Xen Project Contributor Training
Part 3 : Communiucation

Lars Kurth
Community Manager, Xen Project
Chairman, Xen Project Advisory Board
Director, Open Source Business Office, Citrix

lars_kurth

Outline for Part 3 – Communication

Goal: using Effective Communication to come up with the best compromise

The Big Picture

Mindset of Collaborative Engagement

Terse Communication Style

Dealing with rudeness / inappropriate comments

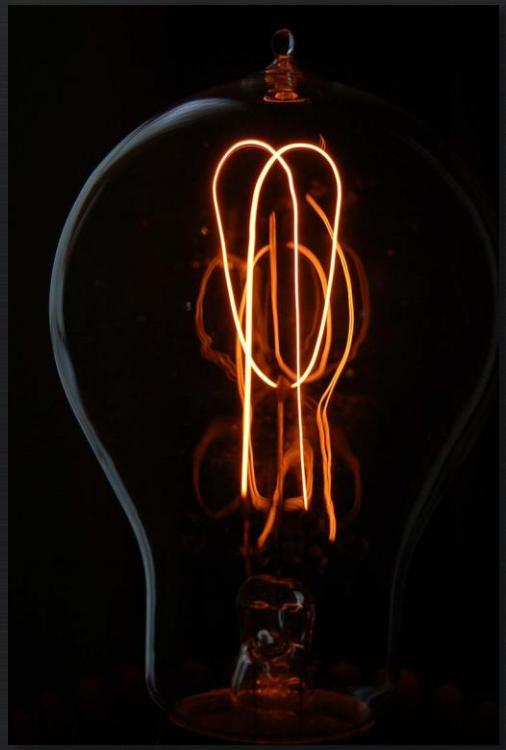
Collaborative Communication Patterns

- High Quality Advocacy
- High Quality Inquiry
- The Left Hand Column
- The Ladder of Inference

Anti Patterns:

- How to spot signs of unproductive communication

Applying Communication Patterns in Code Reviews



Goal:

Enable you to Efficiently communicate with Xen Project Developer Community

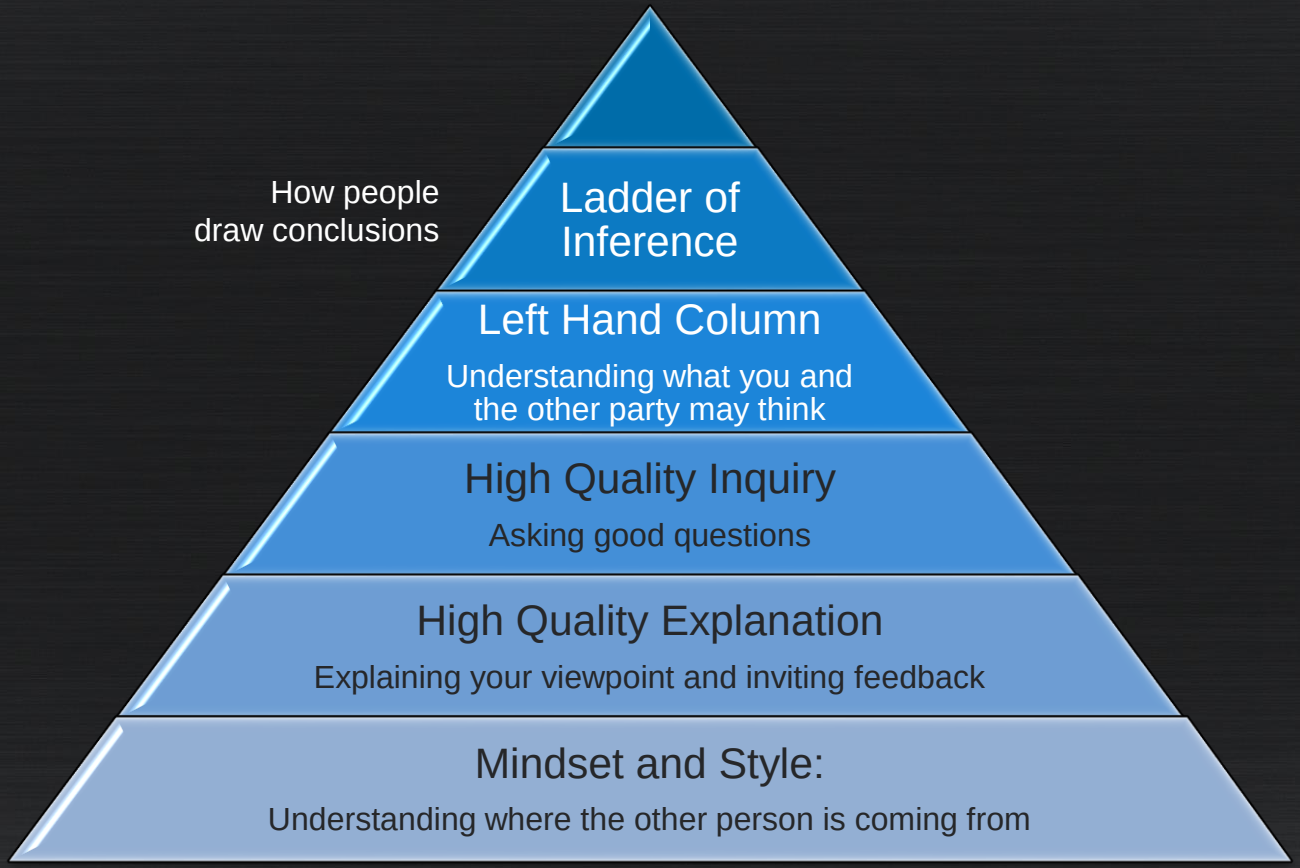
This is a 2 way street

Vinovyn @ Flickr

Communication Patterns

Building blocks of good communication

A positive outcome



Mindset:

Guide what we think
Guide how we feel about people and things

THUS, they influence communication

Thomas Galvez @ Flickr

Adversarial vs. Collaborative Mindset

Adversarial: Two ideas enter, one idea leaves

Collaborative: Participants build off of each others' ideas, working together to create something new

IMPORTANT:

- **It takes two parties in a discussion**
- **BUT: the initiator (in our case the contributor), sets the style by opening the conversation**
- **There are techniques to bring back a conversation from Adversarial to Collaborative**

YOU – ADVERSARIAL:

Assuming there is one best way to understand complex problems

Assuming your point of view is complete and addresses all aspects of the situation

Regarding your viewpoint as fact that should be obvious to others

Inventing ways to bypass others' options wile getting them to buy yours

Minimizing concerns and finding ways to bypass them

Discounting criticism and considering it a threat

Searching for data and views that only serve to confirm your opinion

YOU – COLLABORATIVE:

Assuming there are different ways to understand complex problems

Assuming your point of view is incomplete and misses some aspects of the situation

Regarding your viewpoint as hypothesis to be explored with others

Inventing ways to test or explore options together

Actively seeking others' concerns and revising your plan accordingly

Using criticism to continually improve

Searching for data and views that might change your opinion

CONTRIBUTOR:

Get your code looked at as quickly as possible

May have 1-2 patch series under review at a given time

Understands motivation and background behind the code (as you wrote it)

Get your code into the code line, as quickly as possible and with as few changes as possible

Cares about the process of contributing being predictable

May not fully understand (or in some cases not care as much about) the long term impact of contributed code

REVIEWER / MAINTAINER:

May have a long TODO list of code to look at

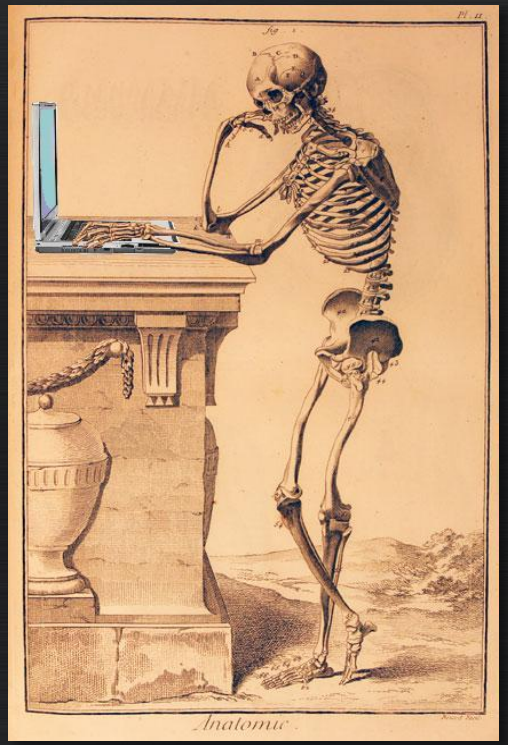
May have to review many different patch series from many different people

Has to reverse engineer motivation, design, etc. from code and conversation – needs to understand in order to be able to accept the code

Wants to ensure that the code does not affect quality, performance, security, etc. in a negative way

Cares about all open questions and concerns raised being resolved

Domain expert, who cares deeply about the long term impact of the code



Before you …

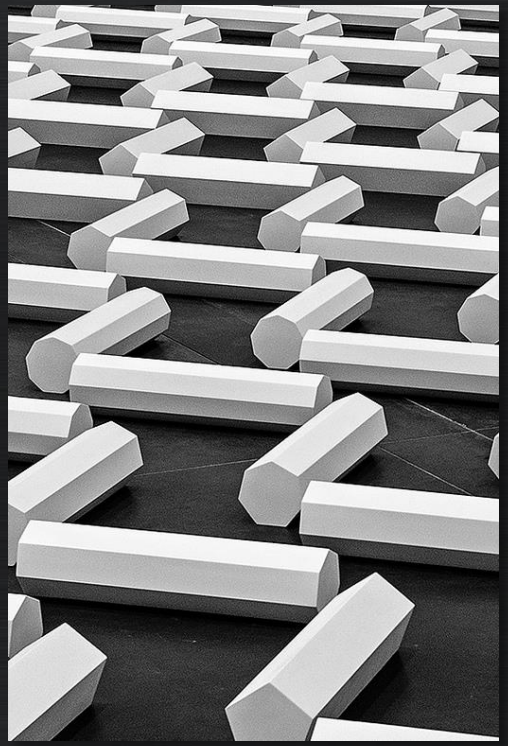
Write the code
(to help you decide whether you need a design discussion, etc.)

AND again before you submit the code
(to help you decide how much information you need to provide)

Think about the

- reviewer's viewpoint
- the contribution process
- anatomy of a good patch series

Mike Licht @ Flickr



Terse Communication:

Reviewers often adopt a terse writing style

They often write 1000’s of messages

Seth Anderson @ Flickr, cropped

Example of a Terse Mail Exchange

[Contributor]
Subject: [Patch v1] Add support for HYPERVISOR_sysctl
Signed-off-by: <Name of contributor>
<Including the code with no other context>

[Reviewer]
> Signed-off-by: <Name of contributor>
Why?

[Contributor]
I'll add authors Signed-off-by before my Signed-off-by in the next patch-set.

[Reviewer]
Sorry, I meant why are you introducing HYPERVISOR_sysctl?

