционный поиск новых правил). Еще одну возможность предоставляют системы поддержки принятия решений, основанные на активно применяемом в последнее время нейросетевом подходе.

К созданию системы был выбран объектно-ориентированный подход. Этот подход предусматривает спиральную модель создания информационных систем, которая состоит из следующих этапов:

Анализ требований.

Проектирование.

Реализация (программирование).

Тестирование.

Спиральная модель подразумевает, что эти этапы циклически повторяются, на каждом цикле в программу вносятся исправления и новые возможности. Этот способ создания программного обеспечения позволяет достаточно быстро получить первую рабочую версию программы, а на следующих циклах реализовывать дополнительно возникающие требования и оптимизировать существующие части программы, не нарушая ее целостности и работоспособности.

Объектно-ориентированный анализ

Ниже приводятся результаты первого этапа - анализа требований к проектируемой системе. При анализе использовались стратегии и образцы, разработанные Питером Коудом, а также методология OMT (Object Modeling Technique). Результатами анализа являются назначение системы, характерные свойства, объектная модель [7].

Назначение системы:

Обеспечить контроль и регистрацию производственного процесса производства партий пластин. Вести учет за прохождением партий полупроводниковых пластин, как материальных ценностей.

Система должна:

Регистрировать передачу партии полупроводниковых пластин на участок.

Составлять график изготовления для каждой партии и следить за его выполнением.

Фиксировать результаты контроля и измерений для последующего анализа.

Фиксировать время начала и окончания выполнения технологической операции.

Определять очередность и времена запуска партий на обработку.

Осуществлять управление очередностью изготовления партий на установках с длительными операциями.

Формировать плановый срок для каждой операции обработки и следить за его соблюдением.

Оценивать потери, с их разложением на каждом участке для каждой смены.

В процессе анализа были выявлены следующие объекты:

Группа контроля - лицо или группа лиц, осуществляющее планирование запуска партий из НЗП в производство и материально ответственное за партии, хранящиеся в НЗП.

Транспорт - служба, занимающаяся транспортировкой партий между участками.

Представитель участка - лицо или группа лиц, материально ответственное за партии находящиеся на участке.

Ведущий технолог - лицо, принимающее решения в случае забраковки партий на контроле.

Установка - устройство, осуществляющее операции над партиями.

Участок - административная единица, выполняющая некий набор операций над партиями.

Участок НЗП - участок, на котором партии хранятся между микроциклами, в НЗП поступают новые партии и готовые партии.

Шкаф - место на участке НЗП, в котором партии хранятся между микроциклами.

Контроль - 1) участок, на котором осуществляется операция контроля,

2) операция контроля, на которой выявляется брак.

Партия - объект, подвергающийся операциям на участках в соответствии с маршрутным листом.

Маршрутный лист - документ, содержащий список операций, которые партия должна пройти в процессе изготовления.

Операция - действие, изменяющее готовность партии, этап изготовления партии согласно маршрутному листу.

Хранение - операция, которой подвергаются партии между микроциклами.

Очередь - место на технологическом участке, где находятся партии в ожидании очередной операции.

Реставрация - изменение в технологическом маршруте, возникающее в связи с браком, осуществляется ведущим технологом.

Микроцикл - часть технологического маршрута; между микроциклами партия может храниться длительное время.

Забраковка - признание партии непригодной для дальнейшей обработки, ее удаление из плана производства и, возможно, запуск новой идентичной партии.

Входной считыватель штрихкода - устройство, фиксирующее поступление партии на участок.

Выходной считыватель штрихкода - устройство, фиксирующее передачу партии с участка на транспорт.

Технологический участок - административная единица, выполняющая некий набор операций над партиями на установках, составляющих данный участок.

Терминал - средство общения между системой управления и персоналом.

Цех - помещение, в котором осуществляется производство.

Участок ВТ - место нахождения Ведущего технолога.

