

# Методика обучения программной инженерии на основе карт памяти\*

Д. В. Кознов  
dim@DK12687.spb.edu

В статье представлена методика проведения экзамена по университетскому курсу «Программная инженерия», основанная на картах памяти (mind maps) — известном подходе к структурированию информации, предложенном английским психологом Т. Бьюзенем. Карты памяти, составленные студентами в качестве ответов на вопросы экзаменационных билетов, являются объективным свидетельством уровня их знаний, очевидным и самим студентам, и преподавателям. Карты памяти хорошо поддерживают идею индивидуальной творческой активности, позволяя организовать процесс погружения студентов в информацию курса, проясняя и обобщая ее, формулируя свое отношение к полученным знаниям. В качестве программного инструмента для работы с картами памяти был использован программный продукт Comapping.

## Введение

При изучении многих наук и дисциплин есть объективные способы проверки полученных учеником знаний. Так, при обучении точным наукам (например, математике) для проверки можно предложить ученику решить задачу. В процессе обучения естественным наукам (физике, химии, биологии и др.) ученику предлагается выполнять лабораторные работы, в ходе которых он вынужден использовать имеющиеся у него теоретические знания, и по результа-

---

\* Работа по созданию и апробации данной методики была проведена при поддержке компании Hewlett-Packard.

© Д. В. Кознов, 2008

там этих работ можно уверенно судить об уровне усвоения материала. При обучении искусствам результат обучения также понятен и очевиден — исполненное музыкальное произведение, нарисованная картина и проч.

При изучении программной инженерии невозможно сформулировать точные критерии усвоения знаний. Например, владение программными продуктами поддержания жизненного цикла разработки ПО таким критерием не может быть, поскольку Software Engineering Curriculum недвусмысленно указывает на то, что обучение в этой области должно вестись не по каким-то избранным технологиям, а по общим принципам и закономерностям разработки ПО [1].

При обучении программной инженерии есть и другая проблема. На лекциях по этой дисциплине студенты получают знания «авансом», изучая проблемы индустрии и методы их разрешения, но не имея при этом практического опыта<sup>1</sup>. Очень трудно в такой ситуации сделать обучение эффективным. Однако оставить этот материал для усвоения на практике, как показывает опыт, нецелесообразно — в этом случае компании-работодатели тратят слишком много производственных ресурсов на обучение молодых специалистов. Более того, многие методы и приемы, с которыми такой специалист не столкнулся непосредственно, так и останутся за кадром его кругозора.

Проблема обучения программной инженерии активно обсуждается мировым сообществом: в [1] содержится программа университетских курсов, ряд нестандартных методов обучения предложен, например, в [2–4]. Однако по-прежнему остается открытым вопрос об эффективности традиционных лекционных занятий. При всех трудностях этого подхода применительно к обучению программной инженерии [2, 5] инерция традиции велика и лекции будут существовать в университетах еще долго. Кроме того, базовые знания по программной инженерии целесообразно изложить студентам в виде регулярной системы, и лекции здесь — идеальный инструмент.

В связи с этим возникает вопрос — как наладить процесс эффективного усвоения лекционного материала? Я предлагаю использовать для этой цели экзамен.

Ситуация, когда студенты на экзаменах стремятся дословно повторить преподавателя — нелепа и ведет к вырождению универси-

---

<sup>1</sup>Аналогичная ситуация имеет место при изучении студентами промышленных технологий в любых инженерных областях.

тетского образования. Давать им задачи, индивидуальные творческие задания? В области программной инженерии для этого используют учебные проекты, моделирующие реальные. Но такие проекты оказываются слишком большими заданиями для экзамена — их целесообразно выполнять в рамках курсового проекта или практических занятий. В то же время подготовка студента к экзамену и сам экзамен могут стать мощным инструментом в усвоении материала. Как сделать здесь усилия студентов максимально эффективными?

В середине 70-х годов прошлого века английским психологом Тони Бьюзеном была предложена техника работы с информацией — карты памяти (mind maps)<sup>2</sup>. Эта техника предлагает структурировать информацию, создавая диаграммы, на которых в центре изображается главный объект — фокус интеллектуальных усилий, а от него радиально рисуются связи и другие объекты, уточняющие, поясняющие и детализирующие центральный. Таким образом, люди могут «распутывать», прояснять и структурировать сложные массивы имеющейся у них информации, делая информацию «операбельной», то есть годной для удовлетворения их запросов и потребностей, совершения каких-либо действий. Например, школьникам, студентам и преподавателям предлагается использовать эту технику вместо обыкновенных текстов при подготовке к урокам, экзаменам, лекциям; писателям — для составления планов статей и книг; менеджерам — для составления планов; людям в их повседневной жизни — для структуризации и планирования своих действий. Карты памяти позволяют человеку сфокусироваться на главном в той информации, в которой он хочет сориентироваться, а также развивают у него навыки системного мышления. Однако это — лишь базовая идея, и для использования в конкретных областях деятельности, ситуациях, для разных людей и аудиторий она нуждается в уточнении и адаптации.

