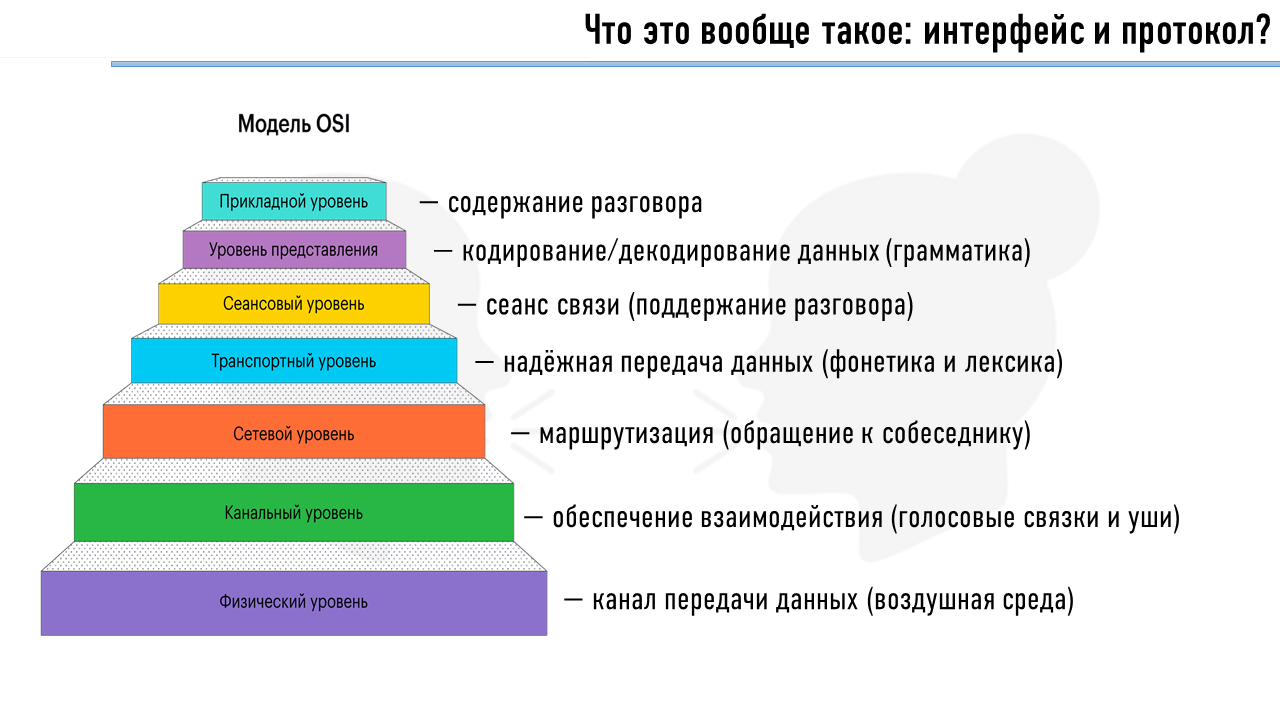
Интерфейс — это механизмы, которые обеспечивают нам возможность взаимодействия (передачи информации). Протокол — это договорённости, которые реализуют это взаимодействие с использованием данных механизмов.

Что есть интерфейс в устной речи? Из аудитории звучит «голосовые связки», но на самом деле — в первую очередь это воздушная среда (или, в более общем случае, другая упругая среда), потом уже органы дыхания, а потом уже голосовые связки. Этого набора в принципе достаточно для формирования звуков — но пока что эти звуки не имеют смысла, они не передают информацию.

Для передачи информации мы используем протокол — и что характерно, если интерфейс у всех одинаковый, то протоколы крайне изменчивы. В протокол устной речи входят:

* фонетика — то есть звучание букв;
* лексика — то есть построение из этих букв слов;
* грамматика — то есть сборка из отдельных слов целых фраз;
* социальные нормы.

Последний пункт не называет практически никто и никогда — кроме профессиональных лингвистов, для которых он очевиден. Но на самом деле — это тоже крайне важный элемент протокола: если я буду разговаривать с вами вне принятых для данного контекста социальных норм (например, перейду на мат), наше взаимодействие не сложится (например, меня забанят).



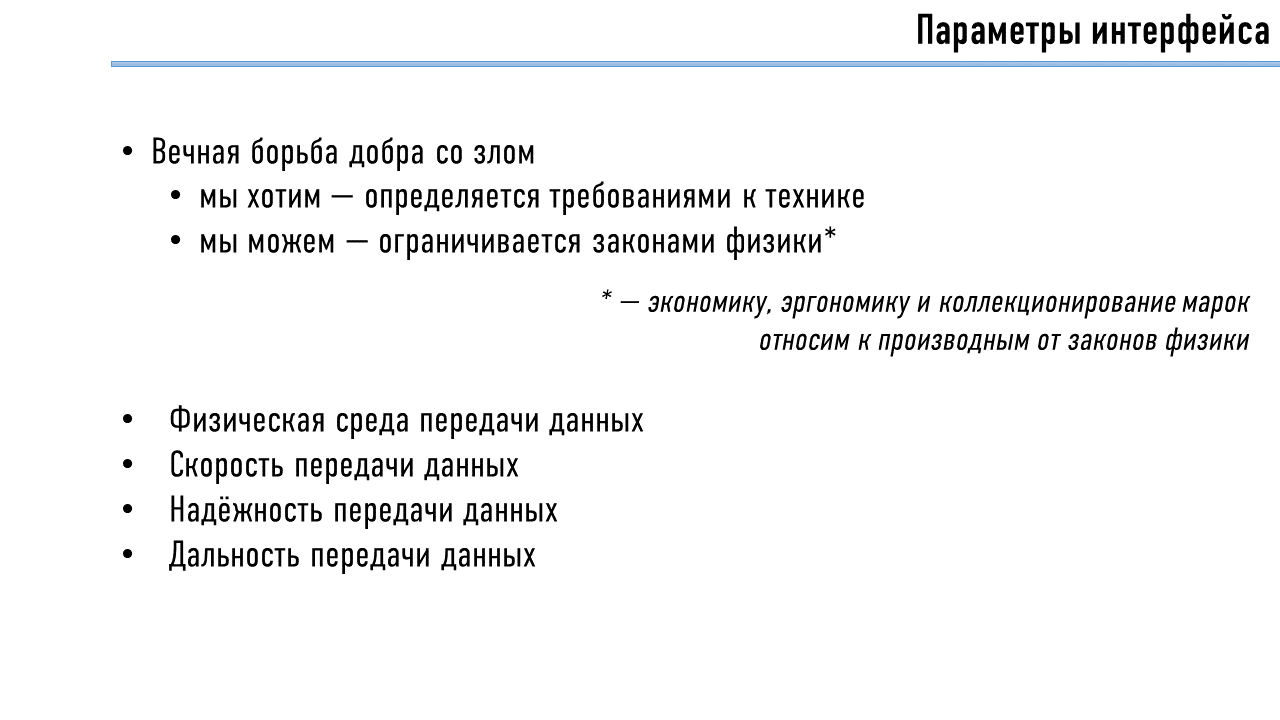
Пока что это всё кажется тривиальным, но сделаем один шаг дальше — положим наше устное взаимодействие на модель OSI.

Да, сейчас придут хардкорные айтишники и расскажут, что модель OSI давно ничего не описывает, потому что вот у нас тут Wireguard поверх PPPoE поверх Wi-Fi (реальный случай, напишу наверное тоже статью на днях) поверх чёрта в ступе, и как вы это в вашу пирамидку... Но — как и любые простые представления, модель OSI может не описывать всю сложность окружающего мира, зато практически незаменима для того, чтобы сколь-нибудь наглядно представить себе его базовое устройство. Ньютоновская механика тоже, знаете ли, реальный мир описывает так себе, но отменять её пока никто не планирует.

Итак, как можно попробовать разложить устную речь в модель OSI?