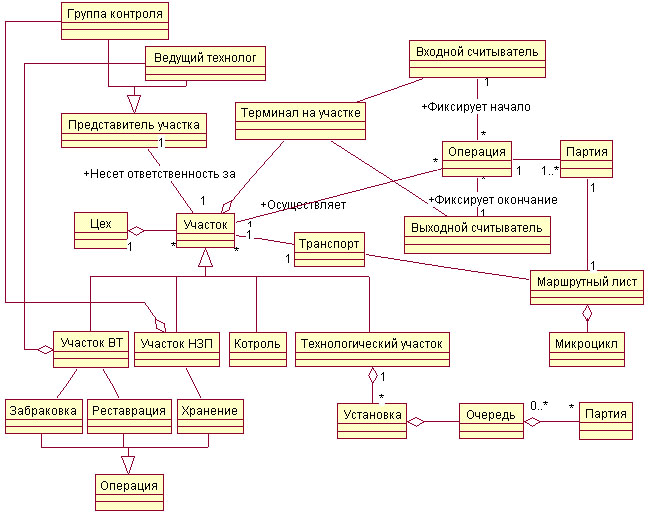


Рисунок 3 - Объектная модель, характеризующая объекты системы управления и связи между ними

Объектная модель показывает планируемую структуру программы, с точки зрения объектно-ориентированного подхода. В этой модели показаны объекты, важные для проектируемой системы и связи между ними. Рис.8 поясняет нотацию (систему обозначений) Unified, в которой приведена модель.

Рисунок 4 - Обозначения, принятые в объектной модели

Участок

Имя объекта

Атрибуты объекта (определяются на этапе проектирования)

Методы объекта (определяются на этапе проектирования)

– агрегация: означает, что один объект является частью другого

– ассоциация: означает, что один объект "знает" о существовании другого, т.е. объекты каким-либо образом взаимодействуют друг с другом

– наследование: означает, что объект-наследник обладает свойствами объекта предка, а также своими собственными свойствами, отличающими его от предка и других наследников

В процессе проектирования и разработки программы управления к модели будут добавляться новые объекты и связи, для объектов будут определяться атрибуты и методы, взаимодействие между объектами будет описано через сценарии. Конечным результатом усложнения модели будет программа управления производством. Все объекты, приведенные в данной модели, будут являться частями будущей программы [8].

В данной программе схема работы представлена двумя основными алгоритмами описывает алгоритм выполнения технологической операции.



Рисунок 5 - Алгоритм выполнения технологической операции

При разработке программы использовались преимущества библиотеки классов MFC, предоставляющей программисту набор мощных инструментов. Библиотека инкапсулирует наиболее важные структуры и функции интерфейса прикладного программирования в группе классов, пригодных для многократного использования.

Использование объектно-ориентированного подхода дает многочисленные преимущества в сравнении с традиционным стилем программирования:

Инкапсуляция кода и данных внутри класса;

Наследование;

Разрешение коллизий имен функций;

Уменьшается общий объем текстов.

Основные возможности библиотеки MFC:

Поддержка всех функций, управляющих элементов и ссобщений Windows, графических примитивов GDI, меню и окон диалога;

Использование соглашения об именах API Windows, при котором по имени класса становится ясным его предназначение;

Избавление от необходимости применять оператор switch/case, который часто является источником ошибок. Каждому сообщению ставится в соответствие функция-член класса;

Архитектура, основанная на широком использовании обработки исключительных ситуаций, делает код приложения более устойчивым к ошибкам.

Cobject - базовый класс, широко используемый при разработке приложений Windows.

CCmdTarget -

CWnd - родительский класс для всех окон.

CWinApp - родительский класс для приложения.

CDialog - родительский класс для окон диалога.

Как видно из иерархии классов, программа состоит из четырех основных классов:

CAngstremApp - основной класс. Разделен на два файла - заголовочным для него является файл cangstremapp. h, файл реализации - cangstrem. cpp.

CMainFrame - этот класс, производный от CFrameWnd, используется для управления главным окном приложения.

CAngstremView - является наследником класса CView. Класс CView, производный от CWnd, служит базовым для пользовательских классов отображения. Отображение (view) служит промежуточным звеном между документом и пользователем. Обычно отображение - это дочернее окно главного окна.