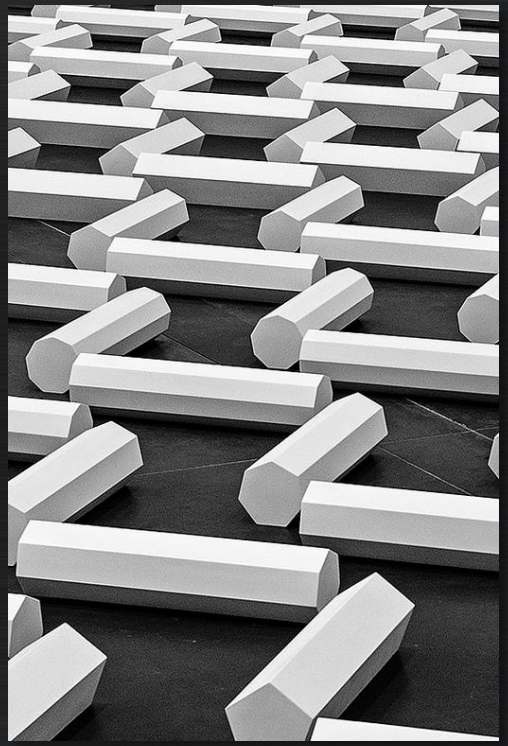
[Contributor]
I use it to get real physical CPUs counter. Also I'll implement a new sysctl operation: XEN_SYSCTL_cpufreq_op. Kernel will use this op to start/stop cpufreq notification events sending.

[Reviewer]
Please add the explanation to the commit message. Also don't add structs and definitions that you don't need, such as xen_sysctl_readconsole and XEN_SYSCTL_SCHEDOP_putinfo.

Group Exercise:

Think about how this email exchange made you feel and share with the group

Thomas Galvez @ Flickr



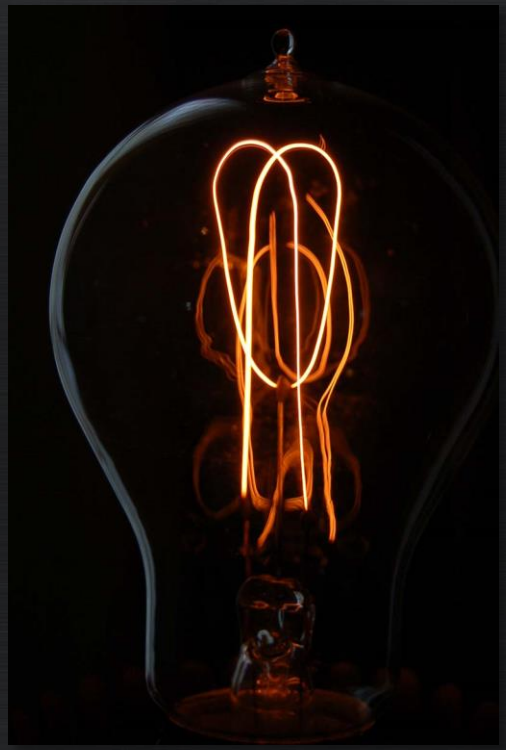
Unintended Consequences:

Terseness can be interpreted as coldness or lack of emotion and can be misinterpreted as rudeness in different cultures

Terseness can pull a conversation into an adversarial direction as it can trigger an emotional response

Terseness can lead to misunderstandings

Seth Anderson @ Flickr, cropped



Observation:

In this example (and similar ones), the contributor started the review without any context in a terse style

Reviewers normally adapt their style

- For newcomers
- For people where they know they don't like a terse communication style

Vinovyn @ Flickr

Contributors: Set the Right Tone

Two possibilities:

- Terseness does not bother you ➔ you don't have to do anything
- Terseness does bother you
  - Start the review thread with in an appropriate format

    ```
    Hi all | Hi <names of maintainers>, please find attached
    the patch series for XYZ, … Regards
    ```

  - If you never contributed to the project before, you can add a quick introduction and some background about yourself and your motivation
  - Normally, this will result in a response in a similar style. If not, you can use the Left Hand Column (see later) and privately approach the reviewer.

Reviewers: Add Context

A terse communication style usually leaves out contextual information that is known in the community. However, for newcomers it is **important** to put it back in

Greetings:	"Hi new user X"
Thanking:	"Thank you for submitting this EFG"
Introductions:	"I maintain XYZ"
Background:	"This is used for ABC"
Sign-off:	"Good luck and I hope the info I provided helps"

Bad Behaviour:

How to spot
What to do if it occurs

Thomas Ricker @ Flickr, cropped

Acceptable:

- Objective feedback on specific issues in code / designs / proposals / etc.
- Objective feedback based on risk at a certain point in the release cycle

Not Acceptable:

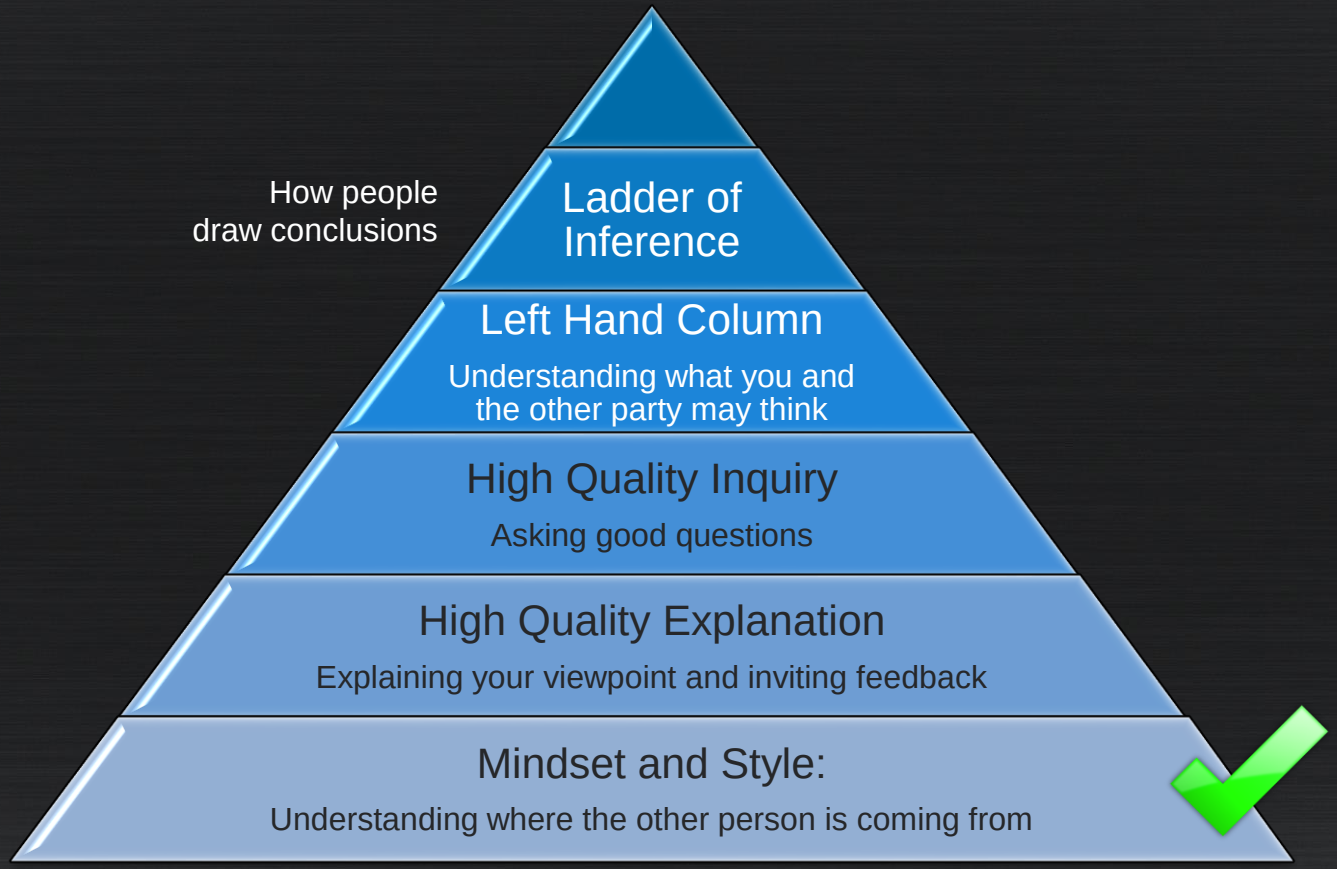
- Personal attack
- Referring to "*track record*" of past patches
- Arguments along the lines of "*I know this better, go away*"
- Rejections of patches based on gender, ethnicity, cultural background, believes, etc. or other non-objective feedback
- Abuse or bullying of any kind

What to do, if Bad Behavior occurs?

Send an e-mail to community dot manager at xenproject dot org outlining your concern

A concern should be raised within a few days of the issue having surfaced, with a reference to the bad behavior on a Xen Project mailing list

A positive outcome



3 elements of High Quality Explanation

Clearly state your opinion, proposal, case, background for a patch series, …

- I (want to) propose an idea/patch/design …
- Here is an idea/patch/design on how to solve …
- I suggest that …
- My point is …
- I think …
- Consider this …
- Let's assume we did …

Offer reasons, assumptions, examples, analysis, … (there can be several)

- I am assuming that …
- Here is an example/my analysis/…
- Here is my thinking (that led to the solution) …
- Given that …
- Let me explain why …
- The reasons for … are …

Seek Challenge

- Did I miss anything?
- Do you have any concerns related to the proposal?
- Are there any assumptions that you don't share?
- Is there anything you would change?
- Are there any concerns that you have?
- Where is my thinking/proposal unclear?
- [If there was some differing views beforehand] Do you think there is a way forward that satisfies both our views?

Sentence starters
(crutches that help you not forget anything)

Example of a Patch Description with HQE

Hi all,

[states the proposal/problem this patch solves]
please find attached a patch that solves the following problem: when using pvgrub in graphical mode with vnc, the grub timeout doesn't work. The countdown doesn't even start. With a serial terminal the problem doesn't occur and the countdown works as expected.

[offers an analysis showing where the problem is suspected]
I debugged the issue and identified that when using a graphical terminal, checkkey() returns 0 instead of -1, when there is no activity on the mouse or keyboard. As a consequence grub thinks that the user typed something and interrupts the count down.

[offers more background information related to the solution]
Here is my thinking on how to fix the issue: we can simply ignore keystrokes returning 0 and avoid the problem. It is safe to ignore 0 as this the NUL character anyway.

[seeks challenge]
Did I miss anything?

Regards
XYZ

Note that "*sentence starters*" are not strictly necessary, but they do help non-experienced contributors and non-native English speakers

Example of a Patch Series Description

What

Hello,

this patchset adds an implementation of the XEN_DOMCTL_memory_mapping hypercall for the arm32 architecture.

As a part of my thesis project (developed with Paolo Valente, in Italy, [1]), I am porting an automotive-grade real-time OS (Evidence's ERIKA Enterprise, [2], [3]) on Xen on ARM, and the port of the OS as a domU has required the OS to be able to access peripherals performing memory-mapped I/O. As a consequence, the I/O-memory ranges related to such peripherals had to be made accessible to the domU.

Why, For Whom, Use-Case, Requirements, Implications

Note that there are NO "*sentence starters*"

I have been working on this patchset for the last weeks after reading a related xen-devel discussion ([4]), and I had no idea that Eric Trudeau had already proposed to this mailing list an implementation of the hypercall ([5], [6]). The code has been tested with Linux v3.13 as dom0 and ERIKA Enterprise as domU.

Any feedback about the patches is more than welcome.

Additional Context Information

[1] http://www.unimore.it/
[2] http://www.evidence.eu.com/
…

Requests feedback

How is this different from stating an Opinion?

