**1. Назначение и процессы управления разработкой программного средства**

***Управление разработкой ПС* (*software management*)**– это деятельность, направленная на обеспечение необходимых условий для работы коллектива разработчиков ПС, на планирование и контроль деятельности этого коллектива с целью обеспечения требуемого качества ПС, выполнения сроков и бюджета разработки ПС [1, 2]. Часто эту деятельность называют также ***управлением программным проектом*(*software project management*)**. Здесь под *программным проектом*(*software project*) понимают всю совокупность работ, связанную с разработкой ПС, а ход выполнения этих работ называют *развитием программного проекта*(*software project progress*)*.*К необходимым условиям работы коллектива относятся помещения, аппаратно-программные средства разработки, документация и материально-финансовое обеспечение. Планирование и контроль предполагает разбиение всего процесса разработки ПС на отдельные конкретные работы (задания), подбор и расстановка исполнителей, установление сроков и порядка выполнения этих работ, оценка качества выполнения каждой работы. Финальной частью этой деятельности является организация и проведения аттестации (сертификации) ПС, которой завершается стадия разработки ПС. Вопросы документации обсуждались в предыдущей лекции. Другие вопросы управления разработки ПС кратко обсудим в настоящей лекции.

Хотя виды деятельности по управлению разработкой ПС могут быть весьма разнообразными в зависимости от специфики разрабатываемого ПС и организации работ по его созданию, можно выделить некоторые общие процессы (виды деятельности) по управлению разработкой ПС:

* составление плана-проспекта по разработке ПС;
* планирование и составление расписаний по разработке ПС;
* управление издержками по разработке ПС;
* текущий контроль и документирование деятельности коллектива по разработке ПС.
* подбор и оценка персонала коллектива разработчиков ПС.

***Составление плана-проспекта по разработке ПС*** включает формулирование предложений о том, как выполнять разработку ПС. Прежде всего, должно быть зафиксировано, для кого разрабатывается ПС:

* для внешнего заказчика,
* для других подразделений той же организации,
* или является инициативной внутренней разработкой.

В плане-проспекте должны быть установлены общие очертания работ по создания ПС и оценена стоимость разработки, а также предоставляемые для разработки ПС материально-финансовые ресурсы и временные ограничения. Кроме того, он должен включать обоснование, какого рода коллективом должно разрабатываться ПС (специальной организацией, отдельной бригадой и т.п.). И, наконец, должны быть сформулированы необходимые технологические требования (включая, возможно, и выбор подходящей технологии программирования).

***Планирование и составление расписаний по разработке ПС***– это деятельность, связанная с распределением работ между исполнителями и по времени их выполнения в рамках намеченных сроков и имеющихся ресурсов. Более подробно этот процесс будет рассмотрен в разделе 3.

***Управление издержками по разработке ПС***– это деятельность, направленная на обеспечение подходящей стоимости разработки в рамках выделенного бюджета. Она включает оценивание стоимости разработки проекта в целом или отдельных его частей, контроль выполнения бюджета, выбор подходящих вариантов расходования бюджета. Эта деятельность тесно связана с планированием и составлением расписаний в течение всего периода выполнения проекта. Основными источниками издержек являются:

* затраты на аппаратное оборудование (hardware);
* затраты на вербовку и обучение персонала;
* затраты на оплату труда разработчиков.

***Текущий контроль и документирование деятельности коллектива по разработке ПС***– это непрерывный процесс слежения за ходом развития проекта, сравнения действительных состояния и издержек с запланированными, а также документирования различных аспектов развития проекта. Этот процесс помогает вовремя обнаружить затруднения и предсказать возможные проблемы в развитии проекта.

***Подбор и оценка персонала коллектива разработчиков ПС***– это деятельность, связанная с формированием коллектива разработчиков ПС. Имеющийся в распоряжении штат разработчиков далеко не всегда будет подходящим по квалификации и опыту работы для данного проекта. Поэтому приходится, частично, вербовать подходящий персонал, а, частично, организовывать дополнительное обучение имеющихся разработчиков. В любом случае в формируемом коллективе хотя бы один его член должен иметь опыт разработки программных средств (систем), сопоставимых с ПС, который требуется разработать. Это поможет избежать многих простых ошибок в развитии проекта.

**2. Структура управления разработкой программных средств**

Разработка ПС обычно производится в организации, в которой одновременно могут вестись разработки ряда других программных средств. Для управления всеми этими программными проектами используется иерархическая структура управления [1]. Она представлена на рис. 1.