Первый шаг в разработке прикладной информационной системы заключается в точном формулировании целей внедрения машинной обработки. Требования к системе еще не создают полной картины. В постановке задачи, разумеется, должны принять участие как представители организации-заказчика, так и те, кто будет заниматься проектированием системы. Необходимо, чтобы этот процесс был гибко организован и продолжался в течении достаточно длительного времени, поскольку на любом этапе разработки или внедрения могут вскрываться ранее не предусмотренные трудности, требующие внесения определенных изменений в проект [9]. В то же время с целью упрощения разработки в договор следует включить пункт, который запрещал бы радикальный пересмотр требований, на стадии реализации системы. В этом же пункте могут быть оговорены условия внесения несущественных изменений. Все договоренности должны оформляться в официальном порядке.

Разработка информационной системы на базе ЭВМ, включает такой обязательный элемент, как определение формы, в которой должна быть представлена информация, хранящаяся в хранилище. Кроме того, необходимо установить какие варианты данных будут использоваться на различных этапах, начиная от сбора исходной информации и заканчивая выдачей результатов. Очевидно, формы, применяемые на этапах сбора данных, их корректировки или выборки с целью проведения той или иной обработки, могут отличаться друг от друга. Варианты данных должны охватывать все входящие в автоматизированную систему и исходящие из нее информационные потоки. Постановка задачи должна охватывать весь процесс сбора, хранения, использования и окончательного представления данных. В ней должны также учитываться ограничения на время обработки, финансовые затраты и вычислительные ресурсы. Кроме того, в постановке задачи следует предусмотреть возможность последующей модернизации системы. Оценка будущих потребностей должна осуществляться совместно как проектировщиками системы, так и ее потенциальными пользователями, поскольку необходимо принимать компромиссные решения, касающиеся соотношения между универсальностью и эффективностью, гибкостью организации данных и стремлением обеспечить текущие потребности.

Этап проектирования в разработке информационной системы представляет собой наиболее ответственный процесс. Результат этого этапа накладывает самый большой отпечаток на всю систему. И грамотное проведение этого процесса в последствие намного облегчит труд разработчиков на следующих этапах. Поэтому наиболее подробно рассмотрим этот этап в разработке программного обеспечения информационной системы.

В любой инженерной дисциплине под проектированием обычно понимается некий унифицированный подход, с помощью которого мы ищем пути решения определенной проблемы, обеспечивая выполнение поставленной задачи. В контексте инженерного проектирования цели проектирования определяются как создание системы, которая:

удовлетворяет заданным (возможно, неформальным) функциональным спецификациям;

согласована с ограничениями, накладываемыми оборудованием;

удовлетворяет явным и неявным требованиям по эксплуатационным качествам и ресурсопотреблению;

удовлетворяет явным и неявным критериям дизайна продукта;

удовлетворяет требованиям к самому процессу разработки, таким, например, как продолжительность и стоимость, а также привлечение дополнительных инструментальных средств.

По предположению Страуструпа: "Цель проектирования - выявление ясной и относительно простой внутренней структуры, иногда называемой архитектурой... Проект есть окончательный продукт процесса проектирования". Проектирование подразумевает учет противоречивых требований. Его продуктами являются модели, позволяющие нам понять структуру будущей системы, сбалансировать требования и наметить схему реализации.

Ясно, что не существует такого универсального метода, который бы провел инженера-программиста по пути от требований к сложной информационной системе до их выполнения. Проектирование такой информационной системы отнюдь не сводится к слепому следованию некоему набору рецептов. Скорее это постепенный и итеративный процесс. И тем не менее использование методологии проектирования вносит в процесс разработки определенную организованность. Как уже утверждалось выше, инженеры-программисты с момента зарождения программирования до с настоящих дней разработали десятки различных методов, которые можно классифицировать по трем категориям:

процедурное программирование;

метод структурного проектирования "сверху вниз";

метод потоков данных;

объектно-ориентированное проектирование.

