Understanding Assurance Cases:
An Educational Series
in Five Parts

Version 2.0 (2020-07-15)

C. Michael Holloway
NASA Langley Research Center
c.michael.holloway@nasa.gov

This material was originally created in 2015-16, as part of the Explicate '78 project. The project was supported in substantial part by the Assurance Case Applicability to Digital Systems task under the reimbursable interagency agreement with the Federal Aviation Administration for Design, Verification, and Validation of Advanced Digital Airborne Systems Technology (IAI-1073 Annex 2 for NASA; DTFACT-10-X0008, Modification 0004 for the FAA). The original presentations were delivered to a selected group of FAA civil servants and NASA Langley personnel. The audio was recorded and partial transcripts (containing only the words spoken by the presenter, Mr. Holloway) produced. The intent from the beginning was to collect the material into a form that could be made available publicly.

The text is based closely on the original transcripts, except where changes have been made to the original presentations to keep the material current with the state of the art and practice. These updates are supported in part by NASA IA-303333/FAA IA NO 692M15-19-T-00029 Annex 1/TO 1.

Additional updates to Modules 3 and 4 are expected to be completed by the first half of 2021.

The full collection, both individually and as a single file, are available for download at

https://shemesh.larc.nasa.gov/arg/uac.html

If you are interested in this material, you may well be also interested in the following two recent (2020) documents:

A Primer on Argument (Overarching Properties Edition)

A Friendly Argument Notation (FAN)

Understanding Assurance Cases:
An Educational Series in Five Parts

Here are the learning objectives for each of the 5 modules making up the *Understanding Assurance Cases* educational material.

Module 1: Foundation concentrates on the key concepts, phrases, notations, and characteristics that define assurance cases. A person completing **Foundation** should be able to

- Provide a good definition of 'assurance case'
- Explain the key concepts of assurance cases and recognize various terms for those concepts
- Identify some existing notations for expressing assurance cases
- Enumerate characteristics that an assurance case should have

Module 2: Application concentrates on the history, current uses, and potential benefits and problems of assurance cases. A person completing **Application** should be able to

- Cite selected past events relevant to the development of the assurance case approach
- List uses of assurance cases in several domains
- Discuss possible lessons learned from past uses
- Explain potential benefits and problems associated with assurance cases

Module 3: Evaluation concentrates on the philosophy and methods for evaluating whether a particular assurance case is sufficient for its purpose. A person completing **Evaluation** should be able to

- Identify positive properties that an assurance case should have
- Identify negative properties that an assurance case should not have
- Enumerate steps for evaluating an assurance case
- Suggest potential corrections for selected deficiencies

Module 4: Creation concentrates on methods for creating assurance cases. A person completing **Creation** should be able to

- Enumerate steps for creating a new assurance case
- Explain essential questions that must be answered while developing a case
- Identify common mistakes made in assurance case creation
- Create a simple assurance case

Module 5: Speculation concentrates on the possible ways assurance cases may fit into current environments, current research in the field, and how to learn more about assurance cases. A person completing **Speculation** should be able to

- Compare and contrast an assurance case approach with other approaches
- Discuss how an assurance case approach could fit into a regulatory environment
- List current areas of assurance case research
- Locate references for further study

The modules are directly accessible through the following links:

- `https://shemesh.larc.nasa.gov/arg/uac-module1-foundation.pdf`
- `https://shemesh.larc.nasa.gov/arg/uac-module2-application.pdf`
- `https://shemesh.larc.nasa.gov/arg/uac-module3-evaluation.pdf`
- `https://shemesh.larc.nasa.gov/arg/uac-module4-creation.pdf`
- `https://shemesh.larc.nasa.gov/arg/uac-module5-speculation.pdf`

This material was originally created in 2015-16, as part of the Explicate '78 project. The project was supported in substantial part by the Assurance Case Applicability to Digital Systems task under the reimbursable interagency agreement with the Federal Aviation Administration for Design, Verification, and Validation of Advanced Digital Airborne Systems Technology (IAI-1073 Annex 2 for NASA; DTFACT-10-X0008, Modification 0004 for the FAA). The original presentations were delivered to a selected group of FAA civil servants and NASA Langley personnel. The audio was recorded and partial transcripts (containing only the words spoken by the presenter, Mr. Holloway) produced. The intent from the beginning was to collect the material into a form that could be made available publicly. The text adheres closely to the original transcript, except where changes have been made to the original presentation since it was first given, as part of work for for NASA IA-303333/FAA IA NO 692M15-19-T-00029 Annex 1/TO 1.

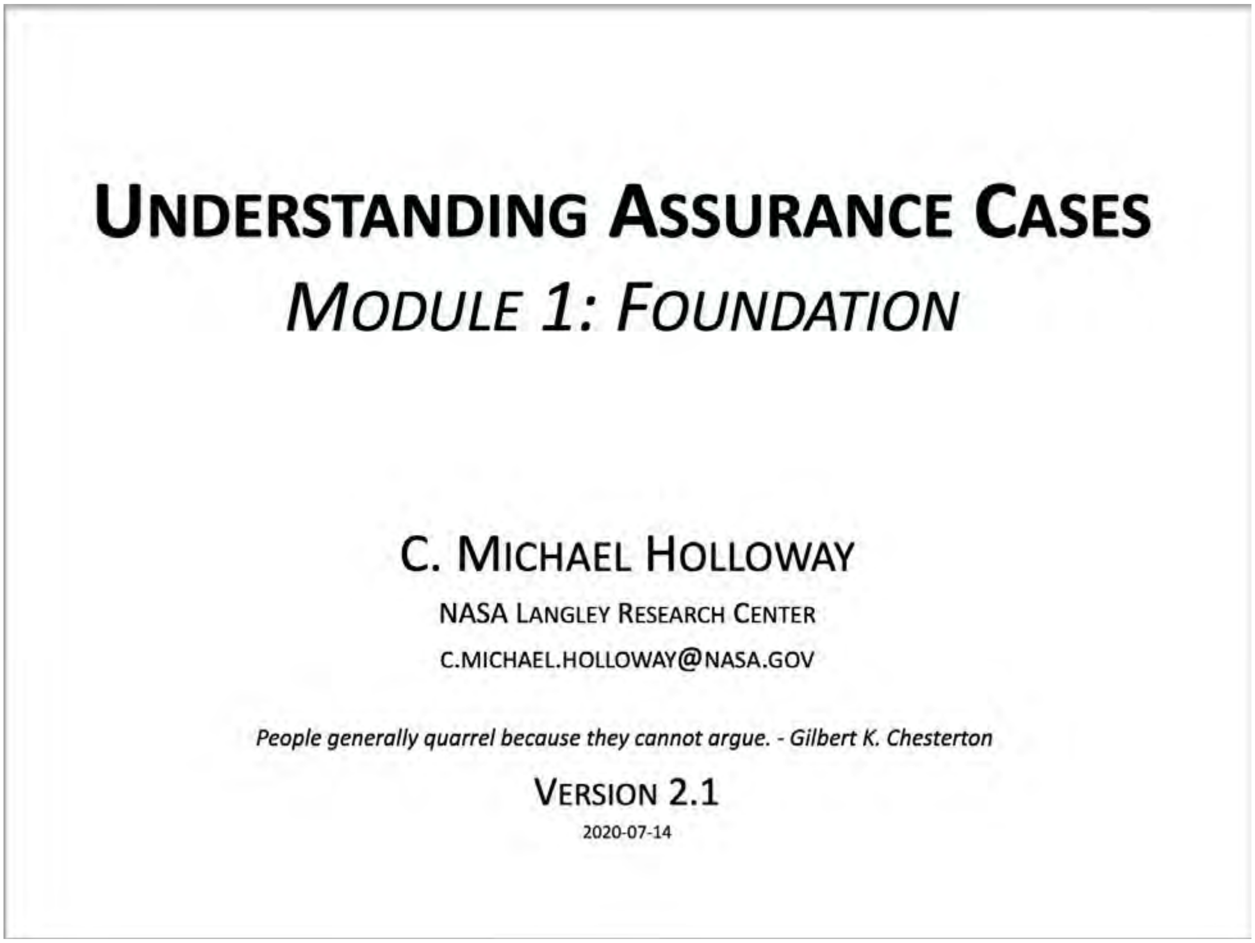
The full collection consists of six documents (including this one), which are available electronically through `https://shemesh.larc.nasa.gov/arg/uac.html`.

