

# 1

## В начале творческого пути

---

### В ЭТОЙ ГЛАВЕ

- ✓ Определения и происхождение термина «творческие способности»
- ✓ Почему стоит быть творческой личностью
- ✓ Обзор семи составляющих Креативного Программиста
- ✓ Тест: решение проблем в стиле креативного программирования

Нам, людям, свойственно созидать. *Homo faber*<sup>1</sup> — человек, созидающий для того, чтобы взять в свои руки судьбу и окружающую среду, — проявляет, по словам философа и романиста Умберто Эко<sup>2</sup>, врожденную человеческую сущность в природе. Приобретя эту книгу, вы сделали первый шаг к вашей врожденной сущности креативного программиста. Поздравляю вас — и добро пожаловать!

Возможно, вы решили прочесть эту книгу, чтобы стать более сильным программистом. Вы обратились по адресу. Только не стоит ожидать, что здесь вам откроются новейшие технические чудеса, такие как динамический компилятор

---

<sup>1</sup> *Homo faber* (лат.) — человек творящий. — *Примеч. ред.*

<sup>2</sup> Эко У. «Открытое произведение», 2004.

некой виртуальной машины, или что вы узнаете более подробно о языке  $x$  или  $y$ . Эта книга далека от привычных книг по программированию.

Мы будем работать на другом уровне. Вы узнаете, как подходят к решению проблем в высшей степени творческие личности (и группы), каковы их привычки и мыслительные процессы и как они находят решения, являющиеся одновременно и результативными, и творческими. Когда вы станете сертифицированным Креативным Программистом, вы будете с легкостью решать любые технические загадки и быстро осваивать сразу несколько языков программирования — во всяком случае, в теории. Выбрали вы программирование как новую для себя область знаний или являетесь опытным разработчиком, я надеюсь, что вы овладеете по меньшей мере несколькими творческими приемами и примете их на вооружение.

Большой опыт в такой технической отрасли, как программирование, не обязательно означает большой творческий потенциал. Я работаю в индустрии разработки ПО свыше десяти лет и был свидетелем немногих взлетов и многих падений. Казалось бы, ПО ждет неминуемый провал. Прагматичный программист и соавтор «Agile-манифеста разработки программного обеспечения» («Manifesto for Agile Software Development») Энди Хант начал свою книгу «Pragmatic Thinking & Learning» на созвучной этому тревожной ноте:

*Программист вы или расстроенный пользователь, возможно, вы уже начали подозревать, что разработка ПО является самым трудным предприятием, которое когда-либо замыслил и воплотил человек. Его сложность заставляет нас работать на пределе наших способностей, а неудачи часто бывают весьма зрелищными и служат пищей для новостей.*

Подход Энди заключается в том, чтобы научить думать и познавать новое; мой же состоит в том, чтобы научить относиться к проблемам творчески. Будучи свидетелем множества провалов программных продуктов (и [бес]сознательно помогая их осмыслить), я пришел к убеждению, что проблема может заключаться не в технических знаниях, а в навыках, не имеющих отношения к технике. Одержимость этой мыслью вновь привела меня к научно-образовательному сообществу, которому я посвятил последние четыре года, исследуя творческие способности в разработке ПО. Плод моей гибридной научно-программистской работы вы держите в руках (если, конечно, вы предпочитаете читать старые добрые бумажные книги). Однако прежде чем мы приступим к работе, нужно найти ответы на некоторые вопросы.

## **1.1. ЧТО ЖЕ ТАКОЕ ТВОРЧЕСТВО?**

Ученые-психологи бьются над этим вопросом десятилетиями. Результатом является около ста разных определений творчества. Если вы зададите вопрос о том,

что такое творчество, вашей десятилетней дочери, то чтобы разобраться в этом, она предложит вам вместе нарисовать что-нибудь красками и кисточкой. Ваш прижимистый сосед считает, что проявлением творческих способностей является уход от налогов. А вы сами, покопавшись в начинке компьютера, можете прийти к выводу, что и ваша дочь, и сосед ошибаются: настоящие творцы — инженеры, которые все это придумали! Так кто же прав?

Возможное решение заключается в том, чтобы свести суть всех различающихся мнений к одному определению. Исследователи креативности Кауфман и Стернберг<sup>1</sup> утверждают, что идея является творческой, если соответствует следующим признакам:

- Считается новаторской и оригинальной.
- Обладает высоким качеством.
- Имеет непосредственное отношение к текущему заданию.