Идея данной работы — использовать карты памяти при проведении экзамена по программной инженерии, максимально приблизив с их помощью экзамен к семинару-погружению и ориентируя его не только на проверку знаний, но и на их более глубокое усвоение прямо во время самой сдачи. Карты памяти, составленные студентами в качестве ответов на вопросы экзаменационных билетов, являются объективным свидетельством уровня знаний, очевидным и са-

---

<sup>2</sup>Список книг Т. Бьюзена по этой тематике, переведенных на русский язык, можно найти на сайте <http://www.mindmap.ru/stat/booktb.htm>.

ним студентам, и преподавателям. На их основе можно прицельно направлять осмысление и изучение студентами тех или иных вопросов — либо прямо на экзамене, либо в качестве ориентиров при подготовке «второго захода». Карты памяти хорошо поддерживают идею индивидуальной творческой активности студентов. При этом акцент на экзамене делается не на выдачу заученных/списанных тезисов из конспектов и учебников, а на выявление, осознание, оформление и представление студентами их собственных взглядов на изученный материал. На основе карт памяти им предлагается строить новые, собственные цепочки связей между знакомыми понятиями курса, выделять то, что произвело впечатление, кажется наиболее важным, полезным на практике именно тому студенту, который отвечает (а не всем людям вообще!). В итоге студент глубже усваивает материал, делает для себя открытия, которые запоминаются, потому что они — его собственные. Здесь важно подбирать и предлагать оригинальные задания и грамотно организовывать процесс.

Данная методика была создана мной в контексте преподавания курса «Технология программирования ПО» в Санкт-Петербургском государственном университете (СПбГУ). В качестве программного инструмента для работы с картами памяти я использовал продукт Comapping<sup>3</sup>. В качестве аппаратного обеспечения использовались компьютеры HP Tablet PC, любезно предоставленные нашему университету компанией HP.

Я хотел бы выразить признательность студентам СПбГУ, вытерпевшим эксперименты по отработке данной методики, в особенности Юлии Йоффе, потратившей время на формулировку обратной связи по экзамену, а также на чтение и комментирование текста статьи. Я благодарен своим коллегам — А. Иванову, А. Абрамян, И. Соболеву, Р. Гагарскому, А. Терехову, В. Уфнаровскому, тепло и искренне поддержавших мою работу над этой статьей и высказавших много критических, порой весьма резких, но от этого особенно полезных, замечаний.

---

<sup>3</sup><http://www.comapping.com/>. Этот продукт — результат сотрудничества датской компании Atea9 и петербургской компании ЗАО «ЛАНИТ-ТЕРКОМ». Основная разработка и развитие продукта происходит в России, на базе математико-механического факультета СПбГУ. Это, конечно, было главной причиной выбора именно этого продукта для использования в методике. Но я искренне считаю, что Comapping — лучший из имеющихся на рынке аналогов.

## 1. Основные идеи

Сегодня в университетах значение экзаменов недооценивается. Часто они (явно или неявно) заменяются просто тестами, тем более, что современные IT-технологии предоставляют для этого много возможностей. Однако в русской университетской традиции, в частности на математико-механическом факультете СПбГУ, экзамен всегда был не только проверкой знаний, но и плодотворным общением профессора-лектора со студентами, значение и важность которого в образовательном процессе трудно переоценить. В результате такого общения профессор получает обратную связь о своем курсе<sup>4</sup>, а студенты непосредственно соприкасаются с носителем знаний. Когда я сам был студентом, экзамены часто длились по шесть-восемь часов — дискуссии с преподавателем, размышления над его вопросами и задачами, поиск ответов, обсуждение с ним очередного варианта решения и т. д. В непосредственном и живом общении стирается грань между предметом обсуждения и участниками, происходит слияние — эмоциональное, ментальное, интуитивное и проч. Рубятся границы «я — и предмет», создаются условия для качественно иного усвоения знаний, они проникают глубоко, оживают. Как говорил А. Эйнштейн, образование — это то, что остается после того, как все выученное забыто.

Основываясь именно на таком понимании экзамена, я поставил себе сверхзадачу превратить его в тренинг-погружение, используя карты памяти как основной инструмент. Я разделил экзамен на две части — общая проверка знаний курса и индивидуальное задание на основе карт памяти.

Общая проверка знаний нужна для того, чтобы, в случае, если знаний недостаточно, предложить студенту подготовиться дополнительно и прийти на пересдачу: в такой ситуации творческое задание для него преждевременно. Кроме того, нужен какой-то способ ориентации студентов на знание всего курса в целом, и не в виде общего понимания, а в виде знания конкретных определений, тер-

---

<sup>4</sup>Наблюдать, как разные студенты тебя поняли — интересно, увлекательно и полезно. Это общение очень животно, если удастся избежать «противостояния на баррикадах». Одновременно оно достаточно затратно, так как необходимо встать на точку зрения студента, увидеть курс его глазами. Но это общение порождает новые идеи, позволяет увидеть разных людей, познакомиться с их отношением к предмету, да и просто понять, что их волнует, на что направлены их усилия в жизни. Без всего этого лектор может оторваться от реальности и читать курс для себя.