Во главе этой иерархии находится *директор*(или вице-президент) программистской организации, отвечающий за управление всеми разработками программных средств. Ему непосредственно подчинены несколько менеджеров сферы разработок и один менеджер по качеству программных средств. В результате общения с потенциальными заказчиками директор принимает решение о начале выполнения какого-либо программного проекта, поручая его одному из менеджеров сферы разработок, а также решение о прекращение того или иного проекта. Он участвует в обсуждении общих организационных требований (ограничений) к программному проекту и возникающих проблем, решение которых требует использование общих ресурсов программистской организации или изменения заказчиком общих требований.

*Менеджер сферы разработок* отвечает за управление разработками программных средств (систем) определенного типа, например, программные системы в сфере бизнеса, экспертные системы, программные инструменты и инструментальные системы, поддерживающие процессы разработки программных средств, и другие. Ему непосредственно подчинены менеджеры проектов, относящихся к его сфере. Получив поручение директора по выполнению некоторого проекта, он организует формирование коллектива исполнителей по этому проекту (в частности, необходимую вербовку и обучение персонала). Он участвует в обсуждении плана-проспекта программного проекта, относящегося к сфере разработок, за которую он отвечает, а также в обсуждении и решении возникающих проблем в развитии этого проекта. Он организует обобщение опыта разработок программных средств в его сфере и накопление программных средств и документов для повторного использования.

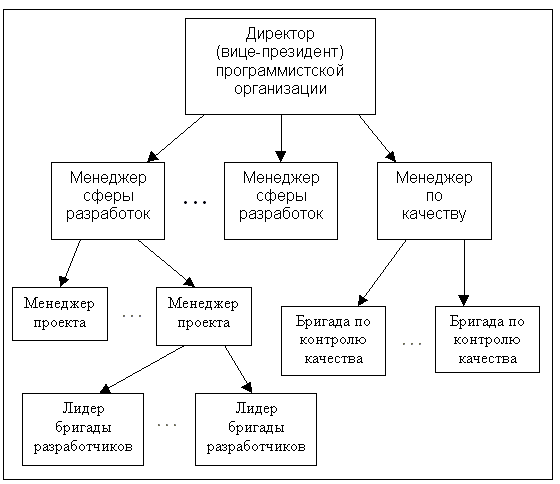


Рис. 1. Структура управления разработкой программных средств.

По каждому программному проекту назначается свой менеджер, который управляет развитием этого проекта. Ему непосредственно подчинены лидеры бригад разработчиков. *Менеджер проекта* осуществляет планирование и составление расписаний работы этих бригад по разработке соответствующего ПС (следующий раздел).

Считается крайне нецелесообразным разработка большого ПС (программной системы) одной большой единой бригадой разработчиков. Для этого имеется ряд серьезных причин. В частности, в большой бригаде время, затрачиваемое на общение между ее членами, может быть больше времени, затрачиваемого на собственно разработку. Отрицательное влияние оказывает большая бригада на строение ПС и на интерфейс между отдельными его частями. Все это приводит к снижению надежности ПС. Поэтому обычно большой проект разбивается на несколько относительно независимых подпроектов таким образом, чтобы каждый подпроект мог быть выполнен отдельной небольшой бригадой разработчиков (обычно считается, что в бригаде не должно быть больше 8-10 членов). При этом архитектура ПС должна быть такой, чтобы между программными подсистемами, разрабатываемыми независимыми бригадами, был достаточно простой и хорошо определенный системный интерфейс.

Используют три подхода к организации бригад разработчиков [1, 3, 4]:

* обычные бригады,
* неформальные демократические бригады,
* бригады ведущего программиста.

В *обычной бригаде*старший программист (*лидер бригады*) непосредственно руководит работой младших программистов. Недостатки такой организации непосредственно связаны со спецификой разработки ПС: программисты разрабатывают сильно связанные части программной подсистемы, сам процесс разработки состоит из многих этапов, каждый из которых требует особенных способностей от программиста, ошибки отдельного программиста могут препятствовать работе других программистов. Успех работы такой бригады достигается в том случае, когда ее руководитель является компетентным программистом, способным предъявлять к членам бригады разумные требования и умеющим поощрять хорошую работу.