Переход на базу данных NoSQL может являться качественным (в смысле «не количественным») решением, как было доказано в прошлом, но я сомневаюсь, что эта идея является оригинальной. Если проблема не связана с данными, то она (идея) может даже оказаться не имеющей отношения к делу. Впрочем, если вы или ваша команда до этого никогда не работали с NoSQL, то она может считаться новаторской.

Этот эссенциалистский подход к творчеству несет в себе множество недостатков: в частности, он полностью игнорирует контекст. Исследования креативности постепенно сдвигаются по направлению к более *общесистемному* подходу, принимающему во внимание параметры контекста. Звучит сложновато, и я уже представляю, как вы готовитесь услышать очередное сухое научное определение. К счастью, все не так.

**УПРАЖНЕНИЕ** Как вы думаете, в каком случае можно считать что-либо творческим? Поразмышляйте над этим вопросом несколько минут. Когда в последний раз вы предлагали решение, которое было воспринято как необычайно творческое?

Готово? Хорошо. Приоткрою завесу тайны: что-либо можно считать творческим, если еще кто-то считает это таковым. Просто, не так ли? Креативность — это *социальный вердикт*<sup>2</sup>. Воплотились ли ваши усилия как программиста в творческий результат, решают ваши коллеги. По всей видимости, вы не можете сами провозгласить что-либо творческим. Это социокультурное явление.

<sup>1</sup> James C. Kaufman and Robert J. Sternberg. Creativity. Change: The Magazine of Higher Learning, 2007.

<sup>2</sup> Vlad Petre Glăveanu, Michael Hanchett Hanson, John Baer, et al. Advancing creativity theory and research: A socio-cultural manifesto. The Journal of Creative Behavior, 2020.

Эксперты в области искусства, которые объявляют ту или иную картину творением гения, диктуют свое мнение нам, дилетантам (рис. 1.1). В ответ мы послушно замираем в благоговении. А если бы эта картина была сочтена критиками примитивной и неинтересной, мы бы не удостоили ее и взглядом. Скорее всего, она никогда не оказалась бы в числе музейных экспонатов. Поскольку у нас нет необходимых технических знаний о живописи, нам приходится полагаться на мнения экспертов в данной сфере.



**Рис. 1.1.** Что есть искусство? Согласно Марселю Дюшану (Marcel Duchamp), произведением искусства может являться писсуар с автографом. Нью-Йоркская галерея отклонила его — и все же «Фонтан» Дюшана произвел известный фурор в искусстве XX века. Сейчас произведение считается в высшей степени творческим. Фото из открытых источников

То же справедливо в отношении программирования, да и любой другой сферы деятельности. Если члены вашей команды хлопают вас по плечу, приговаривая: «Отличный код! Творчески обошел проблему!», — то, стало быть, вас вдруг повысили до звания «Креативный программист», при условии, что члены вашей команды, играющие роль экспертов, не просто проходили мимо. Впрочем, в другой компании то же самое решение может быть сочтено скучным: там все это уже видели.

Систематический подход к творчеству объясняет также печальный количественный перевес непризнанных гениев. Само собой разумеется, что если бы ни одна картина Винсента Ван Гога не была обнаружена, мы не считали бы его гением творчества. Мы не считали бы его таковым и в том случае, если бы ни один эксперт в области искусства не признал картины Ван Гога экспрессивными и новаторскими. Собственно, именно это и происходило при его жизни. Его работы раз за разом отклоняли кураторы Парижского салона<sup>1</sup>, которые традиционно были ответственны за официальные художественные выставки Академии изящных искусств (Académie des Beaux-Arts) в период между

<sup>1</sup> Парижский салон (*фр.* Paris Salon) — одна из самых престижных художественных выставок Франции, официальная регулярная экспозиция парижской Академии изящных искусств. — *Примеч. пер.*

1748 и 1890 годами. Но их консерватизм продлился недолго. Современные художники в конце концов сместили с пьедестала классицизм и возвели на опустевшее место импрессионизм; импрессионисты начали организовывать собственные независимые выставки. Как мы увидим далее, время и место являются в равной степени значимыми факторами творчества. Многие работы Ван Гога ныне входят в число самых дорогих картин из когда-либо проданных.