Understanding Assurance Cases:
An Educational Presentation in Five Parts

Module 1: Foundation

C. Michael Holloway
c.michael.holloway@nasa.gov

Senior Research Computer Engineer
Safety-Critical Avionics Systems Branch
NASA Langley Research Center, Hampton, Virginia, U.S.A.



UNDERSTANDING ASSURANCE CASES

MODULE 1: FOUNDATION

C. MICHAEL HOLLOWAY

NASA LANGLEY RESEARCH CENTER

C.MICHAEL.HOLLOWAY@NASA.GOV

People generally quarrel because they cannot argue. - Gilbert K. Chesterton

VERSION 2.1

2020-07-14

This material was originally created in 2015-16, as part of the Explicate '78 project. The project was supported in substantial part by the Assurance Case Applicability to Digital Systems task under the reimbursable interagency agreement with the Federal Aviation Administration for Design, Verification, and Validation of Advanced Digital Airborne Systems Technology (IAI-1073 Annex 2 for NASA; DTFACT-10-X0008, Modification 0004 for the FAA). The original presentations were delivered to a selected group of FAA civil servants and NASA Langley personnel. The audio was recorded and partial transcripts (containing only the words spoken by the presenter, Mr. Holloway) produced. The intent from the beginning was to collect the material into a form that could be made available publicly. The text adheres closely to the original transcript, except where changes have been made to the original presentation since it was first given, as part of work for for NASA IA-303333/FAA IA NO 692M15-19-T-00029 Annex 1/TO 1. The full collection consists of six documents, which are available electronically through https://shemesh.larc.nasa.gov/arg/uac.html.

Hello everybody.

Welcome to the first module in an educational series about Understanding Assurance Cases. In this module, we will examine the ***Foundation*** of the assurance case concept.

Because talking about the foundations will involve talking quite a bit about *argument*, the quotation you see here from Gilbert K. Chesterton is particularly appropriate:

"People generally quarrel because they cannot argue."

[Chesterton, G. K. 2002. The C*ollected Works of G.K. Chesterton*. Electronic edition: (v35) Illustrated London News, 1929-1931. Charlottesville, Va: InteLex Corporation.]

When we talk about argument we will *not* be talking about emotion-filled disagreements; instead, we'll be talking about rational, careful discussion of reasons for thinking one thing rather than another.

My hope is that this hour will be interactive. There will be several times when I'll ask you a question, and many times when I'll stop to give you a chance to ask me questions. Nevertheless, feel free to interrupt me at *any* point if you have a burning question that you can't hold until later. I'll do my best to extinguish it.

[Question to participants: Does anyone have any questions or comments that you want to make now at the beginning?]

Before going any further, I feel duty-bound to alert you to an intentional act of deception underlying this, and the other, modules.

Within the assurance case community, intramural debates abound about a variety of topics we will discuss. Except in rare instances the existence of these debates[1] is intentionally ignored or mentioned only briefly in this material. Here's why.

Disagreements exist about terms, definitions, notations, philosophy, procedures, tools, and just about everything else.

The depth of the disagreements ranges all the way from *shallow* differences in preferences (which term best denotes a particular concept, for example), to rather *deep* philosophical differences (the feasibility and desirability of formalizing assurance arguments, for example).