Структурное проектирование "сверху вниз" основано на алгоритмической декомпозиции. При таком подходе к проектированию происходит обычное разделение алгоритмов, где каждый модуль системы выполняет один из этапов общего процесса. Структурный подход в проектировании в течение большого промежутка времени был практически лидером среди имеющихся методов разработки программного обеспечения. Но в связи с резким скачком развития аппаратного обеспечения, появлением очень мощных вычислительных комплексов, и как результат - многократное усложнение программного обеспечения, структурное проектирование перестало быть эффективным. Значение структурного подхода осталось прежним, но было замечено, что структурный подход не работает, если объем программы превышает приблизительно 100000 строк. Структурный подход не позволяет выделить абстракции и обеспечить ограничение доступа к данным; он также не предоставляет достаточных средств для организации параллелизма. Структурный метод не может обеспечить создание предельно сложных систем, и он, как правило, неэффективен в объектных и объектно-ориентированных языках программирования. Однако, в некоторых случаях его применение в этих языках оправдывает себя.

В методе потоков программная система рассматривается как преобразователь входных потоков в выходные. Метод потоков данных, как и структурный метод, с успехом применялся при решении ряда сложных задач, в частности, в системах информационного обеспечения, где существуют прямые связи между входными и выходными потоками системы и где не требуется уделять особого внимания быстродействию. Более подробно остановимся на третьем методе проектирования, т.к он на данный момент является самым современным и эффективным средством проектирования информационных систем. Объектно-ориентированное проектирование - это методология проектирования, соединяющая в себе процесс объектной декомпозиции и приемы представления логической и физической, а также статической и динамической моделей проектируемой системы.

В данном определении содержатся две важные части: объектно-ориентированное проектирование, во-первых основывается на объектно-ориентированной декомпозиции, во-вторых использует многообразие приемов представления моделей, отражающих логическую (классы и объекты) и физическую (модули и процессы) структуру системы, а также ее статические и динамические аспекты.

Именно объектно-ориентированная декомпозиция отличает объектно-ориентированное проектирование от структурного; в первом случае логическая структура системы отражается абстракциями в виде классов и объектов, во втором – алгоритмами [10].

Процесс объектно-ориентированного проектирования не сводится к одностороннему движению "сверху вниз" или "снизу-вверх". Хорошо структурированные сложные системы можно создать методом "возвратного проектирования".

В дальнейшем будем называть этот процесс микропроцессом. В этом методе основное внимание уделяется процессу поступательного итеративного развития путем совершенствования различных, но, тем не менее, совместимых между собой логических и физических моделей системы. Возвратное проектирование составляет необходимую основу процесса объектно-ориентированного проектирования.

Микропроцесс объектно-ориентированной разработки приводится в движение потоком сценариев и архитектурных продуктов, которые порождаются и последовательно уточняются в макропроцессе. Микропроцесс, по большей части, - повседневный труд отдельного разработчика или небольшого коллектива разработчиков.

Микропроцесс относится в равной степени к программисту и архитектору программной системы. С точки зрения программиста, микропроцесс предлагает руководство в принятии бесчисленного числа ежедневных тактических решений, которые являются частью процесса создания и подгонки архитектуры системы. С точки зрения архитектора, микропроцесс является основой для развития архитектуры и опробования альтернатив. В микропроцессе традиционные фазы анализа и проектирования умышленно перемешаны.



Рисунок 6- Микропроцесс проектирования

Как показано на рисунке 6, микропроцесс обычно состоит из следующих видов деятельности:

выявление классов и объектов на данном уровне абстракции;

выяснение семантики этих классов и объектов;

выявление связей между этими классами и объектами;

спецификация интерфейса и реализация этих классов и объектов.

Рассмотрим их более подробно:

1. Выявление классов и объектов.

*Цель:* Выявления классов и объектов состоит в том, чтобы найти границы предметной области. Кроме того, эта деятельность является первым шагом в продумывании объектно-ориентированной декомпозиции разрабатываемой системы. Этот шаг в анализе применяется тогда, когда обнаруживаются абстракции, составляющие словарь предметной области и, тем самым, ограничивая задачу, решая, что важно, а что - нет. Такие действия необходимы при проектировании, когда мы изобретаем новые абстракции, которые являются составными частями решения. Переходя к программной реализации, мы применяем процедуру выявления, чтобы изобрести более простые абстракции, из которых строятся более сложные, и обнаружить общие черты существующих абстракций, дабы упростить архитектуру системы.

