

# Учебник советского военного программиста

Крыжановский С.В.

СамСверсталИздат

28 августа 2018 г.

Аннотация

Статья является ознакомительной публикацией редкого печатного источника по подготовке низового руководящего состава программистов в ВС СССР конца 1970-х годов. Читателю предлагаются выдержки из текста, не утратившие актуальность спустя сорок лет от момента издания.

## I. Введение

Под шквалом разнообразных лидерских курсов может сформироваться впечатление, что в былые времена разработчики представляли собою участников броуновского движения, ни в малейшей степени не заботясь ни об организации собственного труда, ни о подготовке свежих кадров, ни о формировании корпуса руководителей.

С экранов всемирной системы видеовещания YouTube и трибун конференций в слушателя вливают тривиальные или давно известные утверждения так, словно они являются недавними открытиями, способными развернуть отрасль в новые, неизведанные и прекрасные дали. Особую опасность несут адепты так называемых Софт Скиллс (soft skills — англ. мягкие, нетвёрдые умения), уже самоназвание которых должно отвратить сознательного человека, но отвращает, к сожалению, не всех.

Потому книга, которую я почти случайно обнаружил при поиске материалов для цикла статей, потрясла уместностью и актуальностью изложенных методик.

Небольшой томик 1978 года издания предназначался для обучения руководителей ниж-

него звена, которых сейчас на американский манер принято называть тимлидами (team lead — англ. вожак, руководитель команды). Прошло сорок лет с момента, как текст вышел в свет, а под слоем несколько архаичного и непривычного языка буднично предлагается то же, что с таким запалом проповедают нам евангелисты Софт Скиллс!

Ценность этого текста кажется настолько высокой, что в сторону были отложены другие статьи и эссе, и я направил все силы на публикацию находки.

Почему же такая замечательная книга была забыта поколениями программистов? Почему её отсканированных страниц не найти в свободном доступе и почему её названия нет в библиографическом разделе учебников МГУ, МГТУ и МИФИ? Я не знаю, могу лишь предполагать.

Возможно, свою роль сыграло стремление некоторых известных слоёв населения скорее забыть опыт СССР, не взирая на количество и качество этого опыта.

Возможно, офицеры некоторых профильных учреждений в составе Министерства обороны СССР (а книга издана именно им) в начале 90-х по ошибке уничтожили вместе с архивами и тираж книги, которую вовсе не

стоило уничтожать.

Возможно, и без того небольшое количество было прорежено руками нерадивых букинистов, безжалостно выкидывающих на свалку всё, что не принесёт и рубля в ближайшей лоточной перспективе.

Как уже сказал, можно лишь предполагать, судьба книг порою сложнее человеческой судьбы, сложности не обошли стороной и найденный мною экземпляр.

## II. О тексте

К 1970-м годам Советский Союз в полной мере осознал нужду в вычислительной мощи, а также в квалифицированных специалистах. Все виды вооружённых сил активно насыщались новым оборудованием, достаточно упомянуть унифицированные ЭВМ «Карат», ЭВМ «Радон», СЦВМ «Атака».

Выпускалась специальная литература, включая широко известную и донныне, например, «Мифический человеко-месяц» Брукса<sup>1</sup>.

Естественно, специфика военной жизни и работы накладывали на личность военных операторов ЭВМ и программистов свои требования к знаниям, умениям, характеру, методам обучения. Гражданские методики адаптировались, оформлялись и выпускались под грифом «Для служебного пользования». Такие издания были призваны заполнить лакуну не только в подготовке рядовых специалистов, но также и руководителей, требования к которым были выше.

По имеющемуся экземпляру книги невозможно с уверенностью сказать, был ли он засекречен. Во всяком случае, ничего секретного или нового текст не содержит. Титульный лист отсутствует, читательских помет чрезвычайно мало, часть страниц отсутствует. В целом физическое состояние хуже удо-

влетворительного, что затрудняет и делает бесполезной попытку сканирования или фотопроизводства.

Авторство установить не удалось, текст максимально обезличен, а по плотности цитирования Ленина и Фрунзе определять руку цитирующего наука пока не научилась.