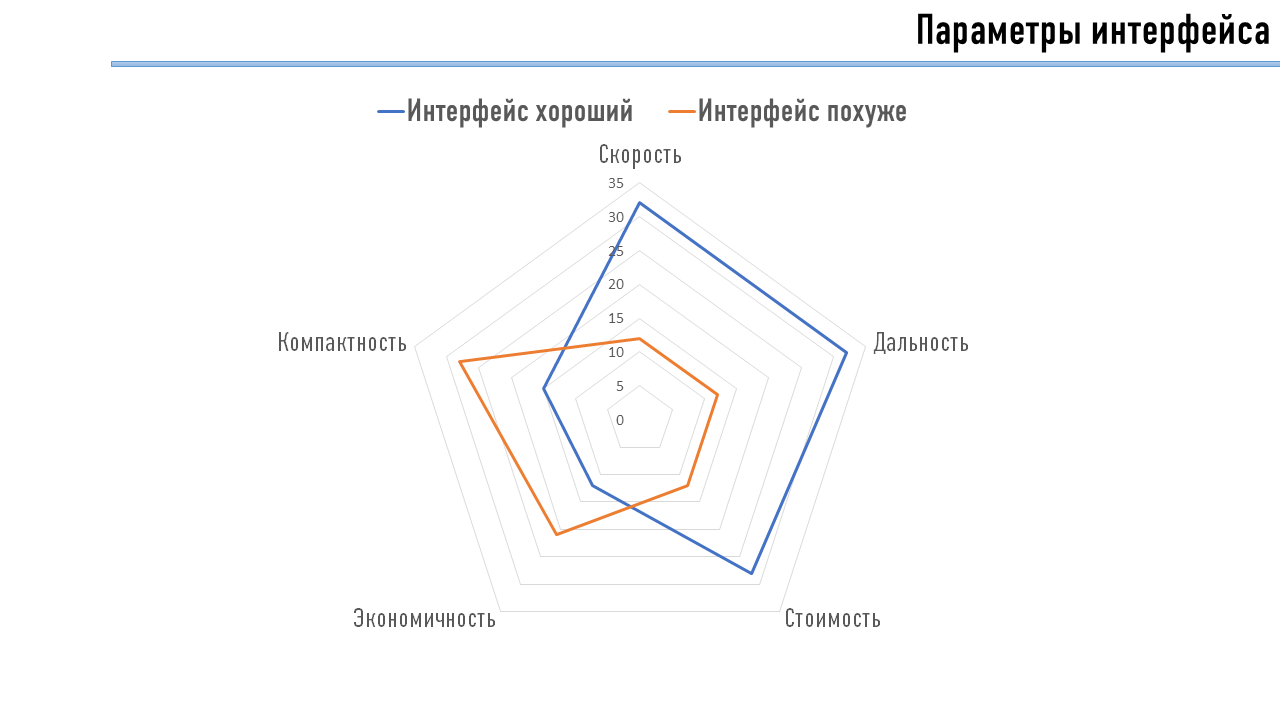
* физический уровень — ну, это очевидно, это упругая среда, как правило воздух, в космосе никто не услышит твой крик;
* канальный уровень — голосовые связи (с одной стороны канала) и уши (с другой стороны), они обеспечивают взаимодействие между двумя людьми по акустическому каналу;
* сетевой уровень — здесь всё становится интереснее, но можно вспомнить, что в него должна попадать маршрутизация, а что у нас есть маршрутизация, если не обращение к собеседнику? Это, кстати, не обязательно произнесение его имени, это и поворот головы в его сторону, и невнятное мычание, и «мальчик жестами показал, что его зовут Хуан». В общем, маршрутизация в устном общении очень рудиментарная, но нельзя сказать, что её нет.
* транспортный уровень — обеспечивает надёжную передачу данных. В устной речи это, пожалуй, фонетика и лексика. «Нрзбр» в транскрипте — это когда сломалось либо то, либо другое, и слово превратилось в набор звуков.
* сеансовый уровень — и тут аудитория угадала с первой попытки, сеансом в речи можно считать разговор. В речи нет явных признаков «этого разговора», но есть много неявных — максимально допустимая задержка между фразами, например, непрерывное нахождение собеседеников в поле зрения друг друга и так далее.
* уровень представления — пожалуй, что грамматика. Если в компьютере уровень представления — это, например, формат файла, который обе системы должны понимать одинаковым образом, то в речи — построение фраз. «Буквы знаю, смысла не понял» — это как раз случай нестыковки на уровне представления, могущего возникнуть как по причине шизофазии у одного из собеседников, так и в случае, например, попытки объяснить номотетический метод немецкого неокантианства человеку, который не очень уверен, что такое «категорический императив».
* прикладной уровень — очевидно, само содержание разговора.

Таким образом немного развлекшись и повысив пластичность мозга, приступим к чуть более формальной части.



Любой интерфейс описывается неким набором технических параметров, всегда фундаментально ограниченных снизу нашими хотелками (интерфейс, не решающий поставленную задачу, очевидно, не нужен), а сверху — законами физики.

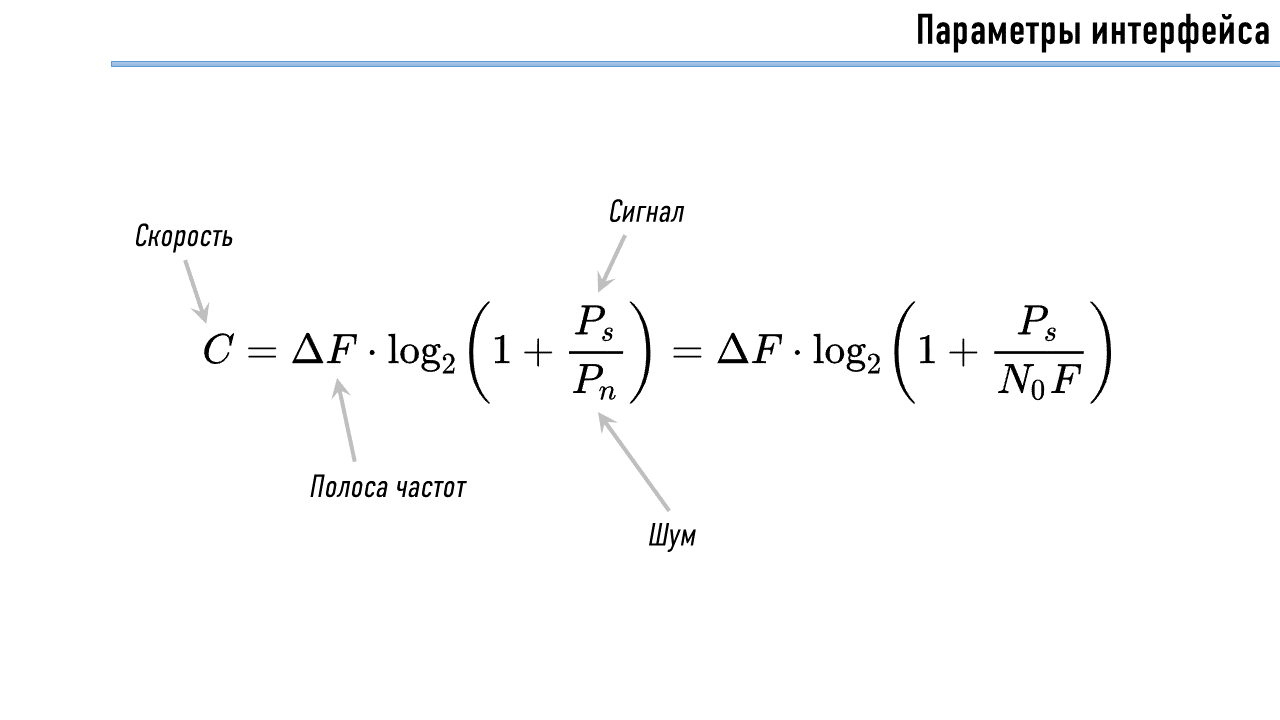
Но при этом очевидно, что не все интерфейсы равны — какой-то хуже, какой-то лучше. Но как именно мы будем это хуже-лучше? Быстрее, но прожорливее — это лучше или хуже?



