

да; медленная, но правильная всегда имеет некоторую ценность, а может оказаться и вполне удовлетворительной» [6, с. 161]. Очевидно, вместо критерия скорости в приведенную выше «формулу эффективности по Майерсу» можно подставить в конкретной ситуации любой иной критерий из всего традиционного набора критериев эффективности, охраняемого большой наукой об ЭВМ.

### О критериях эффективности в программировании

В 50—60-х годах одной из основных характеристик сложности электронного оборудования могло служить число используемых в нем ламп, а затем полупроводниковых вентилях (диодов и транзисторов). Минимизация числа активных компонентов цифровых устройств, как правило, означала в то время существенное упрощение ЭВМ, повышение надежности. Математический аппарат, который позволял проектировщику «экономить мышление» и минимизировать количество используемых в конструируемом приборе вентилях формальными методами, был в этих условиях полезным и широко использовался.

После появления на рубеже 70-х годов интегральных схем средней, а затем и большой (БИС) интеграции старая характеристика сложности цифрового устройства по числу используемых в нем диодов и транзисторов потеряла всякий смысл. С другой стороны, если в 50—60-х годах никому и в голову не могло прийти оценивать конструктивную сложность цифрового устройства суммарным числом «ножек» на цоколях всех его ламп или общим числом выводов на диодах и транзисторах, то в 70-х годах оказалось, что общее число активных компонентов уже не имеет решающего значения, а в большинстве случаев более важными параметрами являются количество выводов и число кристаллов.

В это время уже трудно было бы отыскать разработчика цифровой аппаратуры, который из любви к классическим методам ранее освоенного им математического аппарата занимался бы минимизацией числа транзисторов в цифровой аппаратуре четвертого поколения.

Совершенно иная ситуация сложилась в технологии программирования. Критерии эффективности программ, сформировавшиеся в 50-х годах для ламповых ЭВМ, с трудом «вытравлялись» из науки и практики программирования середины 70-х годов и все еще сохраняют живучесть в середине 80-х. Э. Йодан, защищая в 1977 г. структурное