Clearly state your opinion, proposal, case, background for a patch series, …

What, For Whom, …

- This is what normally happens
- **IMPORTANT:** Usually part 2 & 3 are omitted

Offer reasons, assumptions, examples, analysis, …

Context: Why, How, Assumptions, Examples, Restrictions, Open Issues, …

- This provides enough context for the reviewer to understand your thought process
- It cuts down the number of iterations as you are likely to get fewer questions
- You can direct the reviewer to specific thought processes where you want to get feedback

Seek Challenge

Prompts a collaborative style

- Signals that you are prepared to listen
- It sets the tone for a collaborative discussion
- You can direct focus to a specific area which worries you (e.g. have I missed an assumption)

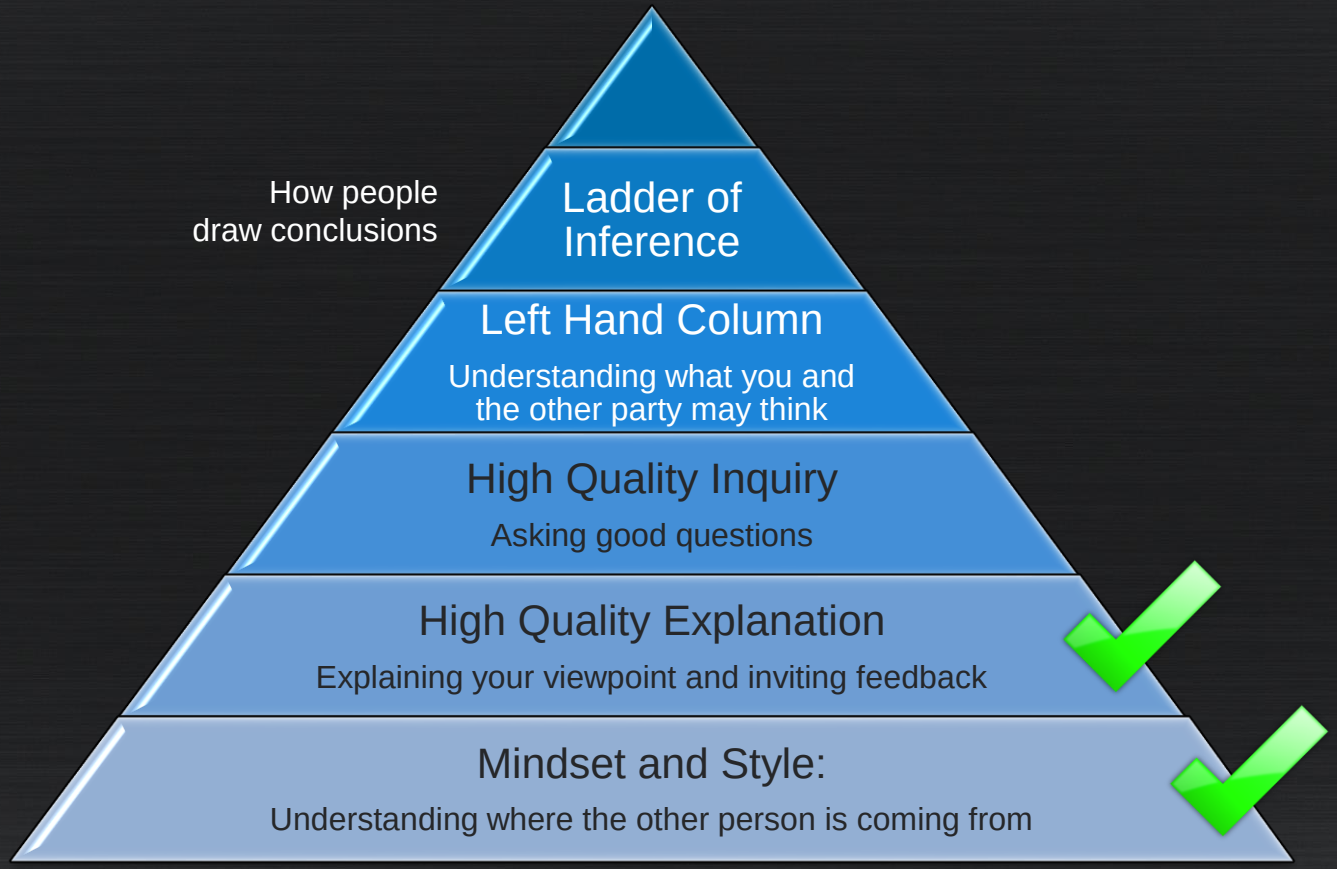
When is this technique useful?

- Explaining a patch/design/RFC/etc.
  - A well structured patch series maps one “HQ Explanation (HQE)” against a one patch
    (because a good patch does one logical thing and can thus be described using this technique)
  - For a patch series, you can us HQE to describe the use-case aka “what & for whom”
    (“what & for whom” often makes a better logical mapping for a series than a simple “what”)
- During a design/patch review when …
  - there is a misunderstanding or potential for a misunderstanding
  - there are several ways to solve the or an element of the problem
  - you are asked why you have chosen a specific approach
  - etc.
- It can be useful in code comments
  (with the exception of seeking challenge)

When is this technique NOT useful?

- Acknowledging agreement, confirmation plans to fix something in a review, etc.
- When a simple answer suffices
  - Remember, that part of the reason to use HQ Explanation is to avoid unnecessary iterations
  - It also signals and invites more comments : thus when the discussion is over there is no need for HQE

A positive outcome



3 elements of High Quality Inquiry (HQI)

Ask a question

- What leads you to …?
- What's an example of …?
- What else have you considered/is wrong/…?
- What data/assumptions led you to...?
- Why do you think…?
- What would it take to do …?

Disclose your reason for asking (there can be several)

- I am asking because, …
- I am unclear why …, so …
- I have made a different assumption, so…
- I made a different observation, so …

Seek Understanding

- What I hear you say is … is this right?
- I hear two different points … is this correct?
- Am I correct, you are assuming …?

Sentence starters
(crutches that help you not forget anything)

The Purpose of High Quality Inquiry

- Encourages the expression of diverse opinions, doubts and concerns
- Helps you find out what you may be missing by encouraging others to identify possible gaps and errors in your thinking
- Generates information for more informed choices
- Facilitates insight and adoption of new perspectives
- Leads to learning, which is important during a code/design review

Example : Using HQI to Clarify

[Proposer]
Can you give me a concrete example of what you mean by Dom0 having to be able to change the frequency of a CPU it does not run on?