В *неформальной демократической бригаде* поручаемая ей работа обсуждается совместно всеми ее членами, а задания между ее членами распределяются согласованно в зависимости от способностей и опыта этих членов. Один из членов этой бригады является *лидером (руководителем) бригады*, но он также выполняет и некоторые задания, распределяемые между членами бригады. Неформальные демократические бригады могут весьма успешно справляться с порученной им работой, если большинство членов бригады являются опытными и компетентными специалистами. Если же неформальная демократическаябригада состоит, в основном, из неопытных и некомпетентных членов, в деятельности бригады могут возникать большие трудности. Без наличия в бригаде хотя бы одного квалифицированного и авторитетного члена, способного координировать и направлять работу членов бригады, эти трудности могут привести к неудаче проекта.

В *бригаде ведущего программиста* за разработку порученной программной подсистемы несет полную ответственность один человек, называемый *ведущим программистом (chief programmer)*и являющийся *лидером бригады*:он сам конструирует эту подсистему, составляет и отлаживает необходимые программы, пишет документацию к подсистеме. Ведущий программист выбирается из числа опытных и одаренных программистов. Все остальные члены такой бригады, в основном, создают условия для наиболее продуктивной работы ведущего программиста. Организацию такой бригады обычно сравнивают с хирургической бригадой [1, 4]. Ядро бригады ведущего программиста составляют три члена бригады: помимо ведущего программиста в него входит дублер ведущего программиста и администратор базы данных разработки. *Дублер ведущего программиста (backup programmer)* такжеявляется квалифицированным и опытным программистом, способным выполнить любую работу ведущего программиста, но сам он эту работу не делает. Главная его обязанность – быть в курсе всего, что делает ведущий программист. Он выступает в роли оппонента ведущего программиста при обсуждении его идей и предложений, но решения по всем обсуждаемым вопросам принимает единолично ведущий программист. *Администратор базы данных разработки (librarian)* отвечает за сопровождение всей документации (включая версии программ), возникающей в процессе разработки программной подсистемы, и снабжает членов бригады информацией о текущем состоянии разработки. Эта работа выполняется с помощью соответствующей инструментальной компьютерной поддержки. В зависимости от объема и характера порученной работы в бригаду могут быть включены дополнительные члены, такие как:

* распорядитель бригады, выполняющий административные функции;
* технический редактор, осуществляющий доработку и техническое редактирование документов, написанных ведущим программистом;
* инструментальщик, отвечающий за подбор и функционирование программных средств, поддерживающих разработку программной подсистемы;
* тестовик, готовящий подходящий набор тестов для отладки разрабатываемой программной подсистемы;
* один или несколько младших программистов, осуществляющих кодирование отдельных программных компонент по спецификациям, разработанным ведущим программистом.

Кроме того, к работе бригады может привлекаться для консультации эксперт по языку программированию.

Важное место в управлении разработкой ПС отводится управление обеспечением качества. Для руководства этой деятельностью назначается специальный менеджер, подчиненный непосредственно директору, – *менеджер по качеству*. Ему непосредственно подчинены формируемые бригады по контролю качества. Эти бригады работают с отдельными проектами, но непосредственно соответствующим менеджерам проектов не подчинены, сохраняя тем самым свою независимость от них.

Управление обеспечением качества означает контроль качества каждой работы, выполняемой разработчиками в рамках программного проекта, контроль каждого документа, включаемого в ПС. Качество ПС не может быть добавлено к ПС после того, как оно будет уже создано. Качество ПС формируется постепенно в процессе всей разработки ПС, в каждой отдельной работе, выполняемой по программному проекту. Поэтому для каждой такой работы, прежде чем она получит одобрение и будет считаться завершенной, организуется *смотр* (*review*) соответствующей бригадой по контролю качества. Этот смотр существенно отличается от контроля, осуществляемого разработчиками в конце каждого этапа разработки, так как последний является техническим процессом, связанным с обнаружением ошибок, тогда как смотр по контролю качества является функцией управления разработкой и связан с оценкой того, насколько результаты этой работы согласуются с декларированными требованиями относительно качества ПС.

Существенную роль в управлении качеством ПС играют программные (софтверные) стандарты [1, 2]. Они фиксируют удачный опыт высоко квалифицированных специалистов по разработке ПС для различных их классов и для разных моделей их качества. Следование подходящим стандартам может существенно облегчить достижение поставленных целей относительно качества ПС, а также упростить смотр по контролю качества. Кроме того, стандарты способствуют формированию взаимопонимания внутри коллектива разработчиков и упрощают процесс обучения новых членов этого коллектива.