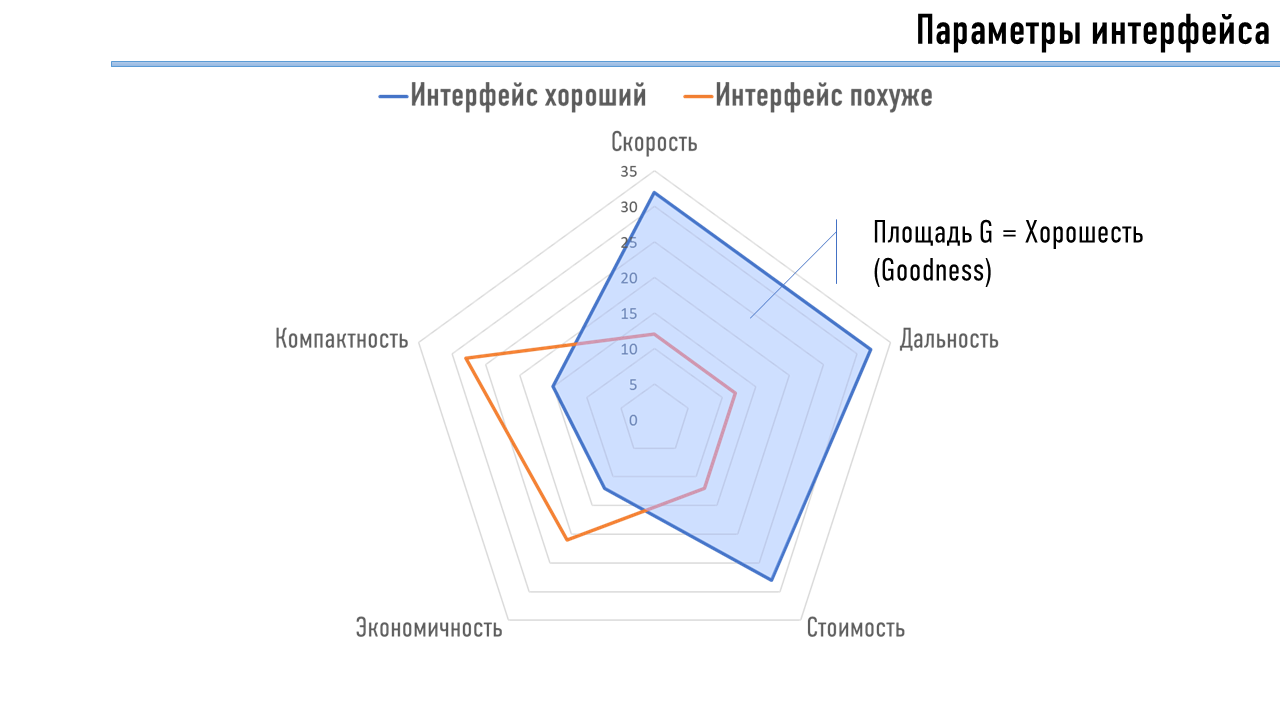
Крайне полезно не просто выписать значимые параметры интерфейса, а нарисовать их на лепестковой диаграмме.

Второй крайне полезный момент — представить себе контур такой диаграммы как резинку (возьмите аптечную или резинку и растяните её на пальцах). Хотите увеличить один параметр, не ухудшая другие? Вам придётся приложить больше усилий, потому что резинка сопротивляется. Не хватает усилий? Значит, чтобы ещё сильнее увеличить один параметр, придётся уменьшить другие.

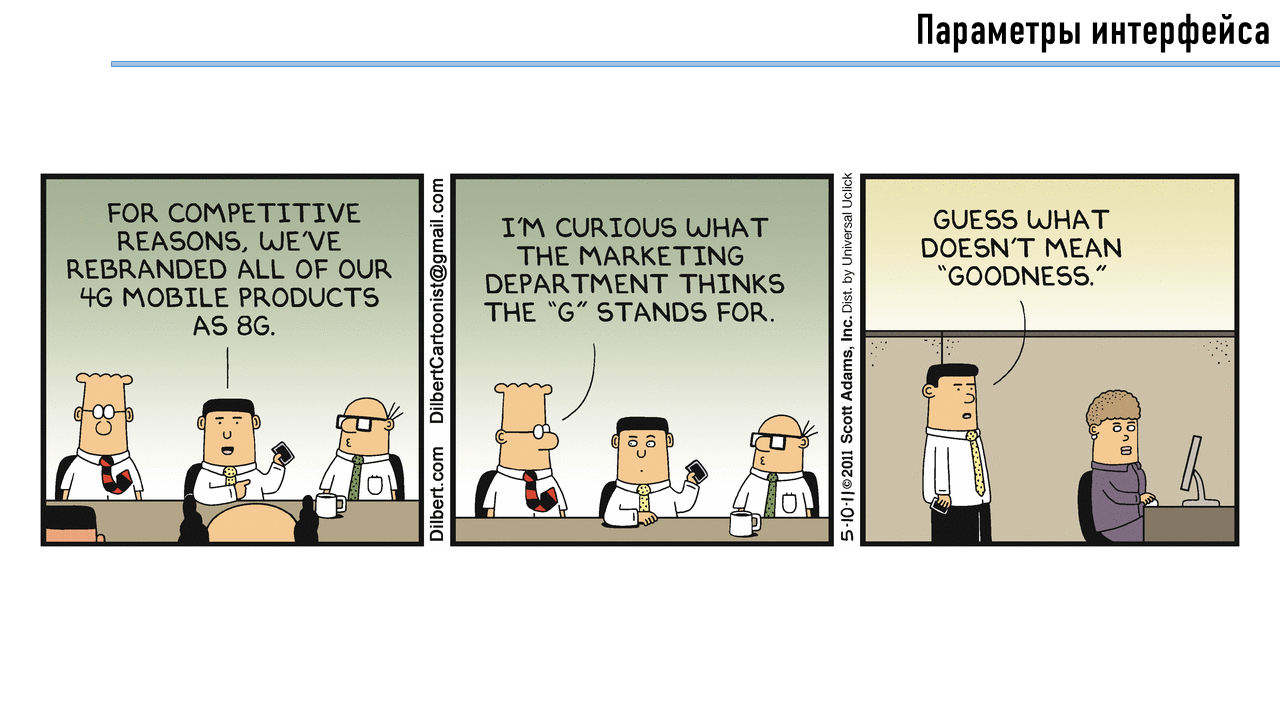
Это момент, который не понимают даже многие начинающие проектировщики, что уж говорить даже про многоопытных интеграторов, — all magic comes with a price. Если вы смогли улучшить один параметр, не ухудшая другие — значит, ваш новый интерфейс объективно лучше старого. Но рано или поздно вы придёте к ситуации, когда невозможно улучшить одно, не ухудшая другого. Неизбежно.



Ну, например, прискорбно малоизвестная формула предела Шеннона на скорость передачи данных в канале с шумами. Это — физическое ограничение, верхний предел, к которому нужно стремиться, но невозможно превзойти. При заданном уровне сигнала (то есть вбухиваемой в канал мощности, то есть энергопотреблении передатчика) и заданной полосе частот скорость может быть только такая. Хотите больше? Можно, но надо, например, вложить большую мощность — а значит, пожертвовать экономичностью устройства.