I am not sure I understood your point correctly. In the case of OMAP changing the CPU frequency affects both cores, but I heard that Snapdragon processors might be able to change CPU frequs independently.

Can you confirm whether this would be a problem?

[Reviewer]
OK.

Let's suppose that the platform has 2 physical cpus, each cpu has 4 cores. Let's also suppose that dom0 has only 2 vcpus, currently running on core0 and core1 of cpu0.

In this case dom0 must be able to change the frequency of core3 of cpu1, despite not actually running on it. If this can be done without any hacks, then we can go ahead with your proposed approach.

[Proposer]
Thank you for explanation. I think this requires more and deeper investigation, but for sure Dom0 must be able to do this.

When is this technique useful?

- During design / architecture / use-case related discussions to further common understanding of the problem to be solved
- To get some more information about a complex situation. By disclosing the reason for asking, you provide more detail and you also point others in a direction which causes a problem for you.
- Very useful to check assumptions or how someone came to a conclusion (also see Ladder of Inference)
- When two viewpoints are clashing (Advocacy War) or a discussion goes nowhere (Recycling) and thus there is no resolution
- This is a technique that reviewers should use where appropriate

Another Example

[Reviewer asking a question, in response to a question about two design options]

Have you asked yourself whether this information even needs to be exposed all the way up to libxl?

I am asking, because I believe that the use-case will heavily determine the design of the hypercall interface. This will also impact your question as to what the best design is.

I would probably approach the design quite differently depending on whether a) the consumer was a low-level CLI tool (e.g. something such as xenpm), b) whether the consumer is some tool or client that consumes the information using XL, or c) whether the consumer was within a cloud orchestration stack such as openstack or cloudstack.

From your initial design proposal, I think the answer is b) or c). Would you agree and can you clarify?

How is this different from asking a Question?

Asking

- This is what normally happens
- **IMPORTANT:** Usually part 2 & 3 are omitted

Disclose your reason for asking (there can be several)

- Provides additional context for the recipients
- Allows the asker to highlight specific problems or concerns
- Guides the conversation in a certain direction (gives the asker some control in influencing the outcome)

Context, Highlights areas of concern, guides the discussion, …

Seek Understanding

- Links back to the previous speaker's point/argument/statement/…
- Signals willingness to collaborate
- Shows that you listened

Prompts a collaborative style

Interlude:

Blinking Words Revisited

Reminder: "Blinking Words”

Blinking Words are words or phrases that take on many possible interpretations, and where definitions blink between different meanings depending upon who hears it.

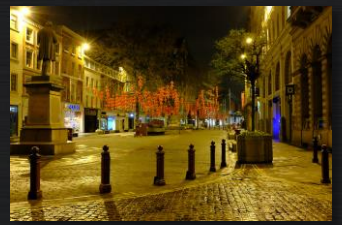
People with similar background can interpret terminology that is commonly used in their field very differently.

Use High Quality Inquiry to avoid issues with Blinking Words!

A bit of Fun: Words that have a different meaning in London and Manchester

Simon & His Camera @ Flickr

Trevor Cummings @ Flickr



Sound

London: Vibrations that travel through the air, noise.
Manchester: Good/decent

Buzzing

London: A low, continuous humming sound
Manchester: Happy/excited

The point: it is very easy to misunderstand each other when from different cultures

A bit of Fun: The Power of Blinking Words

A: This project is a nightmare, I can't wait for it to end.

Identify blinking word, and ask about it

Q: What makes the project a nightmare?

A: It's a nightmare because of my project manager.

Listen, identify another blinking word, and ask about it.
In this example, we have more of a "blinking phrase".

Q: What did your project manager do, to make the situation hard for you?

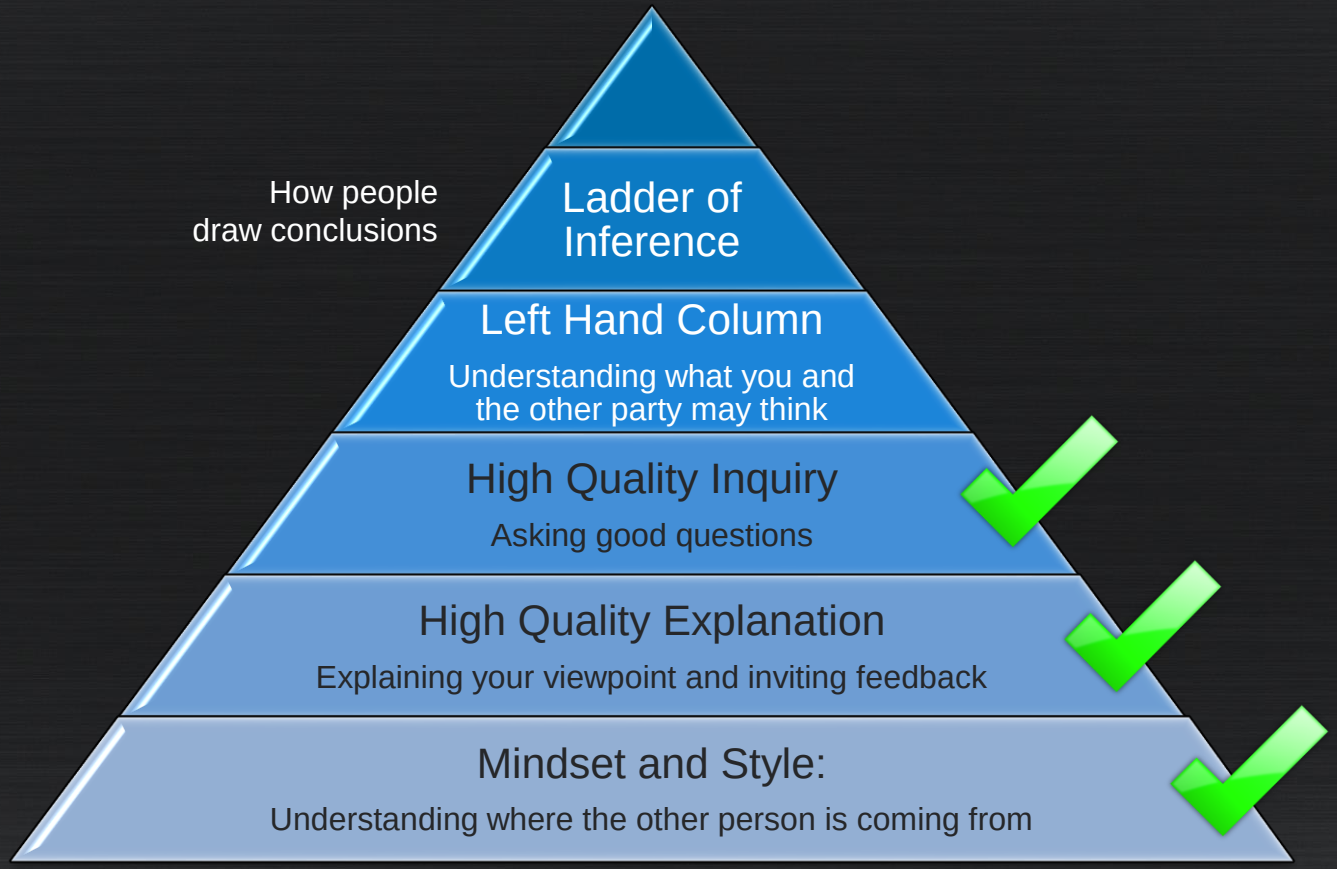
A: He complains about everything. Then he asks for more information.