Впрочем, косвенные признаки (структура текста, характерные обороты, специальная терминология, вектор подготовки) позволяют смело утверждать, что книга издана в 1978 году безальтернативно «Ордена Трудового Красного Знамени Военным издательством Министерства обороны СССР» и должна была служить настольной книгой руководителям небольших бригад (вероятно, потрёпанность экземпляра свидетельствует об усердном читателе).

При публикации ради облегчения восприятия мною допущен ряд вольностей. В оправдание замечу: публикация не научная, но ознакомительная, что снижает планку требований к строгости.

Цитируются только актуальные отрывки. Не публикуется текст, относящийся к партийной деятельности программиста, к действиям на поле боя, к внутреннему распорядку службы.

Нумерация глав нарушена, что не влияет на восприятие текста. Оригинальные названия оставлены.

После каждой цитаты в квадратных скобках номер страницы оригинала.

За редким исключением термины «руководитель подразделения программистов», «руководитель бригады программистов» (не той бригады, что военная) и прочие варианты заменены на «руководитель» без потери смысла.

Исправлены опечатки, ёфикация не проведена.

<sup>1</sup>Брукс Ф.П. Как проектируются и создаются программные комплексы. Мифический человеко-месяц. Очерки по системному программированию. Наука, 1979.

### III. Текст

#### i. Роль и задачи руководителей в обучении и воспитании советских программистов

Руководители подразделений программистов относятся к младшему звену управленческих кадров и представляют собой самый многочисленный их отряд. Они являются помощниками старших руководителей в подготовке идейно убежденных, умелых и мужественных защитников социалистической Родины. Это — ближайшие и непосредственные начальники разработчиков, первые их учителя и воспитатели. [8]

В настоящее время роль младшего звена управленческих кадров еще больше возросла. Это объясняется усложнением задач по обучению и воспитанию разработчиков в связи с сокращением сроков программистской службы и изменениями, вызванными революцией в программистском деле. [8-9]

Требования к обучению и воспитанию каждого разработчика неизмеримо возросли, а сроки их подготовки сократились. Это вызвало необходимость повышения уровня подготовки и ответственности руководителей всех степеней и особенно тех, кто ежедневно непосредственно готовит разработчиков, упорно трудится над тем, чтобы они стали настоящими программистами. [9]

Часть становится готовой к разработке, если хорошо подготовлены и сколочены отделения, экипажи, расчеты и группы, если каждый разработчик безупречно выполняет свои обязанности. [9]

Руководители непосредственно обучают и воспитывают подчиненных, помогают им изучать инструменты и ЭВМ, справочники и руководства, организуют и вместе с разработчиками несут программистские дежурства, ведут ежедневную работу по укреплению программистской дисциплины, твердого производственного порядка и организованности в коллективе. [9]

Огромное значение для победы в современной разработке имеют такие качества, как дисциплинированность, стойкость, смелость и мужество. Эти качества должны воспитываться у разработчиков повседневно, на всех занятиях и учениях, в ходе всей службы. [10]

Особое значение в настоящее время приобретает овладение программистским мастерством, умелое владение инструментами и ЭВМ в разработке. [10]

Современная разработка ведется с применением разнообразных и мощных средств производства программного обеспечения, которые личный состав не только должен хорошо знать, но и уметь с высокой эффективностью применять в самых сложных условиях обстановки, без этого немислима победа в современной разработке. [10]

Быть руководителем — не только большая честь, но и большая ответственность. Чтобы полностью выполнить свою роль, руководители должны иметь соответствующие политические, программистские и специальные знания, уметь передавать их подчиненным, иметь такие качества, как высокий моральный облик и личную примерность. [10]

Руководитель не может решать задачи обучения разработчиков, если сам не будет настоящим мастером программистского дела, специалистом высокой квалификации. Он должен обладать высокими командирскими и методическими навыками. [10]

Говоря о командирских качествах руководителя, следует иметь в виду умение уверенно командовать подчиненными, правильно решать задачи обучения и воспитания и проявлять постоянную требовательность к разработчикам. Высокая требовательность — неотъемлемое качество командира и необходимое условие высокой программистской дисциплины, организованности и порядка в коллективе. [10-11]

Кроме того, руководитель должен быть умелым педагогом и воспитателем. Без этого нельзя обеспечить единство процесса обучения и воопитания разработчиков, направлен-

ного на подготовку их к производству программного обеспечения. [11]