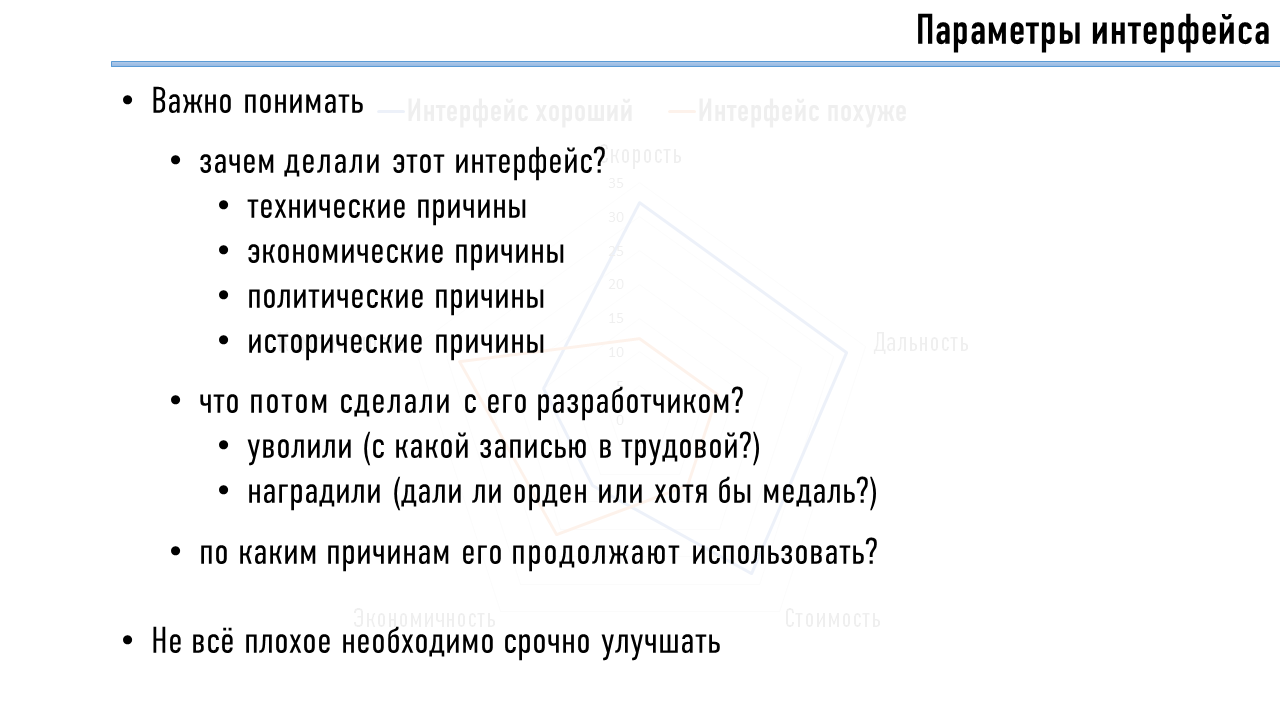
В принципе, уже очевидно, как правильно оценить «хорошесть» интерфейса: по *площади*, которую занимает диаграмма его параметров. Любой интерфейс, который занимает меньшую площадь, менее оптимален — его разработчиками не выбраны доступные физические пределы на его параметры.



Нежно люблю эту картинку — тем более, что она отражает некоторые реальные моменты из моего опыта взаимодействия с маркетинговым отделом.

И она же, если посмотреть на неё с другой стороны, а это вообще полезное упражнение — побыть адвокатом дьявола — на самом деле не так уж однозначна. Потому что маркетинговый отдел, внезапно, тоже может иметь свои причины так обращаться с продуктом — которые может не знать инженер (да, дальше маркетинг действует неправильно, но кто, на минуточку, сказал, что инженеры всегда действуют правильно?).

И эти моменты крайне желательно понимать — не только потому, что они могут повлиять и повлияют на вашу работу (если вы не планируете работать в гордом одиночестве), но и помогут вам правильно понять, почему то или иное решение было сделано в прошлом.



Диаграммы выше описывают чисто технические параметры интерфейса. Но причины, по которым его сделали именно таким — в том числе и не обязательно идеальным — могут быть довольно разнообразны:

* технические причины — здесь всё понятно, это имеющиеся у нас технические пожелания и физические ограничения, которые пришли в какой-то компромисс;
* экономические причины — это не просто стоимость реализации какого-то технического решения, но и ожидаемые доходы от его использования. Хороший пример — дорогой в реализации Lightning, от которого Apple при этом отказывается очень неохотно, потому что... да потому что он принёс Apple миллиарды на продаже аксессуаров, производство которых даже сторонними производителями компания может контролировать.
* политические причины — конкретное решение может даже не приносить денег напрямую, но при этом формировать определённый образ компании («инновационная», «отечественная», и так далее, возможных эпитетов десятки). Отказ от проприетарного решения в пользу стандартного — это часто как минимум одна волна негативных комментариев в стиле «прогнулись, смотрите, они больше не инновационные, теперь у них как у китайцев», и даже это при принятии решений учитывается.
* исторические причины — это, в первую очередь, сложившаяся практика использования решений компании. Например, недавно в одной статье на Хабре проскальзывало удивление, что Siemens годами не модернизирует свои ПЛК, их внешний вид и интерфейс такой же, как двадцать лет назад — но задумывался ли автор, что может быть, Siemens обладает некоторой информацией, побуждающей его делать иметь так? Например, что заказчики между современным интерфейсом и отсутствием необходимости переучивать эксплуатационщиков на новый продукт выбирают второе?

Более того, не менее важный момент — а кто вообще вам сказал, что какое-то решение является удачным, оптимальным, оправданным? Сам факт его использования ещё не означает ничего из этого.

Разработчика этого решения наградили или уволили? Если уволили, то с какой формулировкой? Не осталось ли оно в использовании в основном потому, что уже не было бюджета и времени на его переделку (экономический фактор), было решено не докладывать заказчику о необходимости пересогласования ТЗ (политический фактор), а в последующие годы требовалось поддерживать обратную совместимость с ним (исторический фактор)?

Я очень часто встречаю инженеров, которые очень стремятся что-то улучшить, но редко — задающихся вопросом о влиянии нетехнических факторов, и следовательно, о том, улучшат ли они *ситуацию в целом*, улучшив *техническое* решение.



Как пример развития интерфейса и протокола, возьмём такую штуку, как питание по USB.

Интерфейс USB зародился в середине 1990-х годов как универсальная замена существовавшему тогда зоопарку из RS232/COM, LPT, SCSI, PS/2 и чёрт знает чему ещё. Он обеспечивал передачу данных на достаточной для большинства применений скорости, а также подачу на периферийные устройства питания с напряжением 5 В и током до 500 мА.

Никто вообще не рассматривал в тот момент USB как интерфейс электропитания. Наличие в нём напряжения было чисто вспомогательной функцией для мелкой периферии, которой по определению не надо было много ватт и которая по определению в любом случае общалась с компьютером-хостом — то есть полноценная реализация протокола передачи данных в ней была.

Соответственно, для получения своего устройство сразу после подключения начинает инициализацию интерфейса, в ходе которой отправляет хосту дескриптор bMaxPower, в котором указан желаемый им ток с шагом 2 мА, в однобайтовой переменной. До ответа хоста устройство имеет право потреблять в пределах 100 мА, после ответа — сколько попросило, если хост столько разрешил.