минов, дат и имен. Следующий далее этап ориентирован на существенно иные аспекты, оставляя задачу точного знания в стороне. Для того чтобы осуществить такую проверку быстро, я сделал список вопросов по всему курсу, озаглавив его «Вопросы “на вылет”». Таких вопросов около ста штук, они достаточно простые. Студенты получают этот список до экзамена, в качестве ориентира для подготовки. Отвечать на эти вопросы они должны в режиме блица, то есть сразу. Если из пяти-шести вопросов студент отвечает неправильно на два и больше, то это означает, что он не готов к экзамену.

Успешно прошедшие первый этап переходят к следующему заданию. Оно дается на основе другого списка вопросов — экзаменационного. Таких вопросов значительно меньше, чем вопросов «на вылет» — всего около двадцати. Они представляют собой большие и относительно независимые темы курса и являются обширным полем для структурирования с помощью карт памяти. При подготовке ответа студенту разрешается использовать весь доступный ему материал, вплоть до Интернета<sup>5</sup>.

На основе экзаменационных вопросов и исходя из уровня своих знаний, профессиональной подготовки и мотивации на изучение предмета, студент получает задачу, выполнение которой может занять много времени — несколько часов и больше. Студент может не успеть ее выполнить за время экзамена, и тогда ему предлагается доделать ее в свободное время, предъявив результаты на следующей передаче. Во время выполнения задачи поощряется общение студента с преподавателем — можно задавать вопросы, нужно предъявлять промежуточные результаты. Возможен вариант, когда одно трудное задание получает группа из двух-трех человек и выполняет его в течение одной недели, с публичной защитой в конце. При работе над задачей вне экзамена дистанционное общение преподавателя и студентов (задание вопросов/получение ответов, предъявление промежуточных вариантов/получение обратной связи) происходит по электронной почте и с помощью инструмента *Comapping*.

При проведении экзамена важно максимально использовать индивидуально-групповое взаимодействие, то есть гармонично со-

---

<sup>5</sup>Мне очень не нравится следить за тем, чтобы студенты не списывали. Получается какой-то детский сад, что особенно нелепо на последнем году обучения в университете (курс «Технология программирования ПО» читается мной на V курсе).

читать работу на экзамене со всеми присутствующими студентами как с единой группой, а также уделять необходимое внимание каждому студенту в отдельности. При этом важно следить за тем, чтобы экзамен не превращался только в работу преподавателя с отдельными студентами — это будет тяжело (для преподавателя) и неэффективно (для студентов).

Один «сеанс» (другими словами, сдача) рассчитывается на 10–20 человек<sup>6</sup>. Если студенты подбираются сильные и мотивированные, то легко устраивается семинар-погружение длительностью в 4–6 часов. Складывается рабочая атмосфера, которую нужно только поддерживать — не использовать мобильных телефонов, соблюдать элементарную дисциплину и проч. Кроме того, желательно проводить экзамен в тихом и уединенном месте. Общую атмосферу можно также создавать, давая всем присутствующим одно задание, которое каждый должен выполнять индивидуально. При этом карты памяти оказываются отличным способом индивидуализировать работу: у студентов не возникает желания списывать друг у друга, каждая карта оказывается по-своему уникальна (в приложении представлены три такие карты (рис. 2, 3, 4) на одну и ту же тему). Общее задание можно предварить небольшим объяснением (на 20–30 минут), в свободном режиме, часто вступая в диалог со студентами, предлагая им высказываться по тому или иному вопросу. И далее, при разработке карт памяти, периодически указывать всем участникам на некоторые общие проблемы, трудности — в общем групповом процессе возникают порой удивительные резонансы. Но в любом случае я всегда по многу раз просматриваю промежуточные результаты каждого студента в отдельности, давая конкретно ему рекомендации и уточнения.

## 2. Методика по шагам

Поток студентов, которому читается данный курс, насчитывает примерно 100 человек. В последний раз я проводил около 10 сдач средней продолжительностью по 5 часов. Кроме того, была еще интенсивная дистанционная работа и несколько встреч с небольшими группами по 2–3 человека. В среднем на одной сдаче присутствовало 18 человек. Почти каждый студент посетил экзамен дважды, некоторые — по несколько раз.

---

<sup>6</sup>Если меньше — не создается общей атмосферы, больше — тяжело работать.

Многие сдачи сильно отличаются друг от друга — разный уровень подготовленности студентов, разные задания и формы проведения. Однако в целом экзамен имеет единый сценарий, к изложению которого я и перехожу.