*Результаты:* Главным результатом этого шага является обновляющийся по мере развития проекта словарь данных.

2. Идентификация семантики классов и объектов.

*Цель:* Определить поведение и атрибуты каждой абстракции, выявленной на предыдущем шаге. При этом мы уточняем намеченные абстракции, продуманно и измеримо распределяя между ними обязанности.

*Результаты:* Во-первых, уточняется словарь данных, с помощью которого мы изначально присвоили обязанности абстракциям. В ходе проектирования можно выработать спецификации к каждой абстракции, перечисляя имена операций в протоколе каждого класса. Затем, необходимо выразить интерфейсы этих классов на языке реализации. Например, для C++ это означает создание. h-файлов, в Ada - спецификаций пакетов, в CLOS - обобщенных функций для каждого класса, в Smalltalk - это объявление, но не реализация методов каждого класса. В добавление к этим, по сути тактическим решениям, мы составляем диаграммы объектов и диаграммы взаимодействий, передающие семантику сценариев, создаваемых в ходе макропроцесса. Эти диаграммы формально отражают раскадровку каждого сценария и, таким образом, описывают явное распределение обязанностей среди взаимодействующих объектов. На этом шаге впервые появляются конечные автоматы для представления некоторых абстракций.

3. Выявление связей между классами и объектами.

*Цель:* Уточнить границы каждой обнаруженной ранее в микропроцессе абстракции и опознать все сущности, с которыми она взаимодействует. Это действие формализует концептуальное и физическое размежевание между абстракциями, начатое на предыдущем шаге.

*Результаты:* Основными результатами этого шага являются диаграммы классов, объектов и модулей. Хотя в конце концов необходимо выразить решения, принятые при анализе и проектировании, на языке программирования, диаграммы дают более широкий обзор архитектуры и, кроме того, позволяют раскрыть отношения, которые с трудом формулируются на используемом языке реализации. Нет ни возможности, ни необходимости создавать исчерпывающий набор диаграмм, которые определили бы все возможные виды связей между нашими абстракциями. Нужно сосредоточиться лишь на самых важных отношениях, причем подразумевается, что в число важных входят те связи между абстракциями, которые отражают фундаментальные архитектурные решения или выражают детали, необходимые для реализации.

4. Реализация классов и объектов.

*Цель:* На этапе анализа реализация классов и объектов нужна, чтобы довести существующие абстракции до уровня, достаточного для обнаружения новых классов и объектов на следующем уровне абстракции; они сами будут в дальнейшем поданы на новую итерацию микропроцесса. Этот шаг намеренно выполняется позже всех, так как микропроцесс концентрирует внимание на поведении и откладывает насколько возможно решения о представлении. Такая стратегия оберегает разработчика от недозрелых решений, которые могут не оставить шансов на облегчение и упрощение архитектуры, и оставляет свободу выбора реализации (например, из соображений эффективности), гарантируя сохранение существующей архитектуры.

*Результаты:* Чтобы раскрыть связи между логическим и физическим в нашей реализации системы, вводятся диаграммы модулей, которые можно затем использовать, чтобы наглядно показать отображение архитектуры в ее программную реализацию. Далее можно применить специфические инструментальные средства (например CASE-средства), которые позволяют либо генерировать код из диаграмм, либо восстанавливать диаграммы по реализации. В этот шаг входит и обновление словаря данных, включая новые классы и объекты, которые были выявлены или изобретены при реализации существующих абстракций. Эти новые абстракции являются частью исходной информации для следующего цикла микропроцесса.

Контролирующим по отношению к микропроцессу является макропроцесс. Макропроцесс предписывает ряд измеримых результатов и действий, которые позволяют команде разработчиков оценить риск, внести заблаговременные изменения в микропроцесс и сосредоточиться на коллективном анализе и проектировании. Макропроцесс - это деятельность всего коллектива в масштабе от недель до месяцев.