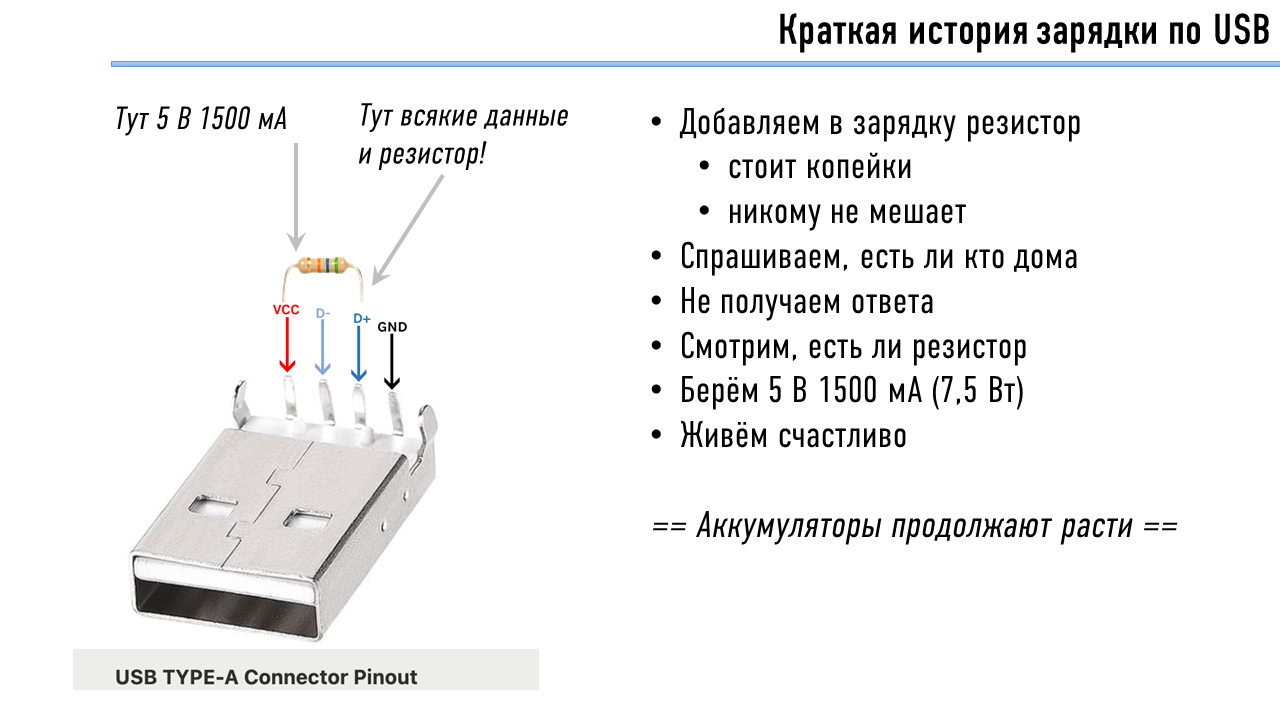
Но потом появились смартфоны.

Galaxy SII тут, конечно, чисто для примера, началось всё раньше. Началось с появления устройств, у которых разъём USB, на тот момент — microUSB, использовался как для обмена данными с хостом, так и для зарядки аккумулятора устройства. И аккумуляторы эти стали быстро расти в ёмкости.

Во-первых, возник вопрос зарядных устройств, не являющихся компьютером, то есть — хостом. Поддержка с их стороны обмена данными по протоколу USB сильно усложняла и удорожала их конструкцию (экономическая причина), соответственно, о какой-то инициализации и обмене дескрипторами речи не шло.

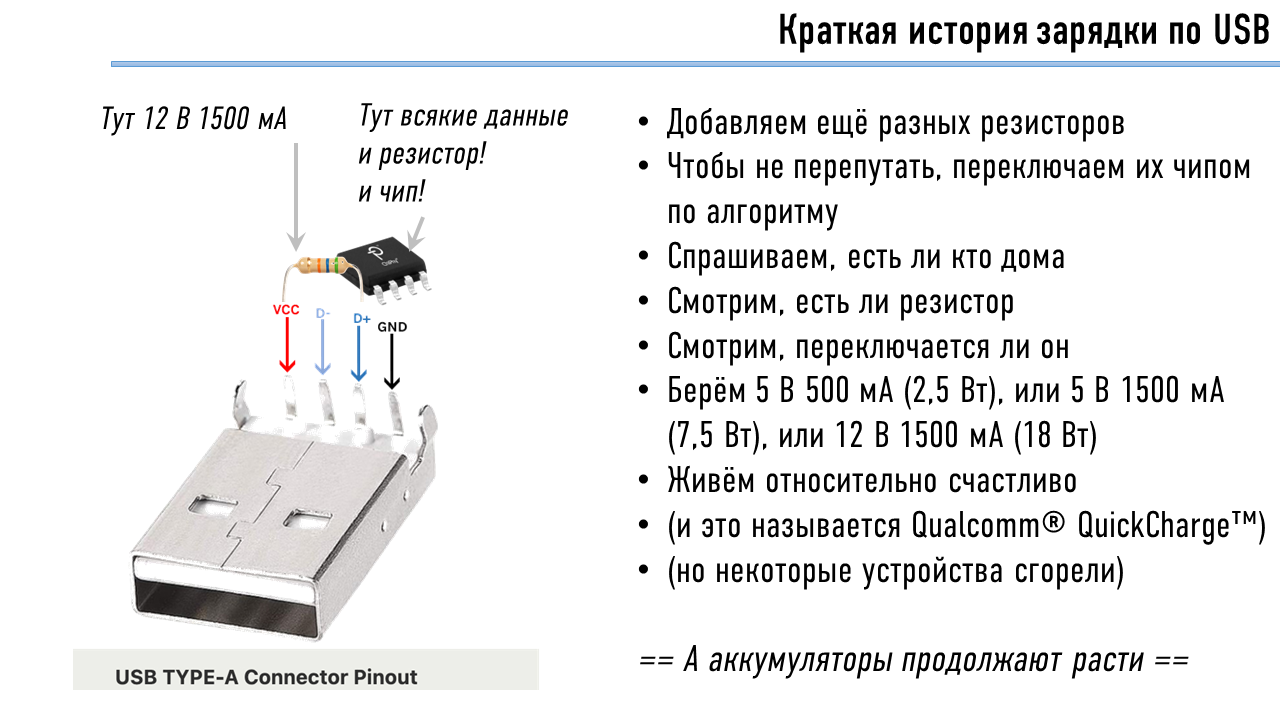
Во-вторых, возникла потребность в больших мощностях данных зарядных устройств, позволяющих сократить время зарядки аккумулятора — благо что со стороны смартфона технических препятствий здесь не было, литиевый аккумулятор довольно быстро может набрать первые 50-70 % ёмкости.

Но это привело к следующей проблеме — а как, собственно, смартфон поймёт, что можно от зарядки потребить больше 500 мА? И сколько конкретно можно-то?



В Samsung сделали просто — повесили на сигнальные линии USB обычные резисторы. Резистор стоит копейки (буквально), а алгоритм получился простой и вроде бы надёжный: смартфон, увидев питание на шине USB, пробует достучаться до хоста, если ответа нет — значит, это не хост, а зарядка, дальше смотрим, какие уровни напряжения стоят на D+/D- и делаем выводы о том, какой ток зарядка может отдать. Нет ответа и нет резисторов — 500 мА, есть резисторы — смотрим по их комбинации из нескольких возможных вариантов.

Один нюанс — аккумуляторы-то продолжают расти. И 7,5 Вт уже не выглядят серьёзной мощностью, а просто поднять ток — чревато перегрузкой разъёма и кабелей, банальным их подгоранием.



И тут приходит компания Qualcomm и предлагает — а давайте увеличивать напряжение! Так при том же токе мы сможем поднять мощность, а уж 5, 9 или 12 В — кабелям вообще без разницы, изоляция там по определению на напряжения на порядок больше сделана. При этом уже зарядка должна понять, какое напряжение можно подавать на устройство, то есть задача внезапно разворачивается на 180 градусов.

Но делать это мы будём по-прежнему дёшево, потому что на дворе вообще-то 2015 год, и встраивать в зарядку какие-то серьёзные протоколы, а уж тем более, довольно сложный в реализации протокол USB — по-прежнему сложно и дорого.

Так возникают по сути те же самые резисторы, но теперь их выставляет не зарядка, а устройство — показывая, сколько вольт оно готово получить. Зарядка же, пользуясь достаточно несложным (то есть дешёвым и простым) чипом, выясняет, сколько вольт надо подать на устройство.