### **2.1. Проверка знаний**

Кроме проверки собственно знаний студента по курсу здесь происходит определение его мотиваций. Данный курс читается мной на пятом, заключительном году обучения, и, как правило, большинство студентов уже определились с выбором профессии. Как показывает опыт, разброс вариантов здесь очень большой. И если студент выбрал, например, не IT-сферу (к примеру, деятельность менеджера по продаже недвижимости), то его заинтересованность в данном предмете крайне низка. Встречаются также просто равнодушные люди, которым кроме зачета ничего не нужно. С такими людьми углубленно работать, применяя передовые, экспериментальные методы — очень неблагоприятное дело. В подобных случаях я ограничиваюсь вопросами «на вылет» и средней оценкой. Другое дело, если студент выбрал карьеру программиста. Тогда работать с ним интересно и осмысленно. Чем больше заинтересованности у студента, чем он компетентнее, чем больше он хочет работать, тем он больше работает на экзамене, тем труднее и интереснее ему предлагаются задания, тем больше пользы и удовлетворения он получает от процесса.

### **2.2. Выбор задачи**

Как говорилось выше, задачи предлагаются на базе особого списка тем — экзаменационных вопросов, крупными блоками покрывающих весь курс. Я предлагал следующие задачи (представлены по мере убывания сложности).

1. Предложить собственное видение того или иного экзаменационного вопроса. При этом нельзя было повторять структуру лекций, учебников и проч. Часто вопрос предлагается из области практической компетентности студента. Иногда я советовал выбрать тот вопрос, в котором студент лучше всего ориентируется. Для этого задания важно, чтобы у студента были достаточно глубокие знания — выше среднего уровня. Именно тогда может существовать это собственное видение,



- и тогда его целесообразно осознавать, оформлять, «отшелушивать», используя карты памяти.
2. Раскрыть и нарисовать какое-нибудь сложное определение (например, определение тестирования, конфигурационного управления, проекта).
  3. Нарисовать карту всего курса — обозначить главные темы, понятия и показать их взаимосвязи.
  4. Нарисовать все связи какого-либо одного экзаменационного вопроса с остальным курсом.
  5. Нарисовать все взаимосвязи некоторого множества экзаменационных вопросов<sup>7</sup>.
  6. Нарисовать карту памяти для темы «Программная инженерия», создавая ее как конспект своей собственной лекции, составленный перед тем, как осветить эту тему для своего товарища-программиста на работе, в качестве ответа на его вопрос: «А что же такое программная инженерия?»
  7. Изобразить с помощью диаграммы какую-нибудь одну тему, используя конспект, какую-либо книгу, учебник и пр. При этом связи должны иметь имена, диаграмма в целом — легко читаться, как текст, лекция и проч.

### 2.3. Проверка правильности понимания задачи

Оказалось, что подобные задачи, при кажущейся простоте, весьма трудны для выполнения. Часто студенты понимают их так, что нужно открыть конспект в определенном месте и перерисовать с

---

<sup>7</sup>Оказалось, что студентам не просто создавать подобные модели. Здесь можно предложить следующий способ. Сначала обозначить основные сущности моделируемой области. Потом соединить их связями — не стесняясь, соединять одну пару сущностей двумя, тремя и большим количеством связей. Затем детализировать связи, разбивая и декомпозируя связываемые сущности на составные части. Посмотрим, например, как темы *процесс* и *СММ* можно связать друг с другом. Главный фокус СММ — это *процесс* разработки ПО, СММ предлагает определенную модель *улучшения процесса*, а также определенную *модель процесса разработки* и т. д. А в курсе есть отдельные темы — *улучшение процесса разработки* и *модели процессов разработки*. Таким образом, у СММ появляется множество атрибутов, через которые и осуществляется связь этой темы с темой *процесс*. Все эти дополнительные атрибуты целесообразно рисовать как отдельные сущности, связанные с СММ, и от них уже рисовать связи с темой *процесс*. Такой способ выполнения данного задания — далеко не единственный.

помощью карт памяти его структуру. Однако это самый легкий вариант задачи (см. список из предыдущего раздела), и далеко не единственный. Тот факт, что на один и тот же материал может существовать несколько точек зрения, осознание конкретной точки зрения, которая задается задачей, — все это не просто осуществить на практике<sup>8</sup>.

Чтобы удостовериться в правильности понимания задачи, я обычно прошу студента написать формулировку своими словами, предлагаю задавать уточняющие вопросы. Однако в последнем случае часто получается, что хитрые студенты пытаются выудить, что конкретно я хочу увидеть в итоге. На первых порах часто случалось, что я вместо них рисовал диаграммы, а они их только дополняли. И то с большими проблемами, поскольку я, увлекаясь, задавал такой процесс, в который им было никак «не встроиться»<sup>9</sup>.