Все это означает, что руководитель не только должен уметь распоряжаться, командовать, но и тщательно готовить и образцово проводить занятия с подчиненными, уметь разъяснять им изученный материал, знать и использовать методические приемы, позволяющие повышать программистское мастерство личного состава, владеть методами воспитательной работы. А для этого необходимо быть хорошо ознакомленным с основами программистской педагогики и психологии, методикой разработческой подготовки. [11]

Самым действенным средством воспитания уважения к дисциплине является собственный пример воспитателя, на которого ежедневно и ежечасно устремлены взгляды подчиненных. Всякие разъяснения и требования, не подкрепленные личным примером, как правило, малодейственны и не достигают цели. [11]

Ежедневное поведение руководителя, его исполнительность и внешний вид оказывают непосредственное влияние на воспитание подчиненных. [11]

## ii. Сущность и основные принципы обучения и воспитания советских программистов

Всякое обучение начинается обычно с получения знаний и имеет конечной целью научить умело что-либо делать. Так, например, чтобы уметь с наибольшей эффективностью применять инструменты и ЭВМ в разработке, разработчик должен хорошо знать их устройство, назначение, тактико-технические характеристики, правила эксплуатации и способы программистского применения. [13]

Обучение программистскому делу отличается особой практической направленностью. Для успеха в разработке мало знать, как надо действовать, необходимо прежде всего уметь использовать эти знания на практике или иметь соответствующие навыки и умения. [13]

Умение как конечный результат процесса обучения заключается в способности правильно использовать знания для решения практических задач в различных условиях обстановки. [14]

Умение не приходит к обучаемым в результате только одного объяснения или показа. Для его приобретения необходимо часто применять знания на практике в ходе практических занятий, упражнений и тренировок. [14]

В отличие от навыка умение проявляется сознательно и должно осуществляться в разнообразных условиях обстановки, исключая шаблон и натаскивание (в ходе практических занятий и учений, уроков отказоустойчивости, программистских упражнений и других полевых занятий). [14]

Умственные навыки и умения обеспечивают быстрое и точное решение различных программистских задач на основе работы мышления (уяснение задачи, оценка обстановки и принятие решения, оценка важности целей и т.д.). [15]

Обогащение знаниями и навыками способствует приобретению умений. В свою очередь прочные навыки и умения создают условия для приобретения более глубоких знаний. Но в армейских условиях нередко бывает и так, что вначале личный состав приобретает необходимые навыки и умения, а в последующем осваивает теорию вопроса. Это делается в интересах поддержания высокой производственной готовности разработчиков. [16]

Революция в программистском деле и характер современной разработки существенно повлияли на содержание обучения личного состава. От него сейчас как никогда раньше требуется в короткие сроки освоить большой объем знаний, а главное, приобрести значительное количество сложных навыков и умений, особенно связанных со способами наиболее эффективного применения инструментов и ЭВМ в разработке. [16]

Неумелые действия хотя бы одного из членов коллектива могут свести на нет усилия

всех его членов в целом, а в отдельных случаях иметь нежелательные последствия. [16]

Воспитание коллективизма и товарищеской взаимопомощи, добросовестного отношения к делу и личной ответственности за порученный участок работы приобретает здесь первостепенное значение. [17]

Знания, навыки и умения при длительных перерывах в учебе могут незаметно ослабевать или утрачиваться. В связи с этим для поддержания профессионального мастерства личного состава на должном уровне необходимы систематические упражнения и тренировки, проводимые в сложных условиях обстановки. [17]

Принципами обучения советских программистов являются: партийность и научность; учить тому, что необходимо в разработке; сознательность и активность; наглядность; систематичность и последовательность; доступность; прочность знаний, навыков и умений; коллективизм и индивидуальный подход в обучении. [17]

Характер современной разработки предъявляет также повышенные требования к психологической закалке личного состава, особенно к его эмоционально-волевой устойчивости — способности противостоять опасности и отрицательным воздействиям на психику условий производства программного обеспечения. [19]

Успех обучения во многом зависит от того, насколько сознательно и активно усваивает материал обучаемый. [20]

В разработке разработчик должен проявлять инициативу и творчество, без чего невозможен успех в современном программировании. Опыт показывает, что только тот может инициативно и творчески применять знания на практике, кто их усвоил сознательно и глубоко осмыслил. [20]