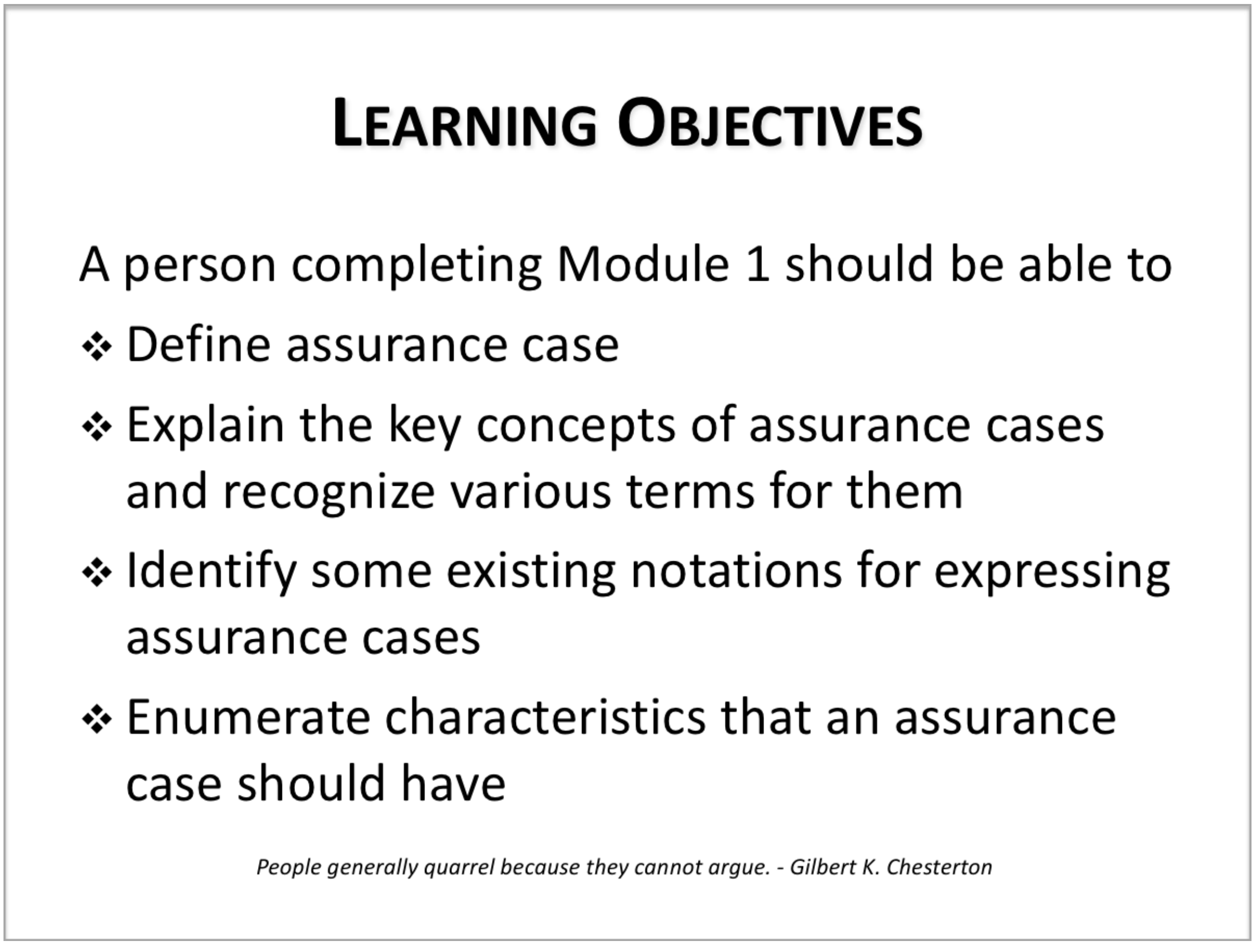
Spending *too much* time on these disagreements would likely make this material deeply confusing; but spending *too little* time on them might hinder your understanding of some materials you may come across.

[1] By using the word 'debates', I'm intentionally obscuring something else, too, namely the fact that *some* of the disagreements have all of the attributes of quarrels (not *all* by any means, but *some*).

In trying to strike a balance, what I've chosen to do is *not* highlight the areas of disagreement on the slides (except occasionally where it is seems essential), but to mention the disagreements where appropriate in my words accompanying the slides.

[Question to participants: Any questions about this issue?]

One other quick note before we proceed: All images you see were either created by me (Michael Holloway) or are in the public domain via CC0 1.0 Universal.



Learning Objectives

A person completing Module 1 should be able to

- ❖ Define assurance case
- ❖ Explain the key concepts of assurance cases and recognize various terms for them
- ❖ Identify some existing notations for expressing assurance cases
- ❖ Enumerate characteristics that an assurance case should have

People generally quarrel because they cannot argue. - Gilbert K. Chesterton

Let's discuss learning objectives.

By the time we're finished today, I hope that you'll be able to do at least four things.

First, provide a definition for the term 'assurance case'. Although I've not listed it on the slide, I also expect that you'll be able to provide definitions for more specific variants such as 'safety case' or 'security case'.

Second, explain the key concepts of assurance cases and recognize various terms used for those key concepts. This is one area where the slides and my oral commentary will both note some differences within the community.

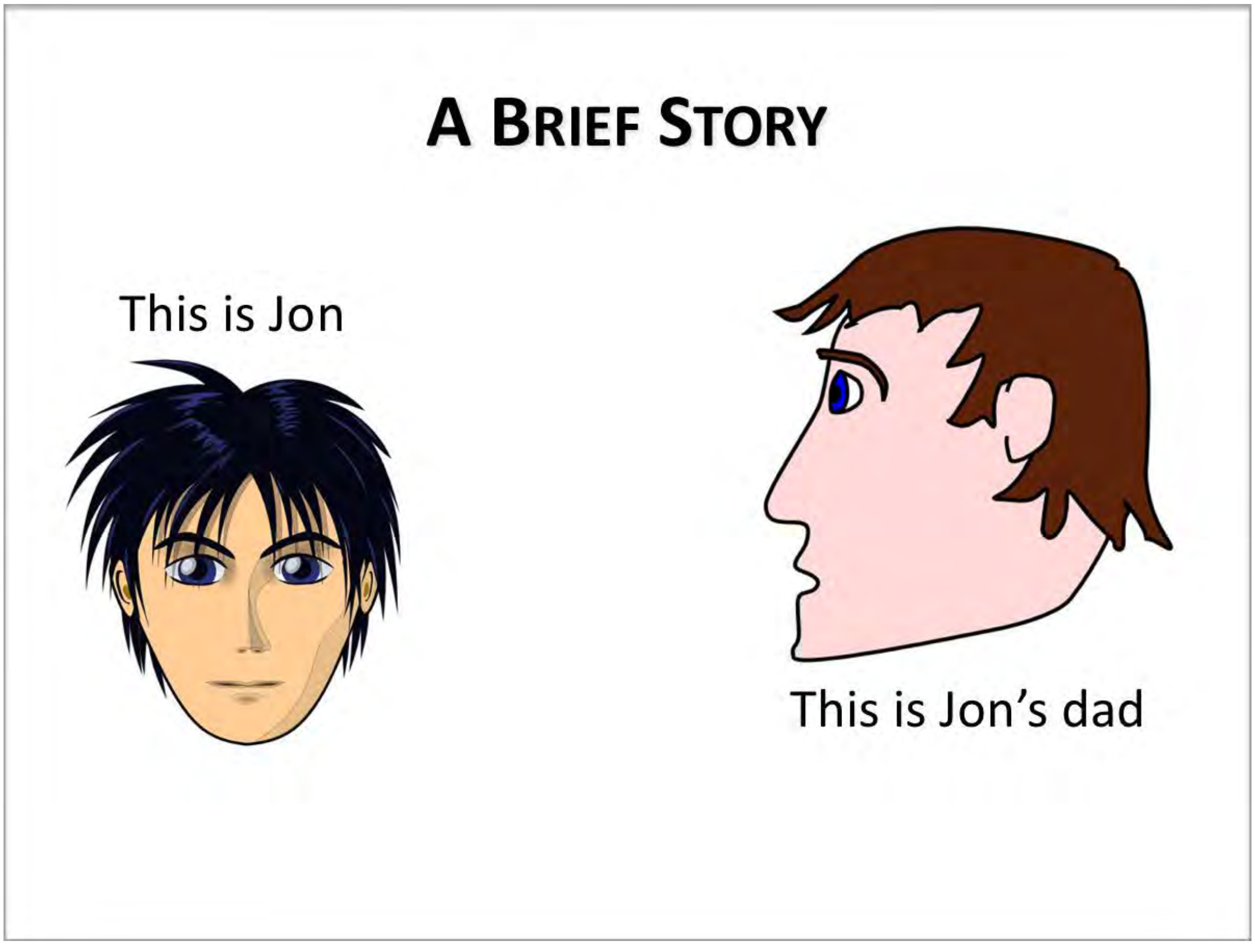
You should also be able to identify some existing notations for expressing assurance cases.