---

<sup>8</sup>Концепция точки зрения (view point) моделирования подробно разрабатывается в специальном разделе программной инженерии — «Визуальное моделирование программного обеспечения», который посвящен созданию языков, методов и программных инструментов для визуализации ПО на основе графовых нотаций [6]. Впервые эта концепция появилась в рамках метода SADT, еще в 60–70-х годах прошлого века, и обозначала ту позицию, в которую должен встать автор модели, чтобы увидеть моделируемую систему в действии, происходящем сообразно выполненной им модели [7]. В начале 90-х годов, при разработке объектно-ориентированных методов анализа и проектирования, концепцию точки зрения впервые ясно сформулировал Филипп Крютхен в своей знаменитой работе [8]. И далее, известный язык визуального моделирования UML (стандарт международного комитета OMG — [www.omg.org](http://www.omg.org)) также создавался как набор разных типов диаграмм, позволяющих моделировать систему с разных точек зрения. Однако на практике использование разных точек зрения часто сводится всего лишь к созданию разного типа диаграмм. А это очень упрощенная трактовка данной концепции — ведь и с помощью одного типа диаграмм можно моделировать один и тот же объект с разных точек зрения. Например, с помощью диаграмм активностей UML можно составить три модели сделки с недвижимостью — с точки зрения продавца, покупателя и агента — и все это будут разные модели одной и той же сделки! Однако, чтобы суметь создать эти модели, необходимо на момент моделирования стать тем, с чьей точки зрения происходит взгляд на процесс. Осознать, увидеть, почувствовать того, кем ты хочешь стать (его стиль работы, мотивы, потребности цели, задачи и проч.), и стать им — главная задача и главная трудность при практическом использовании идеи точки зрения моделирования.

<sup>9</sup>Это часто случается на экзаменах и не имеет прямого отношения к картам памяти. Преподавателю кажется, что он помогает студенту, а студент никак не может взять в толк, чего от него хотят. Дело тут в том, что преподаватель, так «помогая», не становится на точку зрения студента, по-прежнему предлагая недоступные для него (и часто очень индивидуализированные им, преподавателем) знания, и поэтому между ними не возникает понимания.

И дело тут не только в студенческой хитрости. За все время обучения студенты привыкают к тому, что на экзамене они должны воспроизвести правильный ответ. При этом подразумевается, что правильный ответ — единственный. Более того, математическое образование предполагает, что у ответа должна быть конструктивная идея, — использование тех или иных базовых теорем в доказательстве, применение определенного метода для расчетов и проч.

Здесь же у нас ситуация принципиально иная.

1. Существует бесконечное множество разных правильных ответов. Их тем больше, чем выше компетенция студента в данном вопросе. Нахождение и проработка собственного, уникального ответа — отличная возможность глубоко погрузиться в изучаемый материал.
2. У ответа нет простого алгоритма, схемы, идеи, которые можно подсказать, не навязав какого-либо собственного видения. Если такое «навязывание» происходит, то нарушается одна из главных идей данной методики — пробуждение в студентах самостоятельности в работе с лекционным материалом.

#### **2.4. Налаживание работы над заданием и оценка первых результатов**

Итак, студент должен сам найти опорные точки при создании собственной карты памяти. Кто-то идет от визуальных образов, кто-то — от любимейшей ему темы, кто-то — от общего хорошего знания курса и т. д.

Для преподавателя важно увидеть найденное студентом зерно — сами студенты часто не всегда осознают, что нашли оригинальный подход к решению задачи. Далее целесообразно предложить студенту максимально развить именно это зерно, а не гнаться за мифической полнотой изложения и другими «химерами», затуманивающими самостоятельный и оригинальный взгляд на предмет.

#### **2.5. От диаграмм к содержанию курса**

После того как студент разобрался с инструментарием, то есть понял, как ему следует эффективно работать, наступает самая интересная часть — работа через диаграммы с содержанием курса. Важно донести до студента (и не потерять самому преподавателю!)

идею того, что карты памяти важны не сами по себе, а как отражение знаний студента о предмете. Часто бывает, что диаграмма плоха не потому, что неудачно нарисована, а потому, что знания студента недостаточны, поверхностны, фрагментарны. Именно это не дает возможности нарисовать хорошую диаграмму. Если в этой ситуации сосредоточиться на «изобразительном» аспекте, то это — выброшенное время. Необходимо в таких случаях предлагать студенту разобраться в том материале, который у него плохо изображается, который некрасиво, уродливо выглядит.

Кроме того, оригинальная диаграмма является хорошим поводом задать студенту множество вопросов по материалу курса. Например, если он, рисуя карту всего курса, поместил в центр тему «Управление проектами» (я бы так делать не стал, но он может считать, что в программной инженерии это самое главное), то целесообразно спросить у него, почему он так считает, как он понимает эту тему, что такое проект и т. д.