Сознательность обучаемых проявляется в отношении к изученному материалу, приобретенным умениям и навыкам. Если это отношение положительное, обучаемый активно трудится на занятиях, т.е. у него широко

включены в процесс усвоения органы чувств и мышление. [20]

Активность всегда связана с какой-либо потребностью человека, с мотивами его деятельности. [20]

Но в процессе усвоения учебного материала часто приходится считаться и с рядом частных моментов, таких, например, как особенности психического склада отдельных разработчиков, степень развитости у них внимания, состояние психики в данный момент, содержание учебного материала на данном занятии, его сложность и т.п. [21]

Создание на занятиях обстановки активности и творчества достигается также постановкой перед обучаемыми вопросов, заставляющих их глубоко осмысливать и анализировать изучаемый материал, увеличением объема действий по решению вводных и задач, выполнением разработчиками самостоятельных заданий в сложной и быстроменяющейся обстановке. [21]

Любой учебный материал строится так, что каждая тема или учебный вопрос, прием или действие состоят из взаимосвязанных составных частей. Располагаясь в логическом порядке, они составляют то целое, что входит составной частью в общую сумму приобретенных разработчиками знаний, навыков и умений. [22-23]

Успех обучения поэтому возможен только при систематическом и последовательном изложении учебного материала. Это означает, что нужно так располагать и излагать учебный материал обучаемым, чтобы новые знания у них опирались на ранее приобретенные, изученные ранее приемы и действия подготавливали бы обучаемых к усвоению последующих. [23]

Другими словами, должны усваиваться не отдельные, не связанные между собой знания, навыки и умения, а система их, при этом, не усвоив предыдущего, нельзя приступать к последующему. [23]

Учебный материал усваивается хорошо, если он по своему содержанию и объему досту-

пен обучаемым с учетом их опыта и знаний. Доступность в обучении не означает упрощенчество, стремление уйти от сложных вопросов и трудностей в программистской подготовке. Разработчики обязаны знать программный материал независимо от его сложности и объема, приобрести необходимые для программирования навыки и умения. [23]

Прочность усвоения учебного материала во многом зависит также от отношения к нему самих обучаемых. Если разработчики будут знать, какое значение для их дальнейшей службы и деятельности имеют те или иные сведения, они запомнят их более полно и на длительное время. Поэтому в ходе занятий всегда надо подчеркивать практическое значение приобретаемых разработчиками знаний, навыков и умений. [24]

Разработчики большую часть времени обучаются в составе программистского коллектива, в котором обычно собраны люди с различным складом характера, разными привычками и способностями. В ходе взаимодействия их в коллективе и под влиянием различных факторов (влияние командира, всей системы воспитательной работы, партийной и комсомольской организации) формируется психология программистского коллектива (общественное мнение, настроенные привычки, взаимоотношения и характер реакции на события в коллективе и вне его). Положительно направленная психология коллектива способствует росту активности всех его членов, благоприятно сказывается на качестве программистской подготовки. [24-25]

Руководители обязаны всемерно способствовать развитию сплоченности коллектива на здоровой основе, чувства ответственности его за качество учебы каждого разработчика, готовности к взаимопомощи и взаимозаменяемости. [25]

Руководители должны понимать психологию коллектива и учитывать ее при организации подготовки его путем четких и слаженных действий всего личного состава и решения коллективных задач на фоне соревновательности, обсуждения лучших решений и

действий и ежедневного подведения итогов коллективных действий. [25]

Наряду с этим важно знать и учитывать индивидуальные особенности каждого обучаемого и создавать условия для проявления ими своих способностей в интересах подготовки высококвалифицированных специалистов, без которых не может быть слаженного коллектива. Руководители обязаны знать причины отставания отдельных разработчиков в учебе и оказывать им необходимую помощь различными приемами и средствами обучения. [25]

Нельзя серьезно говорить об участии в совершенствовании учебного процесса, если не изучать и не знать педагогические и психологические основы обучения разработчиков: основные принципы, формы и методы обучения, пути формирования навыков, психологические процессы человека в условиях программистской деятельности и психологию личности советского программиста, а также основные способы психологической закалки разработчиков в ходе программистской подготовки. [25]