Finally, and perhaps most importantly, you should be able to enumerate characteristics that an assurance case should have. By this, I do *not* mean to be able to list characteristics of a *good* assurance case versus a *bad* assurance case, but simply to be able to list characteristics to distinguish between something that *can be legitimately* called an assurance case, and something that *does not* deserve the name.

In other words, if someone comes to you with a document that they claim is an assurance case, you should be able to read over the document and say, “Yes, it is an assurance case”, or “No, it is not an assurance case.”

Material about how to distinguish between a good case and a bad case will be covered in Module 3 about Evaluation.

[Question for participants: Any questions about these learning objectives?]



A BRIEF STORY

This is Jon

This is Jon’s dad

The majority opinion among educators today is that telling stories is a very good thing; so here’s a story for you.

The young fellow with black hair on the left is called Jon.

The older brown-haired fellow on the right is Jon’s dad. His name is Mike. (Note: the original presentation included automated slides with the images of Jon and Mike speaking the appropriate dialog. Reproducing those slides here is unnecessary.)

One day, Jon comes up to his dad, and says:

“Tim will give me a ride to the game.”

Because Jon’s dad knows nothing about Tim’s driving ability, he asks,

“Is he a safe driver?”

“Yes, he is,” replies Jon.

Not willing to simply trust Jon’s rather information-free assertion, Jon’s dad asks,

“How do you know?”

Jon, perhaps because he’s a tad miffed that his Dad didn’t just accept without question his claim that Tim is a good driver, replies,

“It’s just one of those things I know.”

Jon’s dad, undoubtedly a bit *more* than a tad miffed with this response, tells Jon,

“That’s not good enough. Try again.”

Jon thinks for a little while, and then says,

“No one says he’s ***not*** a safe driver.”

Jon’s dad, wondering how big the ‘no one’ set is, asks Jon,

“How many people have you asked?”

Jon’s reply is a bit disappointing, but not particularly surprising to his dad,

“Um, well, one, but … ” followed by a pause, which eventually ends with Jon continuing,

“He passed the state test to get a license, so he must drive safely.”

Jon’s dad pauses before replying, deciding whether to ask Jon how in the world he thought hat one person was enough to attest to Tim’s driving ability. After a second or two, he decides to let it pass, and instead address the license issue,

“Just being legal doesn’t mean he’s safe.”

At this point, Jon recognizes that he’s totally lost control of the conversation, and asks, exasperatingly,

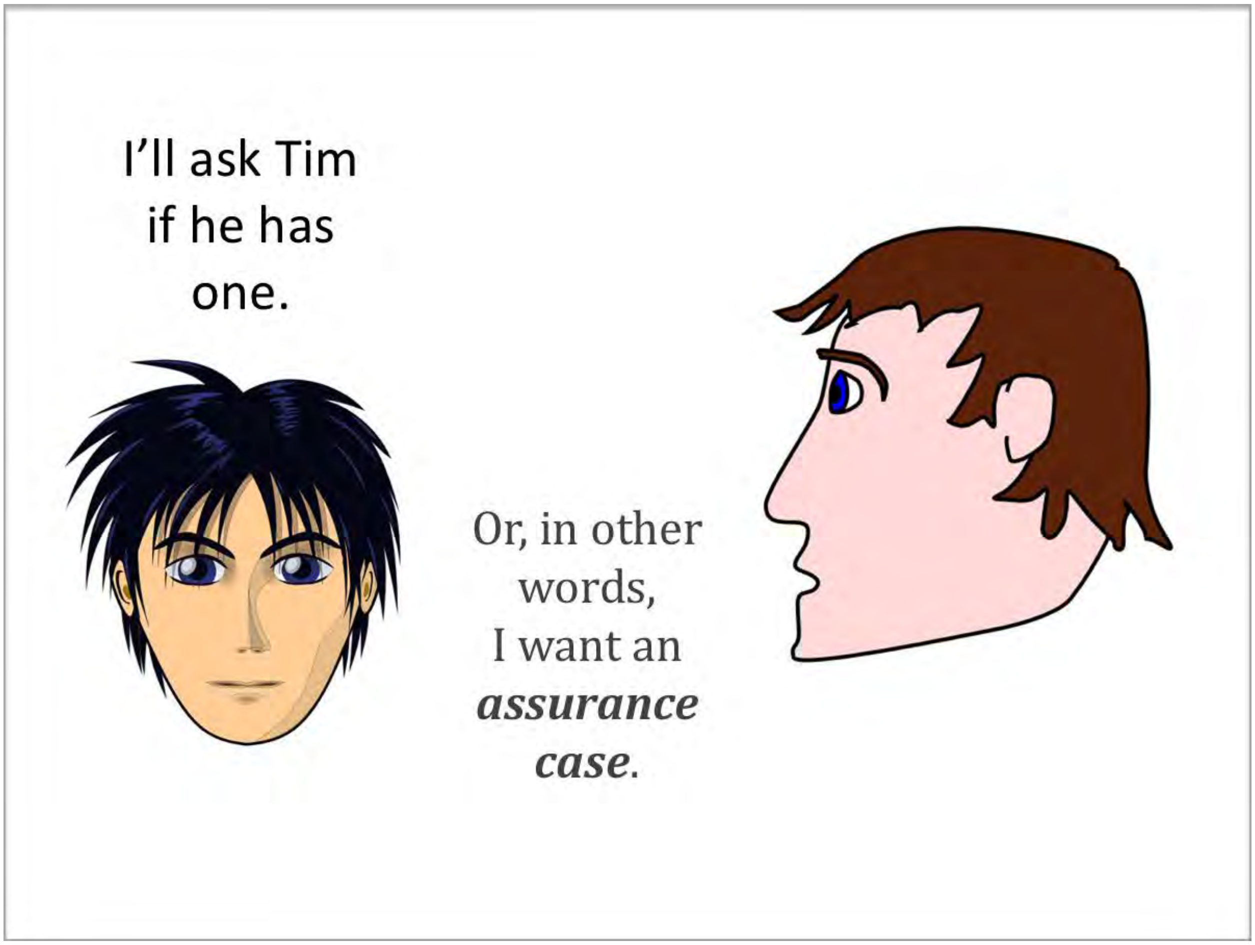
“Well, Dad, what’s gonna convince you to let me ride with Tim?”