Опыт показывает, что часто оказывается трудно отвлечь студентов от кубиков, овалов и стрелок — студенты заикливаются на том, что надо нарисовать диаграмму и сдать ее преподавателю. Здесь сказывается недостаток у студентов умения структурировать материал, несвобода во владении диаграммной техникой представления знаний, а также ошибки и неточности преподавателя при формулировании задач. Иногда бывает, что студенты ведут себя подобным образом от безнадежности, поскольку никак не могут понять, как надо выполнять задание.

### 3. Об инструментах

После постановки задачи и осознания ее условий первые шаги я рекомендую делать без каких-либо инструментов вообще, вспоминая материал, перечитывая конспект, исходные документы (тексты стандартов и проч.). Сходный процесс я описывал в работе [9], когда речь шла о проектировании уникальной системы. То есть не нужно торопиться со структурированием, — эффективно упорядочивать и классифицировать можно лишь имеющуюся информацию, а если ее или нет, или недостаточно, то и структурировать нечего. Сначала следует набрать, осознать избыток информации, в котором затем и наводить порядок посредством карт памяти<sup>10</sup>.

<sup>10</sup>Здесь, впрочем, возможны и иные подходы. Например, когда знаний мало, целесообразно структурировать и рисовать именно «незнание», чтобы полу-

Не следует также в самом начале уделять много внимания тому, как именно структурировать. Был случай, когда я очень подробно объяснял студенту, как именно я бы структурировал информацию в его случае, но дело встало, поскольку после этого он столкнулся с недостатком знаний по материалу лекций. Ситуация приобрела статус глобального кризиса — студент не мог представить никакого целостного результата, объясняться с ним становилось все сложнее. И я только позже осознал, что произошло.

Первичное структурирование информации часто полезно выполнять с помощью ручки, бумаги и произвольных диаграмм. Както рисуется... Такой подход дает больше свободы, чем использование программных инструментов, он лучше ассистирует психические процессы человека по выявлению первых очертаний структурируемого материала<sup>11</sup>.

Следующий шаг — использование карт памяти. В информации появляется центр, ядро, и от него к периферии расходятся уточнения. Карты памяти можно рисовать цветными карандашами, фломастерами на бумаге, а можно — в программном продукте. Есть много сторонников и противников обоих подходов. Я использовал оба варианта. Рисование «от руки» позволяет использовать различные художественные образы при оформлении карт — примеры интересных и необычных карт памяти можно посмотреть в [12]. Однако в этом случае требуется много работы для перерисовки диаграмм, а это часто приходится делать, особенно если у тебя не очень много подходящего опыта. Кроме того, полезно иметь «красивый», «чистовой» вариант для каждой следующей итерации. Программный продукт в этом случае более удобен. Еще один случай, где он также удобен, — при дистанционной работе, когда люди работают с одной картой памяти, физически находясь в разных местах.

Для дистанционной работы мы использовали программный продукт информации о том, что в первую очередь нужно изучать. Пример такого структурированного «незнания» я привел в работе [10], где представил первую диаграмму, составленную в самом начале изучения сложной программной системы техническим писателем. Однако нужно очень ясно понимать, что именно структурируешь — знание или незнание. В любом случае структурируется изобилие, избыток.

<sup>11</sup>По поводу свободы в использовании различных подходов и нотаций при проектировании ПО очень интересно высказывался Мартин Фаулер в своей знаменитой книге «UML, основы» [11]. Я целиком согласен с его воззрениями на этот счет и распространяю их на структурирование информации в любой области, не только при разработке ПО.

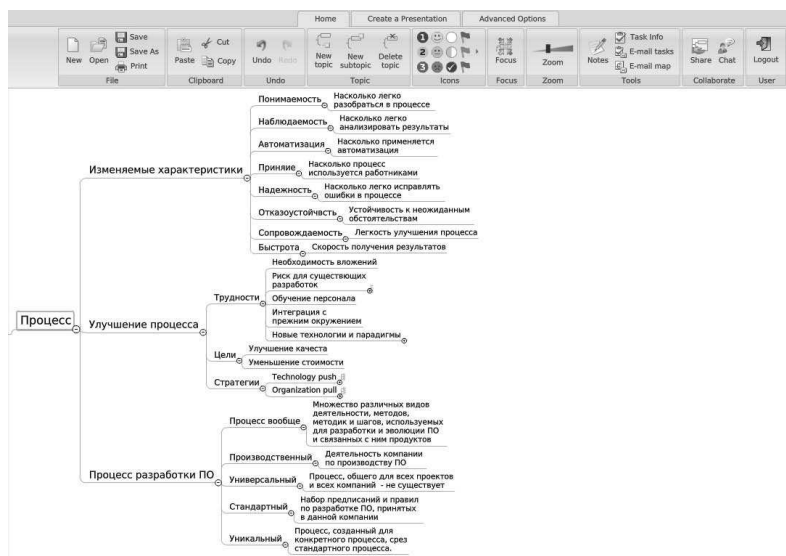


Рис. 1. Пример карты памяти — ответ на экзаменационный вопрос «Процесс разработки ПО».