Однако труд является лишь объективной предпосылкой формирования личности. Труд сам по себе не воспитывает. Интересно в этой связи замечание А.С. Макаренко: «Вы можете заставить человека трудиться сколько угодно, но если одновременно с этим вы не будете его воспитывать политически и нравственно, если он не будет участвовать в общественной и политической жизни, то этот труд будет просто нейтральным процессом, не дающим положительного результата». [31]

Задачи, которые ставятся перед личным составом, должны быть посильными и материально обеспеченными. Нереальные требования к программисту наносят большой ущерб воспитанию. Они подрывают веру в безоговорочный характер приказаний, отдаваемых командиром, и оказывают отрицательное влияние на формирование уверенности в собственных силах, что может замедлить овладение личным составом программистским мастерством. [31]

Постепенное наращивание и усложнение задач раскрывает программистам перспективы их роста, вырабатывает исполнительность, дисциплинированность, настойчивость, веру в свои силы и способности. [31]

Успешная реализация командиром воспитательных возможностей программистского коллектива осуществляется на основе определенных требований. Важнейшие из них следующие: установление и строгое соблюдение в коллективе уставных взаимоотношений, опора на коммунистов и комсомольцев, обсуждение с программистами результатов учебы и службы, организация взаимопомощи, борьба с ложным товариществом, развитие принципиальной критики и самокритики, накопление положительных традиций в коллективе. [32]

Руководитель должен быть старшим товарищем, требовательным, но тактичным и заботливым другом каждого подчиненного. Важно, чтобы он дорожил честью коллектива и стремился постоянно вести его вперед. [32]

Практика показывает, что изучение программиста наиболее целесообразно вести по таким вопросам: жизненный опыт (семья, школа, работа); особенности физического развития и состояние здоровья; направленность и мотивы поведения программиста; нравственные особенности его личности, темперамент и эмоциональные особенности (их сила, подвижность и уравновешенность); характер и воля (отношение к окружающему миру, к программистскому труду, к товарищам, к самому себе); способности и интеллектуальные особенности. [32-33]

Для высокой требовательности руководителя должны быть характерны такие черты, как принципиальность и непримиримость к недостаткам, настойчивость и решительность в проведении в жизнь своих требований, установление персональной ответственности разработчиков за порученное дело, строгий контроль исполнения. [34]

Необходимы также естественность, просто-

та обращения, не допускающая фамильярности и панибратства; принципиальность и настойчивость без упрямства; внимательность и чуткость к подчиненным без подчеркивания этого; юмор и ирония без насмешливости, унижающей достоинство человека; обучение без подчеркивания своего превосходства в знаниях, навыках и умениях; умение слушать подчиненных, не выражая безразличия и превосходства; серьезное отношение к ответу подчиненного независимо от его правильности и грамотности; сочетание требовательности к подчиненным с требовательностью к себе. [34]

### iii. Основные методы и формы обучения и воспитания советских программистов

Беседа практикуется в целях углубления, закрепления или проверки знаний и носит обычно вопросо-ответный характер. Она всегда предполагает свободный обмен мнениями, однако руководитель должен ее направлять в необходимом русле, добиваясь полного раскрытия существа темы и поставленных перед беседой учебных целей. [36]

В ходе беседы всегда надо стремиться вовлечь в нее всех разработчиков, помогая некоторым из них преодолеть робость и нерешительность. При недостаточном раскрытии вопроса одним обучаемым другие дополняют его. Каждый ответ должен быть разобран, подчеркнуты его сильные и слабые стороны, правильное и ошибочное в суждениях разработчиков. [36]

К практической работе приступают после того, как обучаемые усвоят минимум необходимых знаний и приобретут определенные навыки и умения. [38]

Самостоятельная работа является важнейшим методом обучения разработчиков и в современных условиях приобретает все большее значение. Метод самостоятельной работы служит как для закрепления уже приобретенных знаний, навыков и умений, так и для

приобретения новых. [38]

Самостоятельная работа заключается в изучении программистской литературы, и прежде всего руководств, инструментов и ЭВМ, в самостоятельной тренировке в выполнении необходимых приемов и действий, а также в прослушивании теле- и радиопередач по соответствующей профессии тематике. [38]