Различают два вида таких стандартов:

* стандарты ПС (программного продукта),
* стандарты процесса создания и использования ПС.

*Стандарты ПС*определяют некоторые свойства, которыми должны обладать программы или документы ПС, т.е. определяют в какой-то степени качество ПС. При спецификации качества для конкретизации какого-либо примитива качества иногда достаточно указать, какому стандарту он должен соответствовать, в других случаях привязка примитива качества к стандарту может потребовать лишь незначительной дополнительной конкретизации этого примитива. Привязка примитивов качества к тем или иным стандартам сильно упрощает контроль и оценку качества ПС. К стандартам ПС относятся, прежде всего, стандарты на языки программирования, на состав документации, на структуру различных документов, на различные форматы и другие.

*Стандарты процесса создания и использования ПС* определяют, как должен проводится этот процесс, т.е. подход к разработке ПС, структуру жизненного цикла ПС и его технологические процессы. Хотя эти стандарты непосредственно не определяют качества ПС, однако считается, что качество ПС существенно зависит от качества процесса его разработки. Эти стандарты проще контролировать, поэтому повсеместно используются для управления качеством ПС.

Эти стандарты могут быть как международными или национальными, так и специально созданными для организации, в которой ведется разработка ПС. Разработка последних стандартов является одной из функций управления обеспечением качества ПС.

*Бригада по контролю качества* состоит из ассистентов (рецензентов) по качеству ПС. Она проводит смотры тех или иных частей ПС или всего ПС в целом с целью поиска возникающих проблем в процессе его разработки. Смотру подлежат все программные компоненты и документы, включаемые в ПС, а также процессы их разработки. В процессе смотра учитываются требования, сформулированные в спецификации качества ПС, в частности, проверяется соответствие исследуемого документа или технологического процесса стандартам, указанным в этой спецификации. В результате смотра формулируются замечания, которые могут фиксироваться письменно или просто передаваться разработчикам устно.

Для смотра каждой конкретной программной компоненты или документа ПС создается комиссия (группа) во главе с *председателем* (*chairman*), который отвечает за организацию смотра. Он должен иметь достаточный опыт конструирования ПС, чтобы быть готовым принять ответственность за важные технические решения. В эту комиссию включаются два или три ассистента по качеству ПС, один из которых должен быть ответственным за запись решений, сделанных в течение смотра. К смотру обычно привлекаются разработчик исследуемой компоненты или исследуемого документа ПС, а также новые члены коллектива разработчиков в целях их обучения.

**3. Планирование и составление расписаний по разработке ПС**

Общее представление об этой деятельности можно составить по ее описанию на псевдокоде, приведенном на рис. 2 [1]. Это описание показывает, что планирование и составление расписаний по разработке ПС представляет собой итеративный процесс, который заканчивается только после прекращения работ по самому программному проекту.

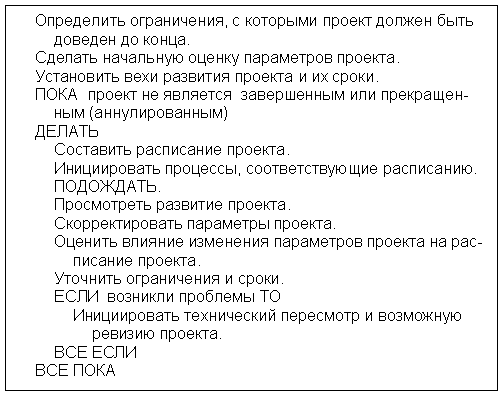


Рис. 14.2. Описание на псевдокоде процесса планирования и составления расписаний по разработке ПС.

В начале этого описания оцениваются общий срок разработки ПС, используемые штаты исполнителей, предельный бюджет и другие ограничения (условия) разработки. С учетом этого фиксируются начальные параметры проекта (его структура и распределение функций). Должны быть также определены “вехи развития проекта” и их сроки. *Веха развития проекта* (*project* *progress milestone*) – это конечная точка некоторого этапа или процесса, с которой связывается выдача некоторого промежуточного продукта, представляющего собой некоторый четко определенный документ. Вехи развития проекта обеспечивают возможность контроля развития проекта и возможность модификации расписаний проекта.