Что мы видим на этом этапе? Да весь спектр причин возникновения именно такого протокола:

* технический — есть физические ограничения интерфейса, которые надо как-то обойти;
* экономический — надо сделать это дёшево, поэтому наиболее прямолинейный путь, заключающийся в использовании линий данных интерфейса по их прямому назначению, не подходит по деньгам;
* политический — конечно, каждый производитель хочет эти новые возможности сделать конкретно под себя;
* исторический — менять разъём USB на что-то другое нереально, только-только все наконец на USB-то переехали (даже эксперименты с microUSB 3.0 на смартфонах успеха не имели, с ним вышли единичные модели, а ведь он хотя бы был полностью обратно совместим с microUSB 2.0).

При этом QuickCharge тоже, с очевидностью, не был панацеей от всех бед. Во-первых, он по-прежнему принадлежал конкретной компании, хоть и готовой лицензировать его на сторону (и тут параллельно QC родилась ещё пачка различных альтернативных протоколов быстрой зарядки, например, Oppo VOOC). Во-вторых, универсального протокола внутри QC по-прежнему не было — была лишь договорённость о том, что несколько конкретных уровней постоянного напряжения на линиях данных разъёма USB сигнализировали о нескольких конкретных уровнях напряжения зарядки.

В-третьих, судя по всему, китайцы довольно быстро пренебрегли в своих QC-зарядках требованиями к обязательному хендшейку — и начались случаи, когда такая зарядка встречалась с устройством, у которого почему-то на D+/D- были выставлены уровни, в QC кодирующие 9 или 12 В. Ну и зарядка, пренебрегшая хендшейком, в ходе которого она могла бы установить, что устройство на самом деле может питаться только от 5 В, туда ему 9 или 12 В и подавала.

Некоторые мои коллеги уже прямо запретили покупать им зарядки с поддержкой QC, потому что объяснить бойцу, что некоторые устройства нельзя включать в некоторые USB-порты (хорошо, если внешне отличающиеся хотя бы цветом), невозможно.

И помимо этого, мощность устройств продолжала упорно расти! Более того, идея о том, что устройства — любые устройства — очень удобно заряжать от USB, захватывала всё больше умов, благо что и доступная мощность уже поднялась с изначальных 2,5 Вт до весьма серьёзных 18 Вт.



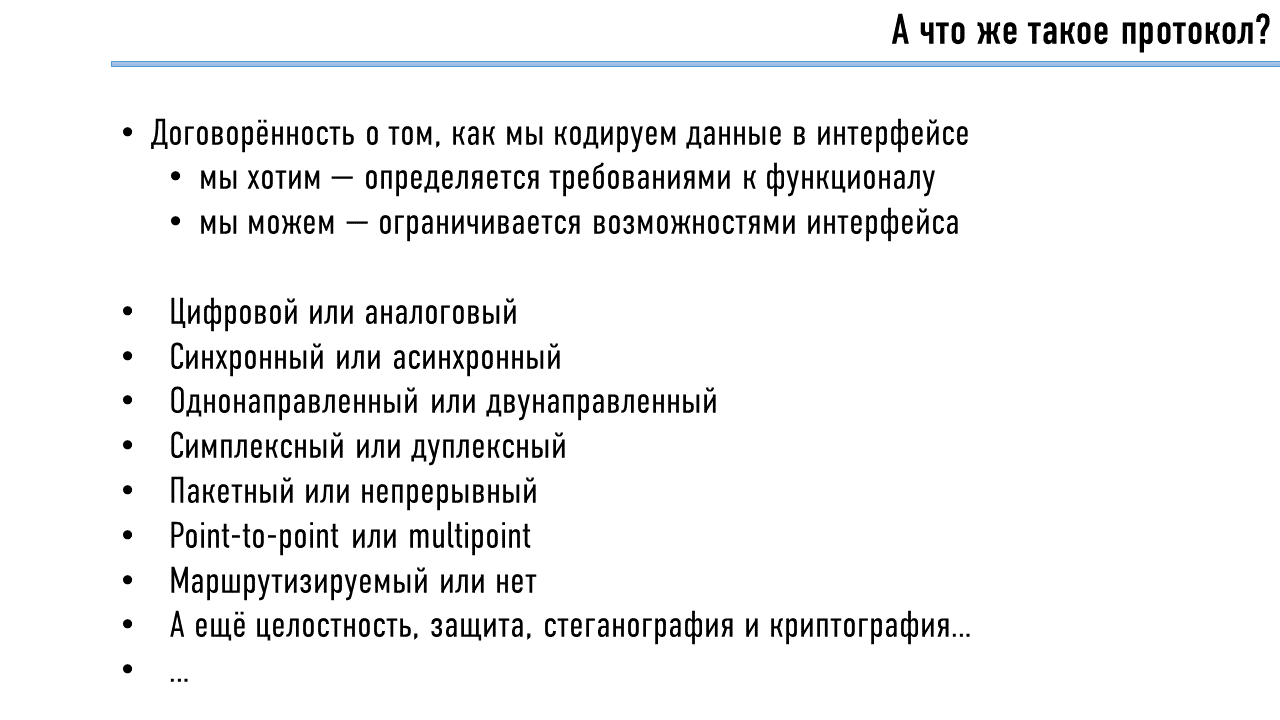
К счастью, в этот момент наконец-то случилась перезагрузка: на смену классическому USB пришёл USB-C, в разъёме которого, помимо всех прочих удобств, можно было сделать очень много ножек — и, с учётом сложившейся реальности, две из них под названием CC выделили на нормальный протокол зарядки, получивший название Power Delivery (PD).

Цифровая двусторонняя связь между устройством и зарядкой, мощности до 240 Вт (в распространённых на данный момент устройствах — до 65 Вт, но и этого уже достаточно для ноутбуков, а не только смартфонов), напряжение до 48 В, возможность точной установки как напряжения, так и тока...

Почему это стало возможным и, более того, популярным?

* технический фактор — разъём перестал быть лимитирующим фактором, так как новый разъём изначально учитывал потребность в зарядке;
* экономический фактор — для управления зарядкой были выделены отдельные линии, что незначительно увеличило цену разъёма, но зато позволило реализовать простой и дешёвый протокол обмена данными конкретно о зарядке (цифровой интерфейс на скорости 300 кбит/с);
* политический фактор — спецификации PD не принадлежали ни одному конкретному вендору, реализовывать их могли все желающие (и это далее позволило Google внести поддержку стандартных протоколов зарядки по USB-C сначала в рекомендуемые, а потом в обязательные требования к Android-смартфонам);
* исторический фактор — смена разъёма на USB-C к этому моменту уже была обусловлена многими причинами, и в целом покупателями воспринята положительно.