### Истоки творчества

Восприятие творчества много раз менялось на протяжении истории человечества. В наши дни, стоит нам услышать это слово, на ум сразу приходит искусство. В античной Греции искусство (по-гречески — *technê*, это слово позже было преобразовано в слово «технология») подразумевало строгое следование правилам. Художники и скульпторы лишь имитировали, они не создавали — и только поэтам дозволялась «свобода действий». Люди творческих профессий открывали, а не изобретали.

Позднее в Европе, где господствовало христианство, созидание стало подразумевать исключительно сотворение богом из ничего (*creatio ex nihilo*). Мы, скромные люди, просто делали себе разные вещи. Мы не могли — нам не дозволялось — создавать<sup>1</sup>.

Только в эпоху Возрождения философы и люди искусства начали осознавать себя изобретателями, воплощая новые объекты в соответствии с собственными представлениями, постепенно освобождая искусство от ремесленности и двигаясь по направлению к созиданию. Понадобилось еще два века для того, чтобы этот термин был применен по назначению, что встретило серьезное сопротивление со стороны христианства.

Творчество не служило предметом научных исследований до 1950-х годов. Как и программирование, это относительно новое понятие!

Я мог бы разработать хитроумный вариант ext4, самой популярной файловой системы для Linux. Наверное, я назвал бы его «WouterFS». В этом так и сквозит креативность (и упадок). Если я никому об этом не скажу, то проект обнаружат разве что после моей смерти. К счастью, я реалист. Технологии развиваются стремительно, и к тому времени выйдет, вероятно, ext65, что, скорее всего, сделает WouterFS ненужной. А может быть, когда-нибудь я решусь показать свою реализацию тем, кто занимается поддержкой ext. Если они воспримут ее как нечто заурядное и незначительное — а вероятнее всего, так оно и случится, — я должен буду принять свое поражение. Однако если к моему коду отнесутся как к творческому продукту, они могут даже внедрить некоторые функции в ext4. Другими словами, все, что я могу сделать, — это усердно работать, а провозглашать свою работу творческой — не мой удел.

<sup>1</sup> Изначальное эксклюзивное право Бога на творчество подчеркивается средствами самих языков: creation — create — Creator, творчество — сотворение — Творец, создать — созидать — Создатель. — *Примеч. ред.*

## 1.2. ЗАЧЕМ ВОООЩЕ НУЖНО ТВОРЧЕСТВО

Звучит уныло, не правда ли? Зачем утруждать себя чтением книги о том, как стать креативным программистом, если все упирается в чью-то прихоть? А затем, что многие привычки и личностные качества, описанные в последующих главах, значительно повышают ваш *потенциал* как будущего креативного программиста.

Тем не менее это не отвечает на вопрос «зачем». Зачем мучить себя попытками стать креативным программистом, если вы и так компетентный специалист? Ответ и здесь лежит во многих плоскостях. Рассмотрим основные причины, по которым стоит вести жизнь креативного разработчика.

Первой причиной, говоря по-простому, является то, что таково желание работодателей. Уже многие годы объявления о приеме на должность разработчика ПО содержат слово «креативный»<sup>1</sup>. Всем известно, что описания вакансий пестрят ничего не значащими словами, которые напихивают в них сотрудники отдела кадров, стремясь привлечь как можно больше кандидатов. В наши дни в тренде находятся *мягкие*, или *гибкие, навыки* (*soft skills*). Вместо того чтобы просматривать объявления, мои коллеги и я провели собственное исследование, просто задав вопрос экспертам в области разработки ПО: «Какие нетехнические навыки, по вашему мнению, требуются, чтобы быть блестящим разработчиком?»<sup>2</sup> Угадайте, какое слово мелькало в ответах. В общем, если вы хотите продать себя, нужно быть креативным.

**УПРАЖНЕНИЕ** В каких случаях вы расцениваете свою работу программиста как творческую? В каких случаях — как какую угодно, только не творческую? В каких случаях вы расцениваете чужой код как творческий? Есть ли разница? Вам может быть неприятно отвечать на подобные приземленные вопросы, потому что ответы способны повлечь за собой (не)приятные открытия.