Далее начинается итерационный процесс, основу которого составляет повторяющиеся составления расписаний. Составление расписания заключается

* в разделении всей работы, необходимой для выполнения проекта, на отдельные самостоятельно выполняемые задания;
* в составлении сетевого графика выполнения заданий;
* в составлении гистограммы выполнения заданий;
* в расстановке исполнителей заданий.

При выделении самостоятельных заданий для каждого из них оценивается время его выполнения и его зависимость от других заданий с точки зрения порядка выполнения. С*етевой график*представляет собой схему (сеть) путей выполнения заданий с указанием времени выполнения каждого задания и с расстановкой вех развития проекта. В сетевом графике должен быть определен *критический путь*, представляющий собой такой путь заданий, суммарное время выполнения которых является наибольшим. *Гистограмма выполнения заданий* (*activity bar chart*) содержит для каждого задания свою временнэю полосу от момента, когда выполнение этого задания может быть начато, и до момента, когда выполнение этого задания должно быть закончено. В такой полосе фиксируется как продолжительность выполнения самого задания, так и возможный запас времени для завершения его выполнения. Это дает возможность модифицировать план развития проекта в определенных рамках без изменения общих сроков выполнения проекта. При расстановке исполнителей оценивается для каждого исполнителя соответствие его квалификации и опыта характеру предлагаемой работы. Особое внимание уделяется расстановке исполнителей заданий, находящихся на критическом пути.

Спустя некоторое время (обычно 2-3 недели) после активизации процессов, указанных в расписании, производится обозрение (просмотр) хода развития проекта и отмечаются возникшие противоречия. С учетом этого производится пересмотр (уточнение) параметров проекта и оценивается влияние измененных параметров на расписание проекта. Если окажется, что эти изменения увеличивают время разработки ПС, необходимо обсудить с заказчиком возможность изменения ограничений проекта и срока его завершения. В том случае, когда заказчик не может пойти на подходящие изменения, производится технический пересмотр проекта с целью поиска альтернативных подходов к разработке ПС.

**4. Аттестации программного средства**

Завершающим этапом разработки ПС является аттестация ПС, подводящая итог всей разработке. *Аттестация* (*certification*) ПС - это авторитетное подтверждение качества ПС [5, 6]. Обычно для аттестации ПС создается аттестационная комиссия из экспертов, представителей заказчика и представителей разработчика. Эта комиссия проводит *приемо-сдаточные* *испытания*ПС с целью получения необходимой информации для оценки его качества. Под *испытанием*ПС здесь понимают [6, 7] процесс проведения комплекса мероприятий, исследующих пригодность ПС для успешной его эксплуатации (применения и сопровождения) в соответствии с требованиями заказчика. В этом процессе проверяется полнота и исследуется качество представленной программной документации, производится необходимое тестирование программ, входящих в состав ПС, а также исследуются и другие свойства ПС, декларированные в его спецификации качества. На основе полученной информации комиссия должна установить, в какой степени ПС выполняет декларированные функции и в какой степени ПС обладает декларированными примитивами и критериями качества. Решение аттестационной комиссии о произведенной оценке качества ПС фиксируется в соответствующем документе (сертификате), который подписывается членами комиссии.

Таким образом, оценка качества ПС является основным содержанием процесса аттестации. Прежде всего, следует отметить, что оценка качества ПС производится по предъявленной спецификации его качества, т.е. оценивается только декларированное разработчиками качество ПС. При этом оценка качества ПС по каждому из критериев сводится к оценке каждого из примитивов качества, связанному с этим критерием качества ПС, в соответствии с их конкретизацией в спецификации качества этого ПС. Различают следующие группы методов оценки примитивов качества ПС:

* непосредственное измерение показателей примитива качества;
* тестирование программ ПС;
* экспертная оценка на основании изучения программ и документации ПС.

*Непосредственное измерение* показателей примитива качества производится путем проверки соответствия предъявленной документации (включая тексты программ на языке программирования) стандартам или явным требованиям, указанным в спецификации качества ПС, а также путем измерения времени работы различных устройств и используемых ресурсов при выполнении контрольных (тестовых) задач. Например, некоторым показателем эффективности по памяти может быть число строк программы на языке программирования, а некоторым показателем временной эффективности может быть время ответа на запрос пользователя.