Jon’s dad decides to ignore Jon’s not-entirely respectful tone, and simply says,

“Reasons … *Good* reasons to believe Tim will get you there and back in one piece.”

And after a brief pause, he adds, “Or, in other words, I want an *assurance case*.”

After an even longer pause, Jon responds, “I’ll ask Tim if he has one.”

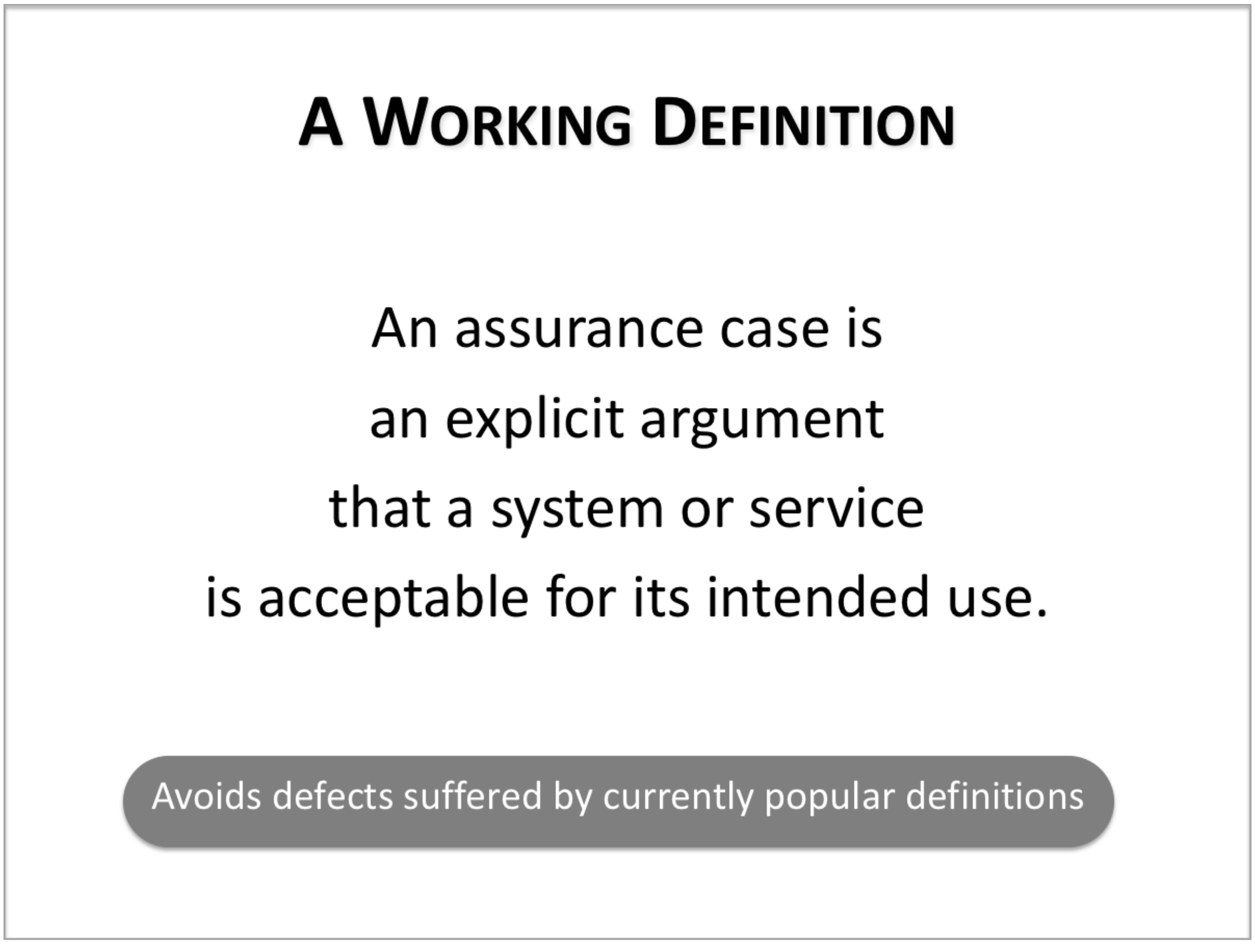

I’ll ask Tim if he has one.
Or, in other words, I want an ***assurance case***.

And thus ends our story, leaving us to wonder: Does Tim have an assurance case? Upon seeing it, will Jon’s dad let him ride with Tim? Who will win the game? Will Jon get home safely? And, what about Naomi?

If you don’t understand that last question, type it into your favorite search engine either before or after the phrase “love of chair.”

Of course this story is a bit silly, and not entirely realistic, but it does illustrate indirectly, and I hope memorably, many of the basic concepts that we’re going to discuss directly now.

So, what is an assurance case?



A Working Definition

An assurance case is
an explicit argument
that a system or service
is acceptable for its intended use.

Avoids defects suffered by currently popular definitions

Here is a working definition: *An assurance case is an explicit argument that a system or service is acceptable for its intended use*.

I call this a ‘working definition’ because it is the definition that we will use throughout this series of educational modules; it captures, I believe, all of the essential elements that distinguish an assurance case from something else[2].

This definition does not exactly match any specific definition currently used commonly in the literature or existing standards and guidelines.

Those definitions, in my opinion, suffer from various defects, and this definition is designed to avoid those defects.