Что же до столь высокой востребованности творческого начала, причина этого заключается в решении проблем. Когда обычные методы не срабатывают, к успеху может привести всплеск креативности. Знание того, как работает творческий процесс, — уже полдела. Например, если ваше веб-приложение с трудом обрабатывает тысячи запросов в секунду, нелишне будет задуматься о постановке сообщений в очередь, балансировке нагрузки, кэшировании или сопрограммах. Если никто из членов команды не предложит этого, вы,

<sup>1</sup> Judy L. Wnekooop and Diane B. Walz. Investigating traits of top performing software developers. Information Technology & People, 2000.

<sup>2</sup> Wouter Groeneveld, Hans Jacobs, Joost Vennekens, and Kris Aerts. Non-cognitive abilities of exceptional software engineers: A Delphi study. Proceedings of the 51st ACM Technical Symposium on Computer Science Education, 2020.

вероятнее всего, будете ходить кругами. Креативный программист вырывается из этого круга.

Впрочем, иногда решения проблем бывает недостаточно. Порой проблема даже не выявлена, не говоря уже о ее определении. В таких случаях типичные навыки решения проблем окажутся неэффективными: для того чтобы *увидеть* проблему, вам потребуется творческое чутье.

Когда в 1831 году Чарльз Дарвин отплыл из Плимута на корабле «Beagle» в путешествие, продлившееся пять лет, у него не было намерения связывать естественный отбор с происхождением видов: этой проблематики тогда просто не существовало. Задачей исследователей британского королевского флота было лишь нанести на карту очертания береговой линии Южной Америки. Экзотический растительный и животный мир, с которым столкнулся Дарвин и о котором он делал подробные заметки, заронил семена его теории, которые взошли через много лет после путешествия.

### Творческий продукт или процесс?

Заметьте, что, восхищаясь творческой работой, мы почти всегда восхищаемся *продуктом* — конечным результатом всех пролитых потоков крови, пота и слез. Продуктом может стать умный алгоритм или изобретенный паттерн проектирования. (Подобные вещи вызовут восхищение в основном у разработчиков ПО.) А конечным результатом может явиться и целое приложение, которое, в идеале, сочтут креативным ваши пользователи.

Однако творческим может быть не только продукт, но и *процесс*, предшествующий созданию этого продукта. Только вот процесс этот, как правило, невидим и оценить его невероятно сложно. Творческие процессы могут порождать творческие продукты. Ударение в этом предложении делается на слове «могут»: результатом вполне может явиться и катастрофа. Справедливо также противоположное утверждение: творческий продукт может оказаться результатом самого обычного процесса.

Привлечение экспертов для оценки креативности продукта называется *методом согласованных оценок* (consensual assessment technique). Это популярный термин, предложенный Терезой Амабиле (Teresa M. Amabile) в 1988 году. Когда вы будете в следующий раз смотреть передачу «America's Got Talent»<sup>1</sup>, помните, что она придерживается надежных научных методов!

Дарвин не решал проблемы — он их *выявлял*. Что можем извлечь из образа мышления Дарвина мы, программисты? Обычно мы завалены небольшими и четко

<sup>1</sup> «Америка ищет таланты» — телепроект на канале NBC. Это шоу талантов, в котором выступают различные талантливые люди всех возрастов, стремящиеся выиграть главный приз — один миллион долларов. — *Примеч. пер.*

определенными (под)проблемами, задачами в пределах своей плавательной дорожки, которые каким-либо образом должны попасть в графу «Готово». Однако, возможно, по пути мы соберем достаточно фрагментов информации, соединив которые сформулируем совершенно новый вопрос. Весьма вероятно, мы обнаружим проблему, о существовании которой наши клиенты даже не подозревали. Креативный программист и выявляет проблемы, и решает их. А к путешествию Дарвина мы вернемся в главе 6.

Другая причина, по которой стоит уделять внимание суждениям других о творчестве, — мнение ваших коллег должно быть значимым. Если вы еще не заметили, разработка ПО — это командная работа. Творчество в отрыве от окружающих бессмысленно (подробнее об этом будет рассказано в главе 3) именно потому, что это социальный конструкт. Ощущение психологической безопасности, которое возникает благодаря взаимному уважению, позволяет каждому чувствовать себя непринужденно, повышая тем самым сплоченность команды. Это открывает возможность учиться и расти самому, а также помогать учиться и расти другим.