Для оценки некоторых примитивов качества ПС используется *тестирование*[5-8]. К таким примитивам относятся, прежде всего, завершенность ПС, а также его точность, устойчивость, защищенность и другие примитивы качества. Однако, во время приемо-сдаточных испытаний нет необходимости проведения тестирование ПС в полном объеме (это может слишком дорого стоить). Аттестационная комиссия должна, прежде всего, изучить предъявленную документацию по проведенному разработчиками тестированию ПС и лишь выборочно пропустить предъявленные тесты. Дополнительные тесты составляются, если у комиссии возникают определенные сомнения в полноте тестирования, проведенного разработчиками. Кроме того, обычно работоспособность и некоторые показатели качества ПС демонстрируются на решении контрольных задач, предъявляемых заказчиком.

В некоторых случаях для оценки качества ПС проводятся дополнительные полевые или промышленные испытания [8, 9]. *Полевые* испытания ПС – это демонстрация ПС вместе с технической системой, которой управляет эта ПС, с обеспечением тщательного наблюдения за поведением ПС. Заказчикам должна быть предоставлена возможность задания собственных контрольных примеров, в частности, с выходом в критические режимы работы технической системы, а также с вызовом в ней аварийных ситуаций. *Промышленные* испытания ПС – это процесс передачи ПС в постоянную эксплуатацию пользователям, представляющий собой опытную эксплуатацию ПС пользователями со сбором информации об особенностях поведения ПС и ее эксплуатационных характеристиках.

Многие примитивы качества ПС трудно уловимы с точки зрения их (объективной) оценки. В этих случаях иногда применяют *метод экспертных оценок*. Этот метод заключается в следующем. Назначается группа экспертов и каждый из этих экспертов в результате изучения представленной документации составляет свое мнение об обладании ПС требуемым примитивом качества. Затем голосованием членов этой группы устанавливается оценка требуемого примитива качества ПС, т.е. получаемая оценка является некоторым усреднением совокупности субъективных оценок. Эта оценка может производиться как по двухбалльной системе ("обладает" - "не обладает"), так и учитывать степень обладания ПС этим примитивом качества (например, производиться по пятибальной системе) в соответствии с требованиями относительно этого примитива, сформулированными в спецификации качества аттестуемого ПС.

Аттестация ПС похожа на смотр различных компонент ПС в процессе управления качеством ПС, однако, имеются и существенные различия [1]. Во-первых, смотр проводится менее представительной группой специалистов. Во-вторых, в процессе смотра не производится полной оценки качества ПС, а выявляются лишь отдельные просчеты и нарушения требований относительно качества ПС, связанные с обозреваемой компонентой (документом), при этом не требуется немедленного устранения выявленных недостатков, если они не мешают проведения последующих работ. *Целью*же аттестации является проверка и фиксация реальных показателей качества предъявленного ПС [7]. Если аттестационная комиссия подтверждает, что предъявленное ПС соответствует всем требованиям относительно его качества, сформулированным во внешнем описании этого ПС, то считается, что его разработка завершена успешно и заказчик обязан принять это ПС. Если же будут обнаружены отступления от этих требований, то должны приниматься определенные решения о продолжении или прекращении разработки предъявленного ПС, но это уже вопрос взаимоотношений между заказчиком и разработчиками. Таким образом, аттестационная комиссия, подписывая документ о произведенной оценке качества ПС, принимает на себя определенную ответственность за гарантию качества этого ПС. Но здесь имеются определенные правовые проблемы, обсуждение которых выходит за рамки темы этой лекции.

**Список литературы**

1. Ian Sommerville. Software Engineering. – Addison-Wesley Publishing Company, 1992. – P. 479-493.

2. В.В. Липаев. Управление разработкой программных средств. Методы, стандарты, технология. – М.: Финансы и статистика, 1993.

3. Б. Шнейдерман. Психология программирования. – М.: Радио и связь, 1984. – С. 128-146.

4. Ф.П. Брукс, мл. Как проектируются и создаются программные комплексы. – М.: Наука, 1979.

5. Г. Майерс. Надежность программного обеспечения. – М.: Мир,1980. - С. 174-175.

6. Е.А. Жоголев. Введение в технологию программирования (конспект лекций). – М.: "ДИАЛОГ-МГУ", 1994.

7. В.В. Липаев, Е.Н. Филинов. Мобильность программ и данных в открытых информационных системах. – М.: Научная книга, 1997. – С. 252-268.

8. В.В. Липаев. Тестирование программ. – М.: Радио и связь, 1986. – С. 231-245.

9. Д. Ван Тассел. Стиль, разработка, эффективность, отладка и испытание программ. – М.: Мир, 1985. – С. 281-283.