Самоподготовка как одна из форм обучения с каждым годом совершенствуется и пополняется новым содержанием, особенно в учебных подразделениях. Это связано как с ростом общеобразовательной подготовки личного состава, так и с увеличением объема знаний, необходимых ему для овладения программистским мастерством. [41]

Основной метод на самоподготовке — самостоятельная работа обучающихся, которая должна направляться и контролироваться руководителями. [41]

Выставляя оценку, руководитель должен ее обосновать, указать на сильные и слабые стороны в подготовке разработчика и потребовать от отстающих учиться с полным напряжением сил. [41]

Убеждение — это активное воздействие на ум, чувства и волю разработчика, имеющее своей целью помочь человеку осмыслить, понять суть идей и требований и в соответствии с ними решать практические задачи. Командир убеждает своих подчиненных словом, делом и примером. [44]

Руководителю необходимо иметь в виду, что убедить разработчика можно лишь тогда, когда сам глубоко убежден в тех идеях и мыслях, которые несешь ему. [44]

При этом программистам необходимо говорить правду, не скрывать от них трудности, с которыми мы встречаемся, раскрывать их причины и пути преодоления. При разговоре с подчиненными нельзя горячиться. [45]

Важно учитывать индивидуальные особенности разработчика, помнить, что он от своих взглядов и убеждений, если даже они ошибочные, отказывается не сразу. Руководитель

обязан терпеливо опровергать неправильные взгляды подчиненного, давать время все продумать, оценить, устранить возникающие сомнения. [45]

От чего же возрастает воспитательная сила поощрений? Если послушать мнение передовых руководителей — командиров отличных отделений, — то они считают, что поощрение должно быть обоснованным и целесообразным. Ничто так отрицательно не воздействует на подчиненных, как неоправданное поощрение, переоценка заслуг программиста. В этом руководители убеждались не раз, особенно тогда, когда поощряли за выполнение элементарных обязанностей, например, за хорошее включение ЭВМ, за содержание в чистоте клавиатуры и т.д. [46]

Суть строгого, требовательного воспитания сводится к следующему: требование должно быть твердым и решительным; четким и определенным; кратким и немногословным; требование должно быть справедливым, исполнимым, последовательным, настойчивым и выражаться в сдержанной форме. Руководитель не должен кого-либо оскорблять. [47]

Технике применения метода принуждения надо учиться. Здесь все имеет значение: и форма обращения к программисту, и тон, и поза, и многое другое. А самое главное — глубокая вера в необходимость требования и вытекающая отсюда настойчивость, воля, решимость. [47]

Личный пример руководителя как специфический метод воспитания является его главным оружием, важнейшим условием успешного выполнения своих обязанностей и влияния на людей. [47]

Положительный пример руководителя приобретает силу воспитательного влияния лишь в том случае, если он органически связан с высокими качествами его личности и проявляется не от случая к случаю, а постоянно. [47]



#### iv. Основы производства программного обеспечения подразделениями

Современная разработка требует от личного состава большого напряжения моральных и физических сил, мужества, смелости, программистской хитрости, стойкости и непреклонной воли к победе. Поэтому успех в разработке в первую очередь зависит от морально-политического состояния личного состава подразделений, от его программистской выучки, физической и психологической закалки. Кроме того, успех разработки зависит от количества и качества инструментов, вычислительной и другой техники, а также умения использовать их в разработке. [58]

Действия личного состава в разработке направляются единой волей командира-единоначальника. Он всесторонне анализирует обстановку и организует разработку, предвидя возможные варианты ее развития. Поэтому успех разработки во многом зависит от умения командира организовать разработку и подготовить личный состав, инструменты, вычислительную и другую технику к разработке. [58]

Инициатива заключается в стремлении в любой обстановке найти наилучшие способы выполнения поставленной задачи, в принятии смелых решений и твердом проведении их в жизнь, не ожидая указаний. Инициативу в равной степени должен проявлять и командир, и подчиненные. В основе действий всех командиров должны быть постоянное стремление выполнить поставленную задачу и готовность взять на себя ответственность за смелое решение. [60]

#### v. Управление подразделениями в разработке

Сущность управления подразделениями заключается в постоянном руководстве подчиненными, в организации разработки и направлении их усилий на успешное выполнение поставленных задач. [70]