Третья причина быть креативным состоит в том, что творчество означает *удовольствие*. Многие эксперты, с которыми мы беседовали, отмечали, что единственная причина, по которой люди работают программистами, — это возможность проявлять свои творческие способности. Креативные программисты получают от своей работы огромное удовольствие. Им нравится погружаться в нее, выходить из своей зоны комфорта, комбинировать необычные идеи, обсуждать различные подходы с другими и быть с ними на одной волне. Если говорить кратко, то креативные программисты поддаются своему *творческому порыву*. Они становятся тем *homo faber*, которого имел в виду Умберто Эко.

Многие творцы пытаются достичь бессмертия при помощи своих творческих достижений, которые могут пережить их бренное тело. Тех немногих счастливиц, кому удастся осуществить свою мечту оставить неизгладимый след в мире, почитают как истинных гениев. Нам же, программистам, работающим с технологией в высшей степени изменчивой, наверное, лучше обуздать свои устремления прославить собственное имя в веках. Уверен, что к тому времени, когда эта книга увидит свет, десятки ныне существующих книг о программировании спокойно переедут на полку для «винтажа». А мы все прекрасно знаем, что это означает.

## 1.3. РАЗНЫЕ УРОВНИ ТВОРЧЕСТВА

Возможно, вы заметили, что, говоря о творчестве, я время от времени употребляю слово «гений». Конечно, для того, чтобы творить, не нужно быть гением.

Пытаясь классифицировать разные уровни творчества, исследователи вывели следующую таксономию<sup>1</sup>:

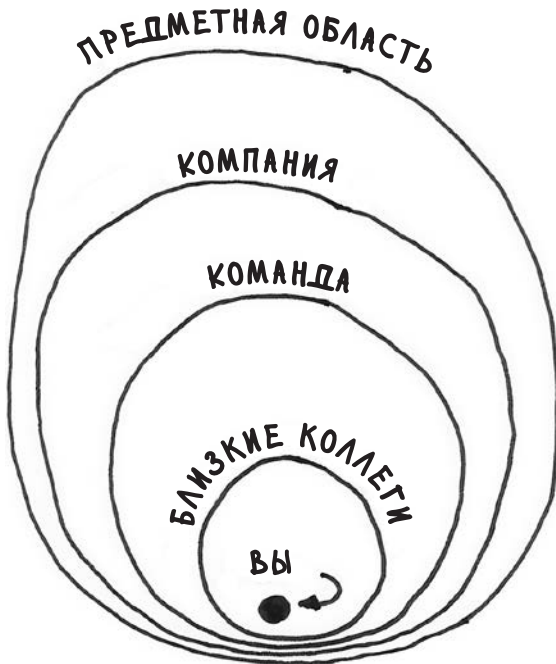
- *Творчество с маленькой буквы «т», или повседневное творчество.* Это создание в масштабах одной личности: когда делаешь что-то новое, такое, чего не делал раньше, например, кросс-компиляция вашей реализации любимой игры на C++11 для игровой консоли Game Boy Advance.
- *Творчество с большой буквы «Т», или выдающееся творчество.* Это означает делать что-то новое, такое, чего не делал никто и никогда, например портирование Ruby 3 для запуска на 486-й машине под управлением DOS 6.22. Кстати, хорошая идея...

Линус Торвалдс — Творец с большой буквы. Он полностью изменил предметную область, связанную с операционными системами (а также с контролем версий). Как утверждают некоторые ученые, значимые творческие произведения, меняющие целые области знаний, являются плодами труда гениев. А вот разработка творческого решения проблемы по обработке запросов в вашем веб-приложении вряд ли потрясет мир.

Как и всё в нашем мире, эта таксономия не могла не подвергнуться критике. Творчество с маленькой буквы «т» часто выставляется как ничем не примечательное и банальное. Величие же Творчества с большой буквы «Т» может заставить творцов уступить неизбывной потребности создать нечто, способное изменить мир. Исследователь творческих способностей Марк Ранко (Mark Runco) полностью отвергает различия между большим и малым, заявляя, что реальность не подлежит категоризации<sup>2</sup>. Другие в ответ разрабатывают собственные теории: так, существуют Н-творчество (historical — историческое: войдет ли изобретение в учебники истории), Р-творчество (personal — личное), а также не столь очевидные слои между маленькой «т» и большой «Т», называемые «творчество с уменьшенной буквы “т”» и «профессиональное творчество с маленькой буквы “т”». Некоторые исследователи, такие как Михай Чиксентмихайи (Mihaly Csikszentmihalyi), проводят интервью с гениями творчества, чтобы вычленил из этих бесед практические методы для творцов в повседневности, в то время как другие утверждают, что это дает искаженную картину. Короче говоря, в научных исследованиях творческих способностей царит беспорядок. Тем не менее существенную пользу может принести представление о творчестве как о системе различных «уровней», что показано на рис. 1.2.