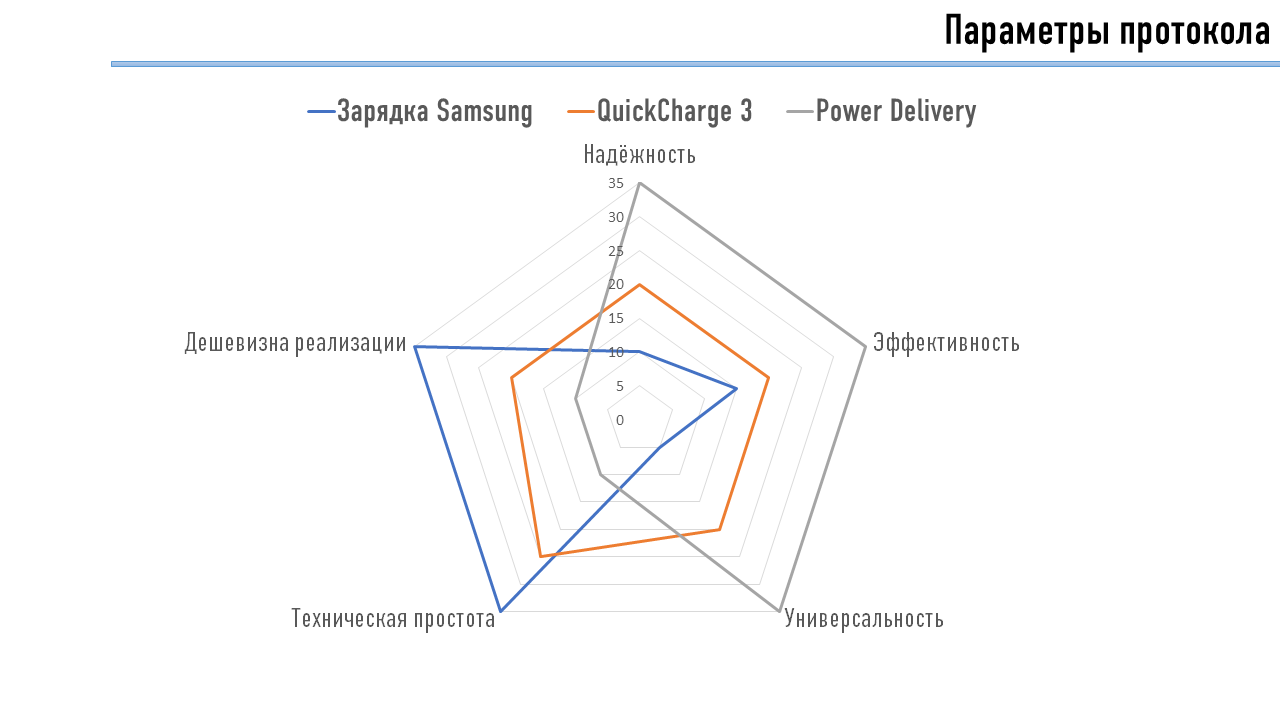
В некотором смысле, технологии развивались по спирали: история началась с bMaxPower в цифровом протоколе USB, прошла через несколько поколений сильно упрощённых аналоговых решений, а закончилась снова на цифровом протоколе USB — только уже другом протоколе и в другом USB.



Внимательный читатель уже заметил, что где-то посередине разговора мы перешли от интерфейса к протоколу, то есть договорённости о том, как мы используем данный нам интерфейс.

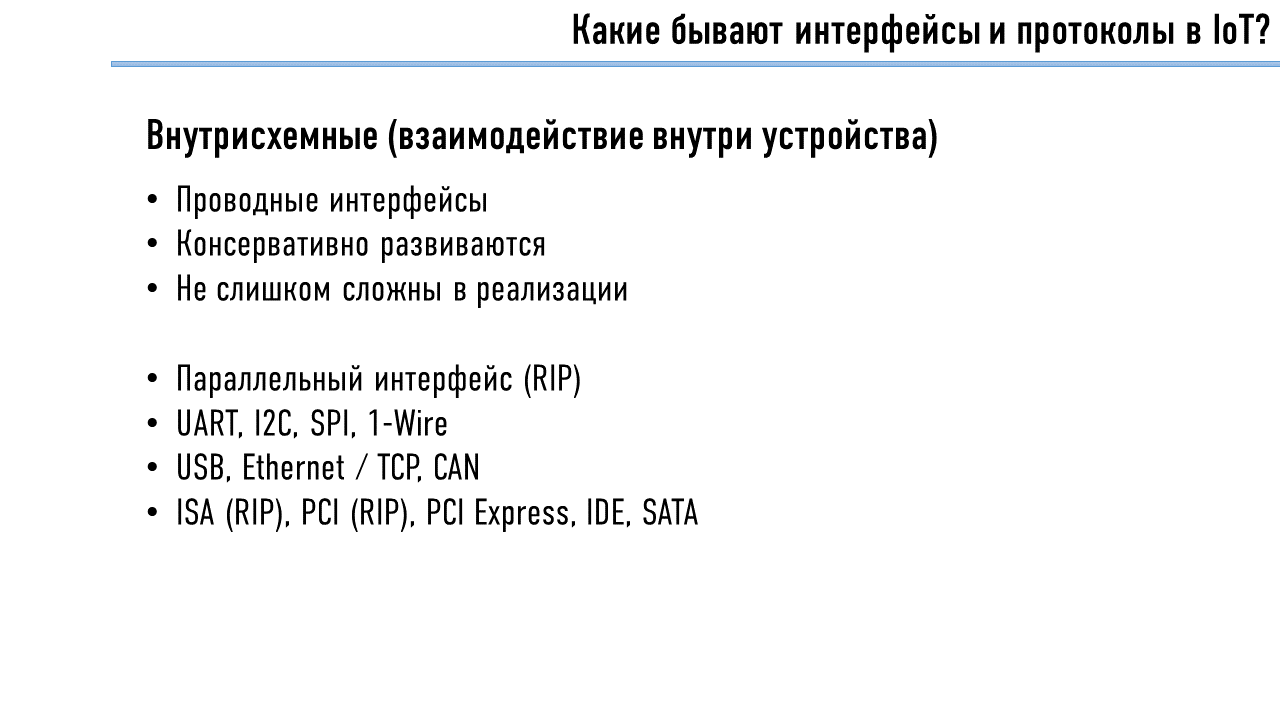
Замечу два важных момента: во-первых, интерфейс может позволять реализовать много разных протоколов, абсолютно несовместимых друг с другом. Не только USB, но и, например, банальный проводной телефон: он имеет протокол передачи номера на АТС, протокол вызова и протокол передачи голоса. Они существенно различны, не могут использоваться в один момент времени и не могут заменять друг друга — но эксплуатируют один и тот же интерфейс, двухпроводную линию связи с центральным питанием.

Во-вторых, если интерфейс ограничен требованиями физического мира, то протокол — уже возможностями интерфейса. Разъём USB-A — не вершина инженерного мастерства и уж точно не предел физически допустимого, но долгие годы именно его конструкцией и возможностями определялось развитие протоколов быстрой зарядки по USB.



При этом «хорошесть» протокола также можно прикидывать по площади на лепестковой диаграмме — только не надо забывать, что содержание этой диаграммы меняется со временем, даже если сами протоколы остаются неизменными. Двадцать лет назад «дешевизна» у Power Delivery была бы в районе нуля, а универсальность упиралась бы в нераспространённость необходимого ему разъёма USB-C.

Это гигансткое лирическое отступление в целом обозначает, что именно я планирую донести до слушателей в рамках курса. Ни в коем случае не наборы справочных характеристик различных протоколов, и даже не «давайте возьмём библиотечку и отправим данные с Raspberry Pi в облако» — я хочу донести сложность окружающего мира, пусть и в контексте только используемых в IoT протоколов и интерфейсов, научить понимать её причины и в этой сложности ориентироваться.