Твердость управления заключается в том,

что командир должен смело принимать решение, настойчиво и решительно проводить его в жизнь в целях точного и полного выполнения программистской задачи. [70]

Гибкость управления обеспечивается постоянным знанием обстановки и своевременным реагированием на все ее изменения, уточнением ранее принятого решения или принятием нового решения в соответствии с изменившейся обстановкой. [71]

Командиры подразделений управляют действиями подразделений лично. [73]

Представление донесений старшему руководителю и информация соседей об обстановке и ее изменениях составляют важнейшую обязанность руководителей в разработке. [75]

Донесения и информации могут быть письменными и устными. Главное требование, предъявляемое к ним, — достоверность, правильность, своевременность и ясность изложения. [75]

#### vi. Дополнительно

Одному разработчику достаточно сделать лишь замечание за тот или иной промах или проступок, и он, надолго запомнив его, впредь не будет допускать оплошностей по работе. Другого приходится постоянно держать под неослабным контролем, систематически напоминать о требованиях руководств и справочников, а иногда даже прибегать и к наказаниям. [77-78]

Определяя меры воздействия на виновных, никогда не стоит горячиться, «рубить с плеча». Нужно сначала выяснить, преднамеренно или неосознанно совершен тот или иной проступок, побеседовать с разработчиком-нарушителем и лишь после этого принимать решение, как с ним поступить. [78]

Для того чтобы обучать подчиненных, руководитель сам должен быть хорошо подготовлен, в совершенстве знать и мастерски владеть тем инструментом, вычислительной и другой техникой, которые вручены подразделению. Кроме того, руководителю необходи-

мо уметь доходчиво передавать свои знания и навыки подчиненным, четко и без ошибок показывать им нужный прием или действие, выразительно излагать свои мысли, подмечать ошибки обучаемых, указывать на них и настойчиво добиваться их устранения. [78-79]

Высоким авторитетом у подчиненных пользуется лишь тот руководитель, который в любой обстановке, без послаблений и упрощений, с полным напряжением моральных и физических сил обучает подчиненных действовать так, как придется действовать в разработке. [79]

При этом руководителю необходимо глубоко усвоить, что каждый отдаваемый им приказ должен быть разумным, осмысленным и направленным на пользу производства. Прежде чем приказывать, следует взвесить возможности выполнения приказа, но приказав, сделать все необходимое, чтобы приказ был выполнен точно и в срок. [82]

Требовательность — непереносимое качество каждого руководителя. Она несовместима с проявлением безволия, панибратства, либерального отношения к отрицательным фактам в поведении отдельных разработчиков. Требовательность должна быть постоянной. Плохо поступает тот руководитель, который сегодня взыскивает с подчиненных со всей строгостью закона, а завтра не замечает у подчиненных упущений по работе и делает им поблажки. Такая непостоянная требовательность создает у подчиненных неправильное представление о сущности программистской службы. [82]

Главной задачей руководителей в процессе обучения является своевременное выявление у подчиненных недостатков и ошибок при выполнении приемов; вскрытие причин этих ошибок; умелое устранение в ходе каждого занятия и повседневной жизни всех недостатков и ошибок; постоянная высокая требовательность к себе и подчиненным. [120]

#### IV. Заключение

Внимательный читатель заметит, что этим кратким и явно вводным учебником затрагиваются разные стороны отношений руководителя и подчинённых.

Упомянуто всё: личный пример, уровни квалификации, обучение и самостоятельная учёба, поощрения и наказания, роль коллектива, психология и педагогика.

Иначе говоря, то, о чём без остановки говорят адепты Софт Скиллс, выдавая озвучиваемое за новейшие разработки западной мысли, давным-давно было известно нашим отцам и дедам.

А если вдуматься ещё глубже, чему-то подобному нас обучают с пелёнок. Будь умным. Будь настойчивым. Будь справедливым. Умей ладить с людьми и добиваться своего. Помогай другим стать лучше.

Чтобы ознакомиться с этими наставлениями, не надо обращаться к консультантам по карьере. Достаточно открыть любой учебник любого уровня руководителя в любой области в любую эпоху (пожалуй, руководства рабовладельческого строя всё-таки будут перебором) и под слоем бытовой архаики («не секи крестьянина своего, не поведав о причине») вы найдёте одни и те же истины.