<sup>1</sup> Peter Merrottsy. A note on big-C creativity and little-C creativity. Creativity Research Journal, 2013.

<sup>2</sup> Mark A. Runco. «Big C, little c» creativity as a false dichotomy: Reality is not categorical. Creativity Research Journal, 2014.



**Рис. 1.2.** Пример ближних кругов, в которых работает программист. Код, который признали креативным близкие коллеги, передается на рассмотрение дальше и получает хвалебные отзывы от команды. Однако в другой команде могут достичь того же, и тогда на уровне компании вашей славе придет конец. Поскольку творчество имеет социокультурную зависимость, смена команд меняет и интерпретацию творчества. Помнить об этих кругах может оказаться весьма полезным. Всемирно распространяя информацию, вы, безусловно, поможете своей команде и компании, но начинать нужно с себя

## 1.4. ДОРОЖНАЯ КАРТА ПУТИ К КРЕАТИВНОСТИ

Эта книга не о том, как стать гением, и тем более не о «генах творчества»: скоро вы узнаете, что их просто не существует, скорее она посвящена процессу решения проблем. Надеюсь, что, применяя различные творческие методы и новый взгляд на творчество, аккуратно вписанные в семь разных, но тесно переплетенных тем, вы станете еще более сильным программистом. Если же вы не программист, не переживайте: вы увидите, что многие из этих методов можно с легкостью перенести в другие предметные области.

Книгу Энди Ханта «Pragmatic Thinking & Learning» открывает нарисованная вручную ментальная карта (она же майндмэп), которая может служить также дорожной картой нашего проекта. Поскольку в своей книге Энди тоже апеллирует к неочевидной стороне программирования, я позволил себе на волне

вдохновения, вызванного этим рисунком, использовать его, чтобы внести в исследование<sup>1</sup> элемент оживления, — это было признано весьма творческим шагом и сразу же принято. Майндмэп, как показано на рис. 1.3 и на внутренней стороне обложки этой книги, служит также путеводителем по ней. Каждое «щупальце» на этой карте представляет главу об определенной теме, связанной с творчеством.

**ПРИМЕЧАНИЕ** Все иллюстрации в этой книге нарисованы мною от руки, в полном соответствии с темой творчества.

### 1.4.1. Семь составляющих креативного программирования

Вот приключения, которые нас ждут:

#### Технические знания

Все, кто создает творческий продукт, должны иметь четкое представление о состоянии дел в своей предметной области. Это может показаться настолько очевидным, что тратить на аргументацию целую главу было бы слишком. Однако программист не может быть креативным, если не будет в первую очередь программистом. Само собой разумеется, что любому делу должно предшествовать обучение. Но остановиться и поразмышлять о различных способах получения информации, постоянного обучения, осознания когнитивных искажений и управления знаниями тоже бывает очень полезно.

Креативный программист понимает, как преобразовать непрерывный поток знаний в новые идеи.

#### Сотрудничество/общение

Творчество не вершится в изоляции: оттачивание идей — процесс социальный. Без какой-либо обратной связи было бы невозможно возвести идею с легким налетом оригинальности в ранг превосходной. Подтолкнуть вас к изменениям могут и ваши коллеги. В главе 3 мы рассмотрим концепцию кластеров гениев, узнаем, как собирать команды мечты, а также познакомимся с методиками повышения креативности команд. В статье, опубликованной моими коллегами и мной<sup>2</sup>, эта тема названа «общением» (communication), поэтому мы будем придерживаться этого термина, хотя, оглядываясь назад, можно сказать, что более подходящим могло бы оказаться слово «сотрудничество» (collaboration).

---

<sup>1</sup> Wouter Groeneveld, Laurens Luyten, Joost Vennekens, and Kris Aerts. Exploring the role of creativity in software engineering. 2021 IEEE/ACM 43rd International Conference on Software Engineering: Software Engineering in Society.