дукт Comapping, который является Web-сервисом, не требует инсталляции (в компьютерных классах, где проходит экзамен, дома у студентов и преподавателей и т. д.). Созданная в нем диаграмма хранится на сервере, для работы с ней можно задать целый список пользователей. После этого, если один из них изменил диаграмму, он может послать прямо из Comapping уведомление по электронной почте другим участникам. Они, просматривая диаграмму, сразу видят изменения, помеченные специальным образом с указанием их автора. Продукт может работать как в режиме соединения с Интернетом, так и без него (например, при обрыве соединения). В последнем случае можно сохранить диаграмму локально, на собственном компьютере, а при появлении соединения переписать на сервер.

Продукт Comapping позволяет создавать карты памяти в виде деревьев, имеет удобные средства редактирования, просмотра и печати. Он имеет очень простой, приятный и удобный интерфейс. Описать детали этого удобства трудно (их слишком много), но все

вместе они складываются в очень позитивное впечатление от работы с продуктом. Пример диаграммы в Comapping, составленной одним из студентов в качестве решения задачи на экзамене, представлен на рис. 1.

Использование ноутбуков Tablet PC HP позволило нам проводить экзамен в непринужденной и свободной обстановке, временами приближая его к тренингу-погружению. Каждый студент имел свой компьютер и мог расположиться в любом месте аудитории, а также за ее пределами. Поддержка WiFi технологии для выхода в Интернет, наличие планшетов, возможность долгое время работать без подзарядки — все это было очень удобно.

## Заключение

Я предвижу вопросы относительно того, почему карты памяти использовались именно на экзамене, а не в процессе всего обучения, в течение семестра. Ответ очень простой: потому, что именно эта часть обучения оказалась наиболее подконтрольна преподавателю ввиду свободного посещения лекций на последних курсах нашего факультета, и потому, что почти все студенты к этому времени уже работают в бизнес-компаниях. Использование какой-либо методики требует погружения и специальных усилий — все это оказалось достижимо именно на экзамене. Однако для более успешного проведения таких мероприятий студенты нуждаются в дополнительной подготовке. Им неожиданно трудно оказалось структурировать свои знания — нет соответствующих навыков, сильна традиция «зубрежки», и ориентации на единственно правильную точку зрения, на единственно верный ответ. Соответственно студенты тратят много усилий на поиск этого ответа вместо непосредственной работы с информацией. Необходимы специальные занятия, где бы навыки структурирования развивались отдельно, а также воспитывалось бы более свободное и творческое отношение к учебной информации.

Я понимаю, что предложенная методика нуждается в дальнейшей доработке и более качеством воплощении. В частности, задач должно быть больше, а их методическая точность и проработанность — существенно выше. Отмечу также, что прием экзамена по приведенной методике весьма затратен и требует наличия не одного преподавателя, а целой группы, которую также нужно специально готовить.

Я осознаю, что методик, подобных предложенным в этой статье, можно придумать очень много. Я далек от оптимизма Тони Бьюзена и его единомышленников, утверждающих, что карты памяти — это радикальный и суперэффективный способ мышления. Человек, его сознание, его мышление устроены сложнее и многообразнее, чем какая-либо одна (тем более такая простая) концепция. Я считаю, что карты памяти — это хорошая, но не единственная основа для создания своих собственных способов структурирования информации<sup>12</sup>. Каждый творческий человек, интенсивно работая с информацией, непременно создаст свой метод. Тем более, что даже если брать карты памяти как таковые, то при их практическом использовании возникает множество индивидуальных особенностей, больших и малых методических задач, которые нужно разрешать.

Я далек от идеи распространять мою методику для использования другими «as is». Истинная цель моей статьи — призвать активнее использовать в высшей школе нестандартные подходы, делая обучение более современным, точнее отвечающим потребностям изменяющегося мира, а также менее занудным и скучным. Я надеюсь, что высказанные в статье идеи найдут отклик у других преподавателей, а также заинтересуют студентов, в итоге подвигнув и тех, и других к собственным экспериментам.

## Список литературы

- [1] Рекомендации по преподаванию программной инженерии и информатики в университетах / Пер. с англ. М.: ИНТУИТ.РУ Интернет-Университет Информационных Технологии, 2007. 462 с.
- [2] *Pavlov V. L., Boyko N., Babich A. et al. Applying Pantomime and Reverse Engineering // Frontiers in education conference — global engineering: knowledge without borders, opportunities without passports, 2007. FIE '07. 37th annual. Section T1E. October 10–13, 2007, Milwaukee, WI. P. 1–5.*
- [3] *Stiller E., LeBanc C. Effective Software Engineering Pedagogy // Journal of Computing Sciences in Colleges, Volume 17, Issue 6, May 2002. P. 124–134.*

---

<sup>12</sup>В [13] я привел метод «одной страницы», используемый мной при руководстве дипломными работами. Иногда я использую еще один подход к разработке дипломов и статей — упор на создание качественного литобзора, который позволяет выделить в работе фокус и расставить акценты. И так далее. Какой конкретно метод использовать и когда, а может, изобрести новый — все это «the matter of practice», то есть решается в ходе живого процесса, исходя из особенностей сложившейся ситуации.