The most common definitions one sees in the literature are definitions derived from early definitions of ‘safety case’. They include in the definition notions of ‘goodness’ that I don’t think appropriately belong, beginning, for example with something like this: “A reasoned and compelling argument …”

[2] The phrase ‘assurance case’ (or ‘safety case’) is used in a variety of ways, not all of which require the existence of an explicit argument. These other uses are not relevant for the purposes of this module. The interested reader can explore the following paper, which was written after these materials were originally created: Graydon, P. J. 2017. “The Safety Argumentation Schools of Thought.” *3rd International Workshop on Argument for Agreement and Assurance (AAA)*. November 13-15, 2017. Tokyo, Japan. Accessed October 11, 2018. http://hdl.handle.net/2060/20180000378.

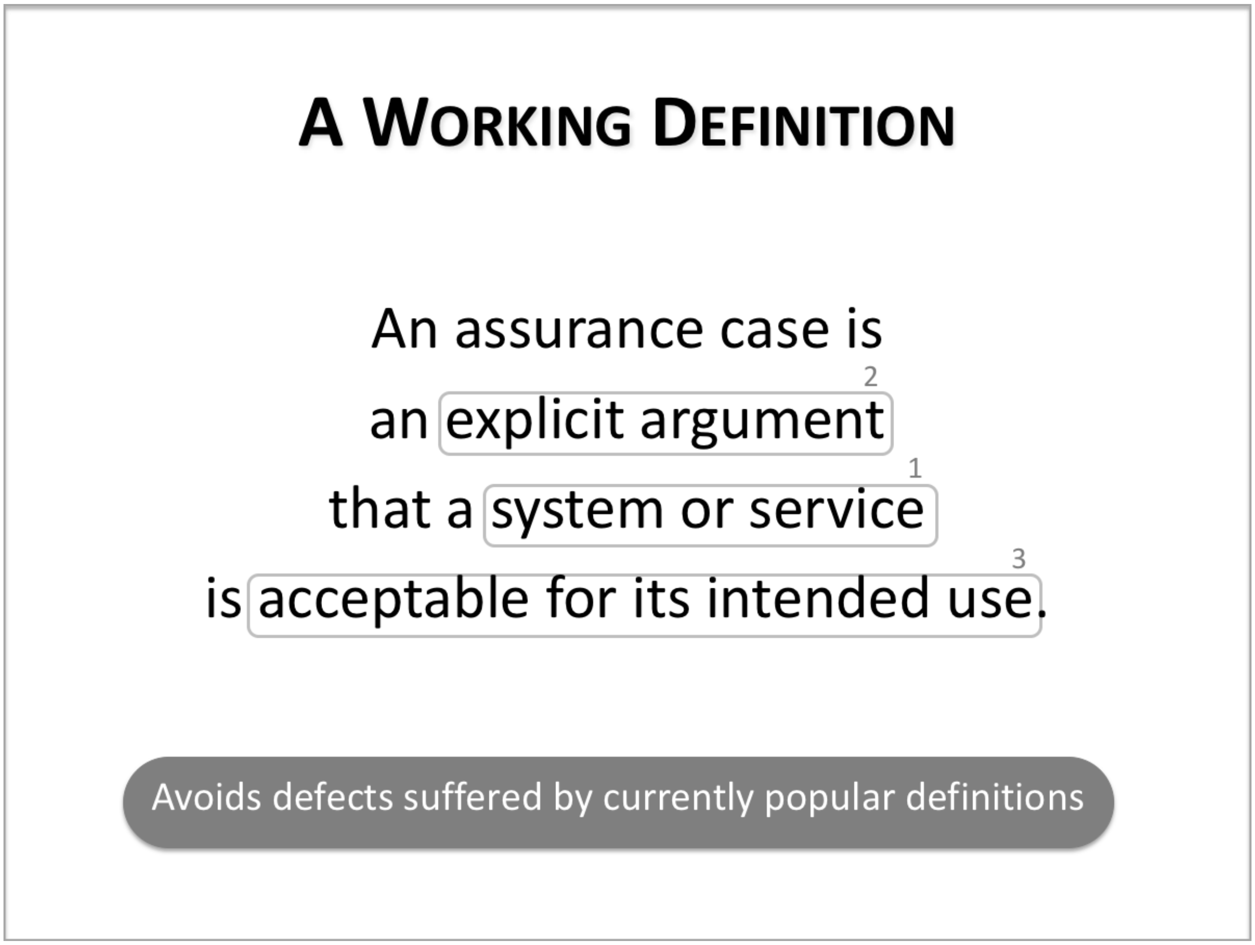
Well, ‘reasoned’ and ‘compelling’ are certainly characteristics one wants in a *good* assurance case, but including them in the basic definition is akin to defining ‘student’ as something like, “an enthusiastic and diligent learner ….” What then do you call folks running about our schools and colleges who are not particularly diligent and perhaps a tad bored?

The definitions currently used in some standards and guidelines also are encumbered with ‘goodness’ ideas, while additionally suffering from verbosity and poor wording choices.

For example, ISO/IEC 15026-1, section 3.1.3 gives this ugly definition of ‘assurance case’: “reasoned, auditable artifact created that supports the contention that its top-level claim (or set of claims), is satisfied, including systematic argumentation and its underlying evidence and explicit assumptions that support the claim(s).” There are also 5 lines of *‘note’* attached to the definition that specify the contents of an assurance case. [ISO/IEC 15026-1:2013, Systems and software engineering - Systems and software assurance - Part 1: Concepts and vocabulary, Switzerland, Geneva.]

The simple definition that you see here does not suffer from any of these defects, so it is the one we’re going to use.

Most of the rest of this presentation will involve discussing further the three main aspects of the definition.



A Working Definition

An assurance case is
an explicit argument
that a system or service
is acceptable for its intended use.