<sup>2</sup> Wouter Groeneveld, Hans Jacobs, Joost Vennekens, and Kris Aerts. Non-cognitive abilities of exceptional software engineers: A Delphi study. Proceedings of the 51st ACM Technical Symposium on Computer Science Education, 2020.

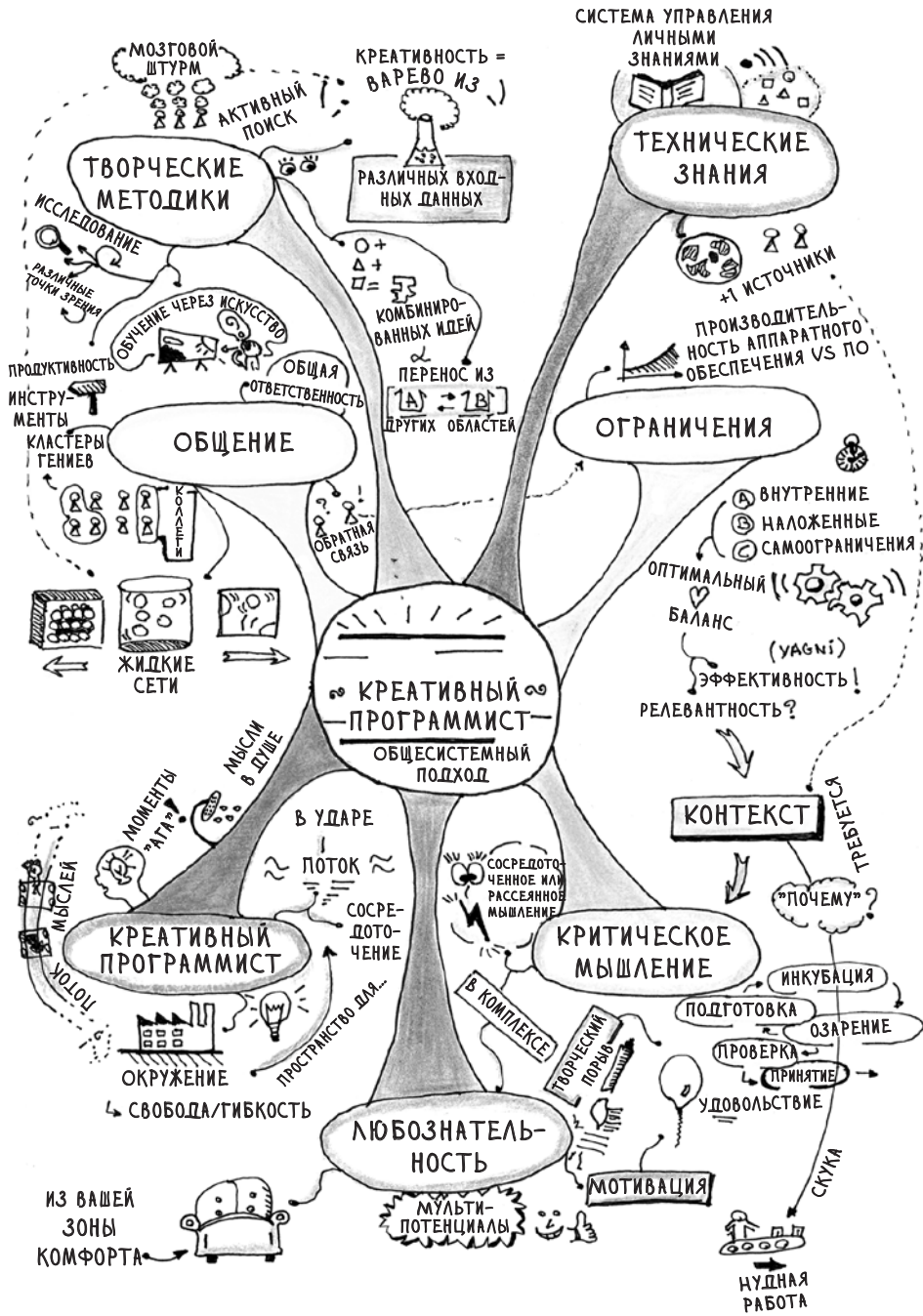


Рис. 1.3. Майндмэп книги «Креативный программист»