Listen, identify another blinking word, and ask about it.

Q: Tell me about his biggest complaint?

…

A positive outcome



Left Hand Column

Assumptions: during conversations (verbal or e-mail) there are actually two conversations taking place

- Explicit (Right hand column): the words that are spoken or written in an exchange between two or more persons
- Left Hand Column: What the individuals are thinking and feeling but not saying

Purpose:

- Become more aware of the assumptions, thoughts and feelings that govern our conversations and contribute to blocking us
- We can also use HQ Inquiry to try and uncover what's in the other person's left Hand Column
- This allows us to resolve difficult situations or conversations which have become stuck

Example: How does it work?

LHC: What Di is thinking & feeling but not saying

RHC: What Di and Bill are saying

Bill: Di, we're a bit behind with our organisation for the camp perhaps we should meet after school.

Di: I've been very concerned about the camp but I've been very busy with reports and family stuff. But, of course, we can squeeze in a meeting.

Bill: Well it's occurred to me that we could use better coordination between us. There are probably some ways I could help.

Di: Well, I'm happy to talk through any ideas you have in mind.

Bill: I don't have anything specific in mind.

Di: Oh well maybe we can meet Thursday and make a list of what needs to be done.

Group Exercise:

Try and guess what Di was thinking and fill out the Left Hand Column

Afterwards: we will discuss whether you discovered some insights

Thomas Galvez @ Flickr

Example: How does it work?

LHC: What Di is thinking & feeling but not saying

RHC: What Di and Bill are saying

Bill: Di, we're a bit behind with our organisation for the camp perhaps we should meet after school.

Di: I've been very concerned about the camp but I've been very busy with reports and family stuff. But, of course, we can squeeze in a meeting.

Bill: Well it's occurred to me that we could use better coordination between us. There are probably some ways I could help.

Di: Well, I'm happy to talk through any ideas you have in mind.

Bill: I don't have anything specific in mind.

Di: Oh well maybe we can meet Thursday and make a list of what needs to be done.

Example: How does it work?

LHC: What Di is thinking & feeling but not saying

It's three weeks until camp and I didn't think he was bothered. I was hoping we would catch up.

I need to make it clear that I'm willing to take responsibility for some things, but I don't want to be the one left holding the bag for this.

He never offers his help until most of the work has been done. Why can't he get organized earlier? But it's too late now to bring this up.

Not helping sooner is the real reason we're behind. He knows I will do the work and never offers help.

No he wants me to do it so he doesn't have to.

I'll have it all done by Thursday and just tell him what to do. It is so unfair.

RHC: What Di and Bill are saying

Bill: Di, we're a bit behind with our organisation for the camp perhaps we should meet after school.

Di: I've been very concerned about the camp but I've been very busy with reports and family stuff. But, of course, we can squeeze in a meeting.

Bill: Well it's occurred to me that we could use better coordination between us. There are probably some ways I could help.

Di: Well, I'm happy to talk through any ideas you have in mind.

Bill: I don't have anything specific in mind.

Di: Oh well maybe we can meet Thursday and make a list of what needs to be done.

Example: How does it work?

LHC: What Di is thinking & feeling but not saying

It's three weeks until camp and I didn't think he was bothered. I was hoping we would catch up.

I need to make it clear that I'm willing to take responsibility for some things, but I don't want to be the one left holding the bag for this.

He never offer
work has been
earlier? But i
up.

Not helping so
behind. He kno
offers help.

No he wants me

I'll have it all done by Thursday and just tell him what to do. It is so unfair.

RHC: What Di and Bill are saying

Bill: Di, we're a bit behind with our organisation for the camp perhaps we should meet after school.