**ЗАКЛЮЧЕНИЕ**

Автоматизированная система управления или АСУ входит в состав основных элементов автоматизированного производства компании. Основная задача АСУ — это управления всеми составными частями производства, то есть управление основным используемым при обработке оборудованием, а также дополнительным (к вспомогательному, но не менее важному оборудованию ГПС можно отнести различное технологическое оснащение, необходимое для выполнения определенной операции технологического процесса обработки детали, промышленных роботов, роботов транспортеров и т.д.). Кстати, «технологическим процессом» называется часть «производственного процесса» (производственный процесс начинается с обработки заготовки и заканчивается сборкой деталей в узлы) содержащая действия (совокупность операций и переходов.

Можно сказать, что объектно-ориентированный метод проектирования со времени своего появления и до настоящего момента стал основным и, с уверенностью можно утверждать, лучшим средством в разработке программного обеспечения информационных систем. Этот метод не дает универсальных рецептов, однозначно ведущих к успеху в разработке ПО. Можно сказать, что, "не существует рецептов, которые могли бы заменить ум, опыт и хороший вкус в проектировании и программировании... Различные фазы программного проекта, такие, как проектирование, программирование и тестирование, неотделимы друг от друга". Однако, несмотря на это, процесс объектно-ориентированного анализа и проектирования определен достаточно хорошо, чтобы быть предсказуемым и воспроизводимым в умелых руках.

Необходимо подчеркнуть, что каждый проект уникален, и, следовательно, разработчик, если он придерживается объектно-ориентированной методикой разработки, сам должен поддерживать баланс между неформальностью микропроцесса и формальностью макропроцесса, описанных выше.

Следует помнить, что средства программного обеспечения и аппаратурные средства являются двумя взаимосвязанными компонентами современной вычислительной техники, и с развитиями программного обеспечения необходимо отдавать должное аппаратным средствам производства.

Так же целесообразность автоматизации производства заключается в основном направлении автоматизации производственных процессов является именно внедрение гибких производственных систем, то есть АСУ. На основе применения принципов групповой технологии, использования перепрограммируемого технологического оборудования и программного управления разрешают в определенной степени противоречия, возникающие между единичным характером изделий, то есть максимальной изменчивостью производства, и необходимостью массового применения однотипных операций для обеспечения минимальных экономических затрат на выпуск продукции, что является экономически выгодным.

**СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ**

1. Белогорцев Е.В. «Автоматизированные системы управления», 2014г.-218с.
2. Медведев В. П. Вороненко, В. Н. Брюханов «Технологические основы гибких производственных систем», 2018г.-321с.
3. Осьмаков А.А. «Технология и оборудование производства электрических машин», 2013г. – 125с.
4. Зайцев С.Л. «Автоматизированные системы управления», 2017г.-365с.
5. Поспелов Д.А. Ситуационное управление. Теория и практика. М.: Наука, 1986, 285 с.
6. Искусственный интеллект. - В 3-х кн. Справочник. М.: Радио и связь, 2015.
7. MES Explained: A High Level Vision for Executives. White Paper of MESA International (2017). Интернет-адрес: [http://www.mesa.org/html/main. cgi? sub=7](http://www.mesa.org/html/main.cgi?sub=7)
8. Technical Articles of Consilium Company. Интернет-адрес: [http://www.consilium.com/Publications/technica. htm](http://www.consilium.com/Publications/technica.htm)
9. Уоссермен Ф. Нейрокомпьютерная техника. Теория и практика. М.: Мир, 2012. -238 с.
10. Гайсарян С.С. Объектно-ориентированные технологии проектирования прикладных программных систем. Интернет-адрес: [http://www.citforum.ru/programming/oop\_rsis/index. shtml](http://www.citforum.ru/programming/oop_rsis/index.shtml)
11. P. Coad, M. Mayfield. Object Models: Strategies, Patterns and Applications (Yourdon Press Computing Series). 2016.
12. Буч.Г. "Объектно-ориентированный анализ и проектирование с примерами приложений на C++", М.,"Бином", 2018 г.