Avoids defects suffered by currently popular definitions

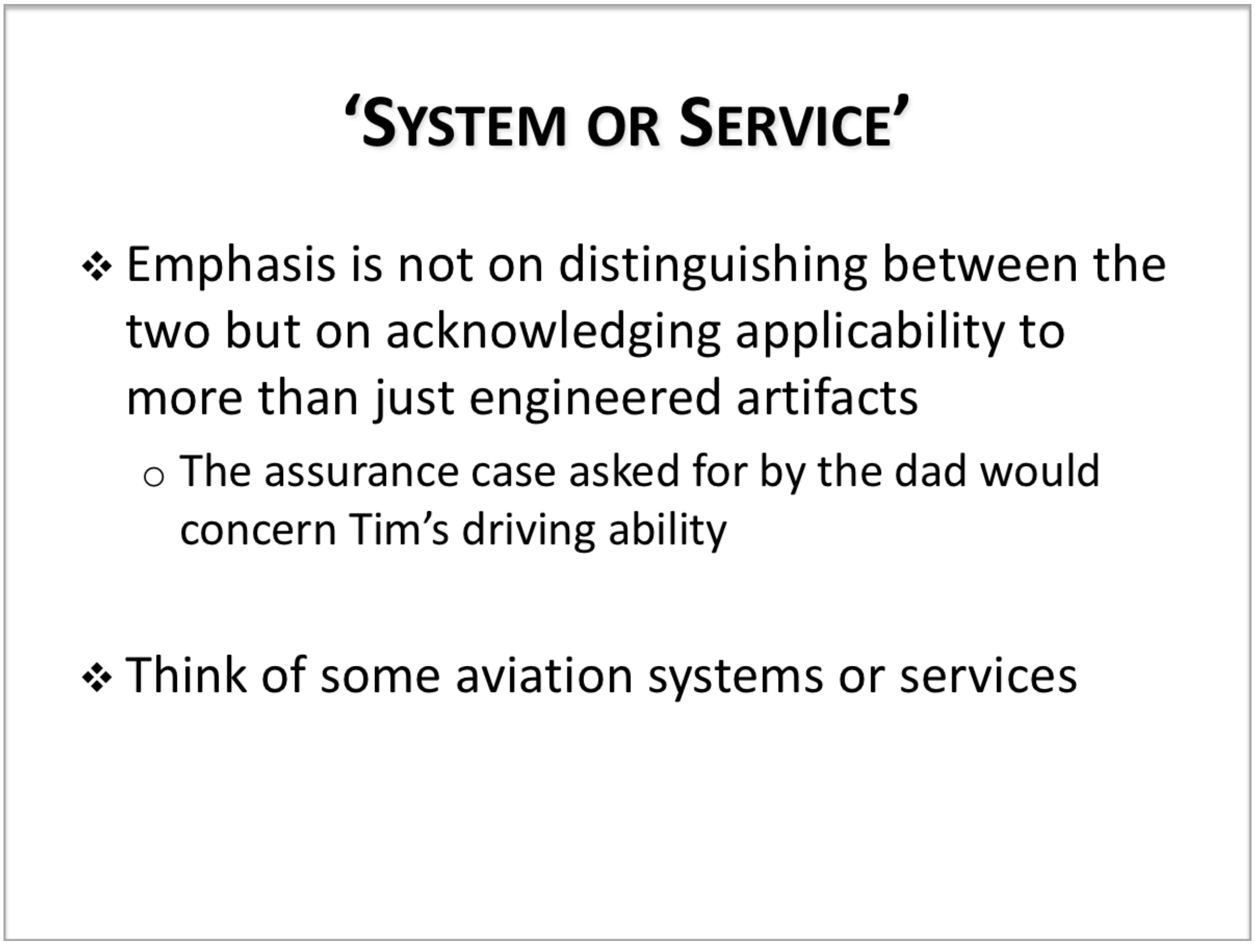
The first aspect we'll discuss, quite briefly in fact using only a single slide, is 'system or service'.

The majority of our time will be spent discussing the second aspect, which is 'explicit argument.' While we're discussing explicit argument, we'll touch on the third aspect: 'acceptable for its intended use.'

We'll then talk a little bit more specifically about this aspect. As you probably have already surmised, One can easily change 'acceptable' to 'safe' to give a definition for safety case, and 'acceptable' to 'secure' to give a definition for security case.

[Question for participants: Does anyone have any questions at this point?]

Before we proceed, I should probably mention that the word in this working definition that would generate the most controversy in a room full of assurance case people is almost certainly the first 'is'. Some people would strongly insist that 'is' *must* be replaced by 'contains'. Their insistence stems from wanting to emphasize that any practical assurance case will need to have additional information besides just the bare argument itself. Items such as a system description, a list of system hazards, and a discussion of the safety management system are among the sorts of information they have in mind. Although I agree that such items usually need to be part of the case, I also believe that they can also legitimately considered to be part of the argument; hence, 'is' is appropriate, and 'contains' unnecessary.



'SYSTEM OR SERVICE'

- ❖ Emphasis is not on distinguishing between the two but on acknowledging applicability to more than just engineered artifacts
  - o The assurance case asked for by the dad would concern Tim's driving ability
- ❖ Think of some aviation systems or services

The use of the phrase ‘system or service’ in the definition is not intended to emphasis distinguishing between the two, but rather to make clear that assurance cases do *not* just apply to engineered artifacts (which is what many people often think of when they hear the word ‘system’). Assurance cases at least as well (and in fact, as we’ll see in Module 2 are perhaps most firmly established) for operational procedures, maintenance activities, collections of ‘best practices’, and other such things that are often not called ‘systems’ but rather ‘services’.

As a specific example, motivated by the tragic, apparently intentional crash in France recently[3], one can imagine an assurance case being developed to cover new rules and procedures for when and how to lock the cockpit door.

I’m sure that all of you can think of many aviation systems or services. And I suspect that an assurance case could be developed for any of them, so long as it is possible to identify what the system or service is intended to provide.

That’s all that I intend to say about ‘system or service’, unless someone has a question.

[Question for participants: Are there any questions about ‘system or service’?]

We’ll move now to talking about ***explicit argument***.

I’ve structured this discussion based on the terminology and ideas described by Stephen Toulmin in his seminal book *The Uses of Argument*. [S. E. Toulmin. 1958. *The Uses of Argument*. Cambridge, UK: Cambridge University Press, updated ed. 2003.], with some changes in terminology I’ve made over the years.

There are plenty of other ways the discussion could be structured, but Toulmin’s ideas have strongly influenced the notion of assurance cases over the years, and I think his general framework is easier to understand than most others, and corresponds more clearly to many people’s intuitive notions; and in those places where his ideas may not be quite so clear, I’ve made some modifications I hope will add clarity.

Not everyone agrees with my opinion. Formalists, for example, tend not to like Toulmin’s ideas very much, and it is safe to say that he was not overly fond of their ideas, either.

The next several slides build on one another. Almost certainly you’ll have questions after seeing one slide, but it is likely that at least some of your questions may be answered on the next slide, or perhaps a couple down the road; for this reason, I’d like for you to hold your questions for a little bit, until I explicitly ask for them. Also, please keep in mind that we’ll first be talking about simple, self-contained arguments, which rarely exist in pure form in the real world. Later on, we’ll talk a bit about real world arguments, which are usually rather messy, and quite complicated.