Di: I've been very concerned about the camp but I've been very busy with reports and family stuff. But, of course, we can squeeze

hat we could
us. There are

ugh any ideas

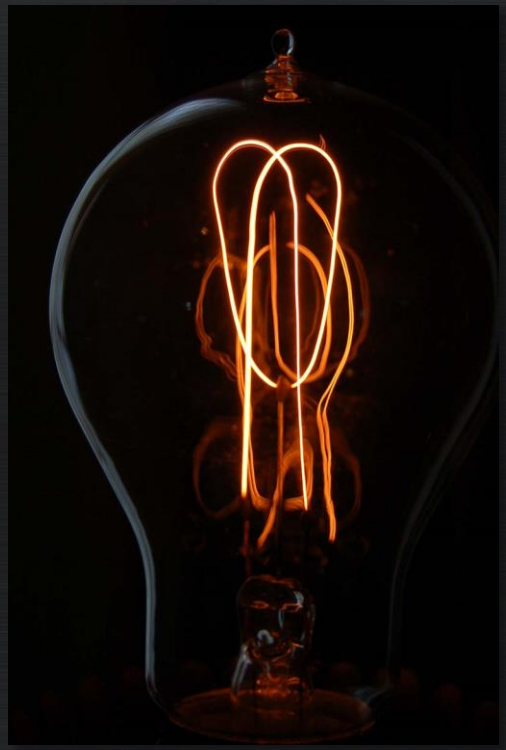
ific in mind.

Di: Oh well maybe we can meet Thursday and make a list of what needs to be done.

In this example Bill would have left the conversation feeling that there were no concerns.

Di however would still feel frustrated that her issues about planning, and shared responsibilities hadn't been raised.

By mapping the LHC and reflecting Di may choose to raise some of her concerns on Thursday and clear the air.



Observation:

What was unsaid, is a key reason for misunderstandings

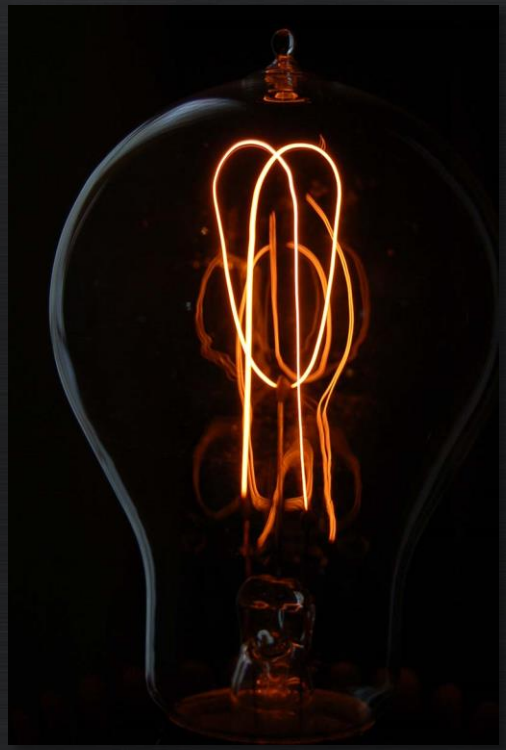
It is also a key reason that generates frustrations and impacts your mindset towards your communication partner and thus your ability to collaborate

HQ Explanation and Inquiry are designed to proactively disclose upfront what otherwise would be unsaid and can help

Vinovyn @ Flickr

A word of Caution

Watch a trailer of “What Women Want” on https://www.youtube.com/watch?v=VFwHs7fEUNs



Lesson:

If we are going to be explicit about our LHC, we need to choose our very words carefully!

Especially in a conversation in public

Vinovyn @ Flickr

3 elements of expressing Left Hand Columns

Name What you Feel

- My concern is / I am concerned that …
- I get frustrated when…
- I suggest that …
- I am impatient with …
- I believe / feel …

Make an Observation to support your case (there can be several)

- What I observe / notice is …
- How I see it …
- What I hear …
- It appears that …
- As far as I can see …

Redirect (suggest a resolution and invite feedback)

- What would happen if we …?
- Given X I suggest …?
- Shall we move to…
- We could try out …

Sentence starters
(crutches that help you not forget anything)

Example: Stuck Patch due to disagreement

Contributor has proposed a patch series and it turns out that there is disagreement between two reviewers. You are deadlocked because of it.

You can apply LHC, to try and unblock the deadlock

> [Contributor]
> Hi X, Y.
>
> *I believe* that this patch series is not progressing, because of differing views that have been expressed.
>
> *How I see the situation is that*, X supports solving <problem> in <this way>. And Y beliefs that <problem> is best solved in <that way>.
>
> *We could try and* set up an IRC meeting to explore the trade-offs in more detail and find a solution. What do you think?

You would then use a combination of HQ Inquiry and Exploration techniques to get to the bottom of the problem.

Example: Variant of the previous

I believe that this patch series is not progressing, because of differing views that have been expressed.

How I see the situation is that, X supports solving <problem> in <this way>. And Y beliefs that <problem> is best solved in <that way>.

As far as I can see, the cause of this disagreement is that we have not fully explored <use case ABC>. I believe this because of <XYZ>. Would you agree?

Here we have a HQ Explanation nested within the Left Hand Column

Should we try to explore <use case ABC> some more on the list? What do you think?

Example: Concern about use-case

Contributor has proposed a patch series and the maintainer is concerned that the solution is not general enough. The maintainer can apply LHC, to raise the concern

[Maintainer]

Based on our discussion about <interface X>, *I am concerned* that the interface is not generic enough.

It appears that only <use-case A> has been considered.

As you know, we cannot change interfaces due to backwards compatibility guarantees. The reason is that we have many large users, who would be impacted by interface changes. Do you see this differently?

Given, that we cannot change interface later on and we seem to not have considered all use-cases, could I ask you to think about more use cases to test the interface design and get back to me?

When is this technique useful?

During technical discussions we frequently deal with trade-offs and concerns.

- The LHC pattern is extremely useful, to highlight concerns in a non-confrontational way
- The Make an Observation segment, ensures that the conversation stays grounded and does not become too personal (whenever emotions are involved, there is potential to damage a relationship)
- The Redirect segment, provides an actionable way forward on how to resolve the concern (aka move past it) and continue to work together