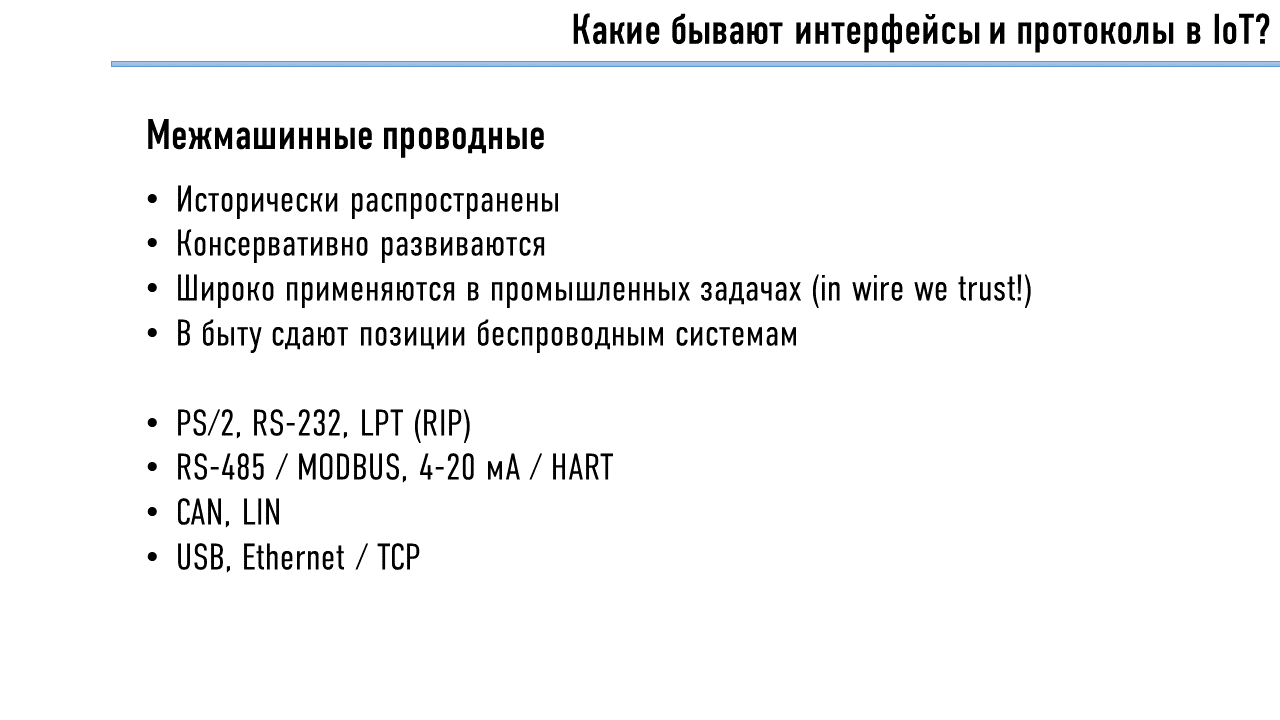


Какие вообще бывают интерфейсы и протоколы в IoT, а также какое внимание мы им уделим?

Первая группа — внутрисхемные интерфейсы, обеспечивающие взаимодействие внутри устройства. Они практически всегда проводные, крайне консервативно развиваются (старшим из них уже десятки лет, и ещё десятки лет они будут с нами), а главное — обеспечивают связь на небольших расстояниях, обычно не более десятков сантиметров.

Иногда, впрочем, в качестве внутрисхемных используются внешние интерфейсы, такие как USB, Ethernet или CAN — в зависимости от конкретных потребностей в скорости, наличия аппаратных приёмопередатчиков и т.п. При этом они могут технически упрощаться, например, интерфейс Ethernet внутри устройства может использовать без разделительных трансформаторов.

Проводным внутрисхемным интерфейсам будет посвящена следующая лекция.



Межмашинные проводные интерфейсы и протоколы — использующиеся для проводной связи между различными устройствами. Также развиваются весьма консервативно, но в быту в последнее десятилетие активно сдают позиции беспроводным интерфейсам — в промышленности, впрочем, держатся и будут держаться ещё долго, ибо in wire we trust.

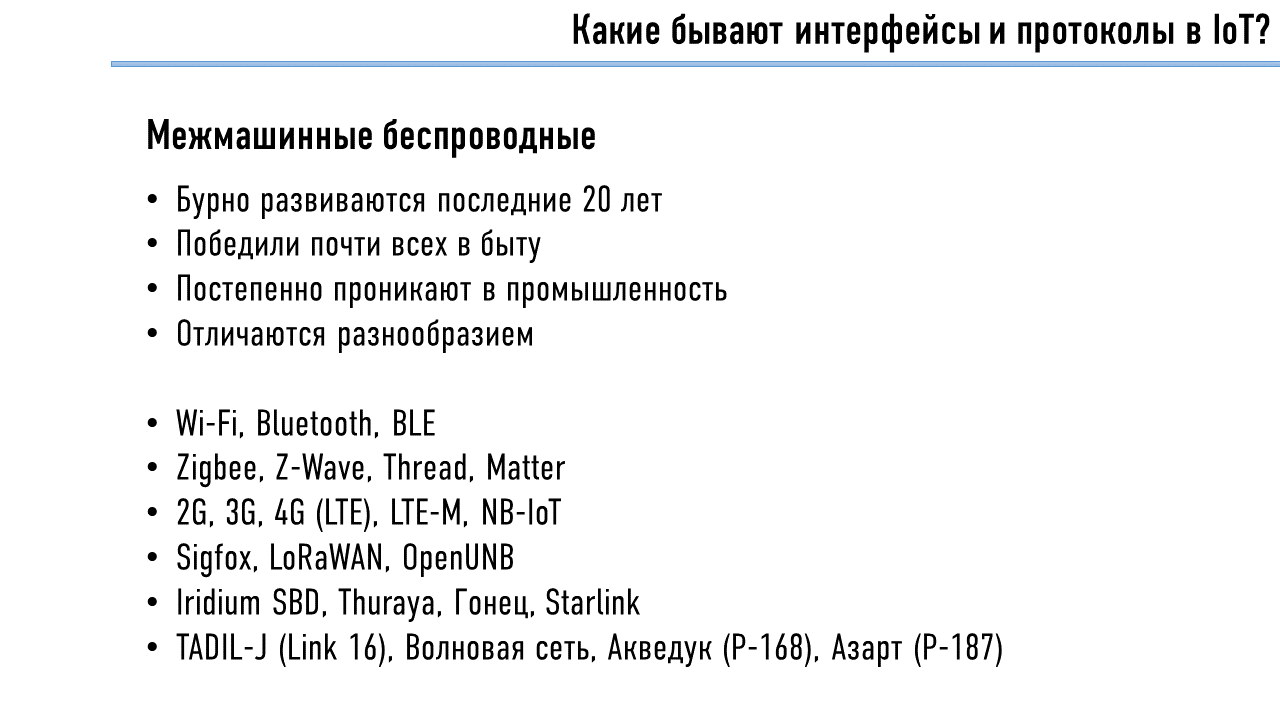
Это могут быть специализированные бытовые интерфейсы, такие как PS/2 (мыши и клавиатуры в компьютерах), LPT (параллельный порт принтера там же), RS-232 (модемы) — эта категория сейчас уже фактически полностью вымерла.

Это могут быть промышленные интерфейсы — RS-485, аналоговая токовая петля 4-20 мА и цифровой HART поверх неё.

Это могут быть автомобильные интерфейсы — CAN и LIN — сейчас уже вышедшие за пределы автомобильной промышленности.

Наконец, это современные универсальные бытовые интерфейсы, такие как USB и Ethernet.

Проводным межмашинным интерфейсам также будет посвящена одна лекция.



И наиболее интересная часть — это беспроводные интерфейсы, бурно развивающиеся последние два десятилетия, уже серьёзно потеснившие проводных конкурентов в быту и офисе, а также активно пробивающиеся в промышленное и даже военное применение.

Они крайне, невероятно разнообразны, не менее запутана и их эволюция с годами — и потому они представляют особенный интерес для изучения.

Это бытовые интерфейсы общего назначения — Wi-Fi, Bluetooth (что интересно, разрабатывавшийся в своё время как интерфейс довольно узкого применения), BLE. Это протоколы умного дома — Zigbee, Z-Wave, Thread, Matter, BLE Mesh. Это протоколы сотовой связи — от 2G/GPRS до 4G и его IoT-отпрысков LTE-M и NB-IoT. Это весьма интересные сверхэкономичные низкоскоростные протоколы дальней связи специально для IoT-систем — Sigfox, OpenUNB, LoRaWAN. Это спутниковая связь — которая, как ни странно, существовала задолго до Starlink, хотя последний и привнёс в неё новое качество. Наконец, это военные системы связи, которые также весьма своеобразны в своём развитии — но благодаря этому позволяют продемонстрировать некоторые вещи, важные и интересные, но несущественные или попросту незаметные на «гражданке» (возьмём тот же ППРЧ или широкополосную модуляцию — в некотором смысле их назначение в Bluetooth и «Азарте», в LoRaWAN и «Волновой сети», конечно, схоже...).

Именно беспроводные протоколы и займут оставшиеся лекции. Как они эволюционировали, зачем создавались и почему умирали, какими физическими ограничениями определяются их свойства, каких целей пытаются достичь их разработчики и эксплуататоры, как в них обеспечивается надёжность, криптографические и стеганографические свойства, как работают mesh-сети и прочая, и прочая.