- [4] *Bracken B.* Progressing from student to professional: the importance and challenging of teaching software engineering // Journal of Computing in Colleges, Volume 19, Issue 2, December 2003. P. 358–368.
- [5] *Гагарский Р.* Программа подготовки специалистов в IT-компаниях // См. настоящий сборник.
- [6] *Кознов Д. В.* Основы визуального моделирования: Учебное пособие. М: Интернет-Университет Информационных Технологий; БИНОМ. Лаборатория знаний, 2008. Серия «Основы информационных технологий». 246 с.
- [7] *Marca D. A., McGowan C. L.* SADT Structured Analysis and Design Technique. McGraw-Hill, 1988.
- [8] *Kruchten P.* The 4+1 View Model of Architecture // IEEE Software, 1995, 12(6). P. 42–50.
- [9] *Кознов Д. В., Перегудов А. Ф.* «Человеческие» особенности в использовании UML // Системное программирование. Вып. 1 / Под ред. А. Н. Терехова, Д. Ю. Булычева. СПб.: Изд-во СПбГУ, 2005. С. 4–17.
- [10] *Кознов Д. В., Перегудов А., Романовский К. и др.* Опыт использования UML при создании технической документации // Системное программирование. Вып. 1 / Под ред. А. Н. Терехова, Д. Ю. Булычева. СПб.: Изд-во СПбГУ, 2005. С. 18–35.
- [11] *Фаулер М.* UML, основы. 3-е изд. / Пер. с англ. М.: «Символ-Плюс», 2006. 192 с.
- [12] *Бьюзен Т., Бьюзен Б.* Супермышление / Пер. с англ. Минск: «Попури», 2003, 304 с.
- [13] *Koznov D. V.* Visual Modeling in the Software Management // Proceedings of 2nd International Workshop «New Models of Business: Managerial Aspects and Enabling Technology» / Ed. by N. Krivulin, 2002. P. 161–169.

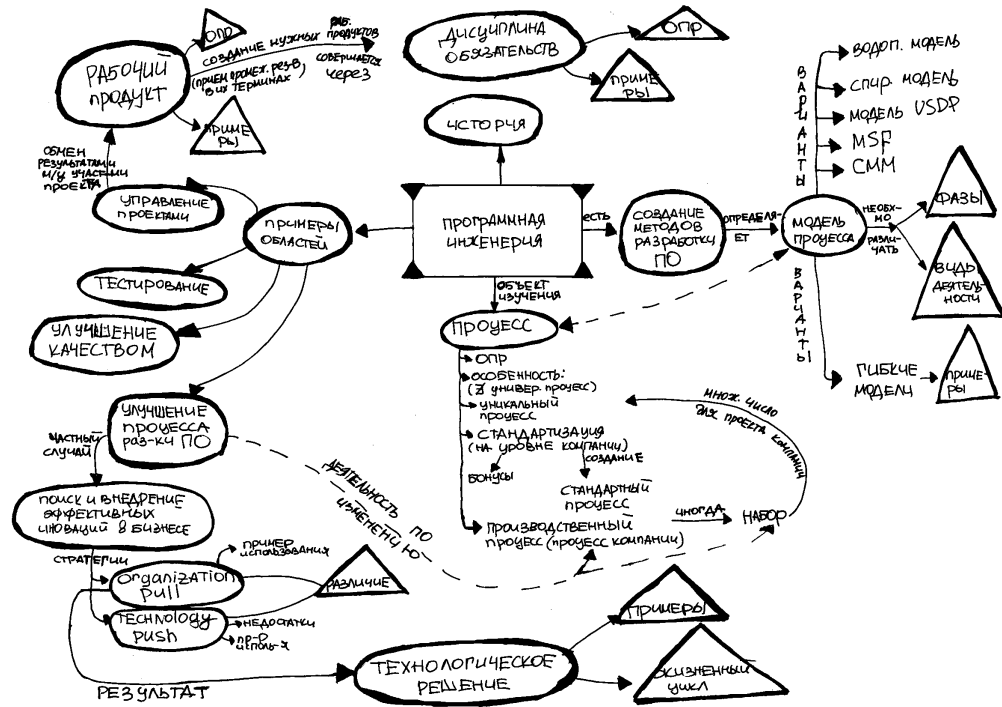


Рис. 2. Карта памяти, нарисованная студенткой Кристиной Лочаковой, на тему «программная инженерия».