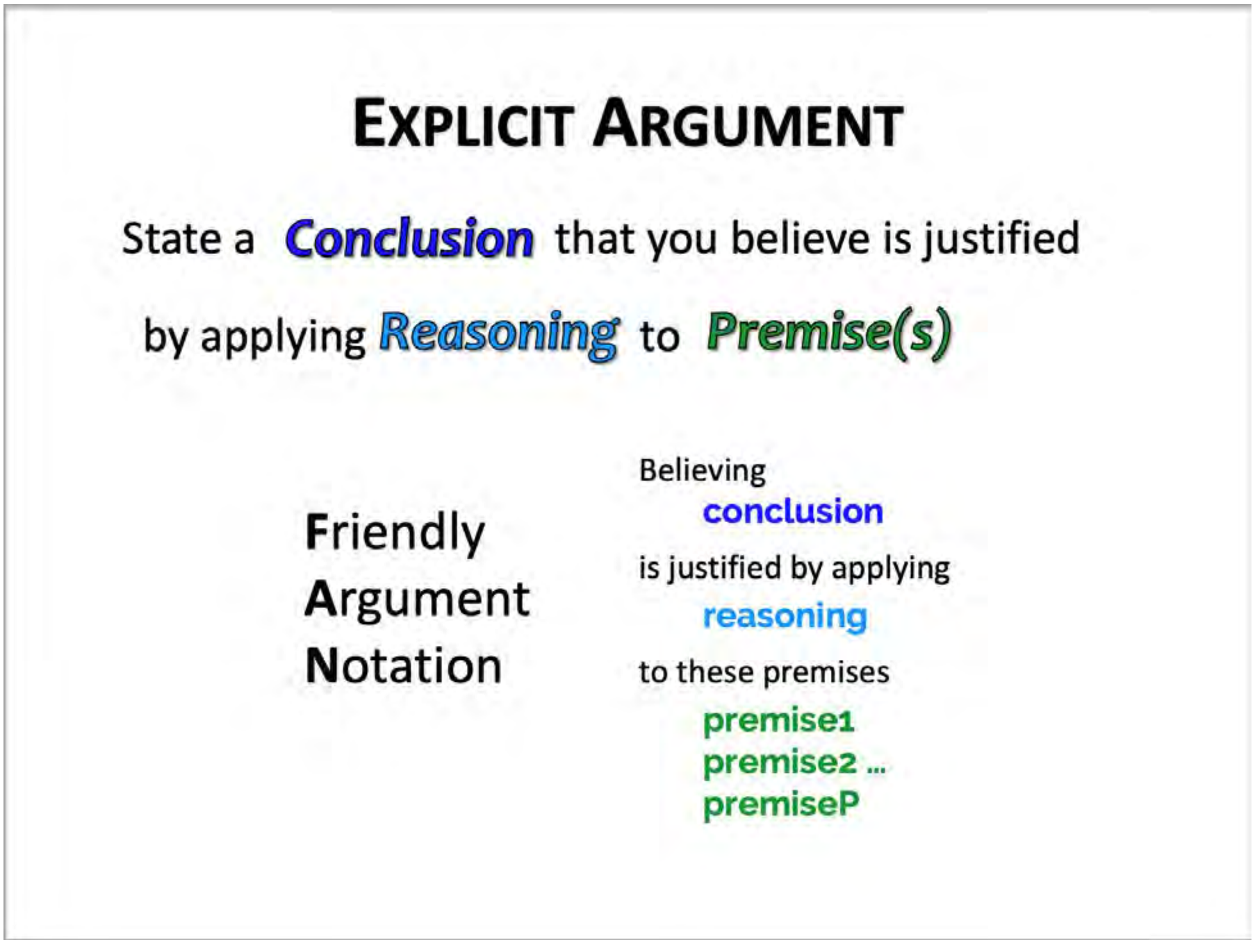
In the following slide, you see a simple form of an explicit argument.

It has three parts: a **conclusion**, one or more **premises**, and **reasoning.**

[3] At the time of the original presentation, the Germanwings flight 9525 crash investigation was not complete, but the evidence was becoming compelling that the co-pilot caused it intentionally.

The arguer (let’s say it is you) wants someone else (let’s say it is me) to believe that a particular **conclusion** is true.

To convince me, you’ll give me some **premises**, which are statements that you are confident that I will accept as true, and explain your **reasoning** why the truth of the **premises** is sufficient to justify the truth of the **conclusion**.



EXPLICIT ARGUMENT

State a **Conclusion** that you believe is justified

by applying **Reasoning** to **Premise(s)**

Friendly
Argument
Notation

Believing
conclusion
is justified by applying
reasoning
to these premises
premise1
premise2 …
premiseP

Or, to put this into what I call the Friendly Argument Notation (FAN), you might write Believing **conclusion** is justified by applying **reasoning** to these **premises**.

The **conclusion** is *what* you want me to believe.

The **premises** are things that you think I *already* believe.

The **reasoning** explains *why* taking the step from believing the **premises** to believing the **conclusion** is an ‘appropriate and legitimate one’ (to use Toulmin’s language).

Here are three examples …

Given (**premise**) “*Annette was born in Lynchburg, Virginia”* you should believe (**conclusion**) “*Annette is a US citizen”* because (**reasoning**) “*People born in Virginia are US citizens*.”

Why is the fact that Annette was born in a city in Virginia enough to justify belief that she is a US citizen? The **reasoning** provides the answer to the question.

Second example.

Given (**premise** 1) "*A = B*" and (**premise** 2) "*B = C*" you should believe (**conclusion**) "*A = C*" because of (**reasoning**) "*the transitive property of equality.*"

Why is A=B and B=C enough to justify believing A=C? The transitive property of equality explains.

And finally, one more example, In which I'll use a different ordering of elements, you should believe *"Angela is eligible to run for President of the US*," given these three **premises***: "Angela is a natural born US citizen," "Angela is 54 years old,"* and *"Angela has lived in the US all her life."*

Why? Because of this **reasoning***: "The eligibility requirements of Article II Section 1 of the Constitution are natural born citizen, at least 35 years old, and having lived in the US for 14 years".*

I'm sure you can think of many examples of your own.

You may also be able to think of different names that you've heard given to the three parts: **conclusion**, **premises**, and **reasoning**.

Toulmin himself tended to refer to **premises** as 'data', the **conclusion** as a 'claim', and **reasoning** as the 'warrant'.

Other terms used for concepts similar to **reasoning** include simply 'reasons', quite confusingly the word 'argument' itself, and (as just noted) 'warrant'.

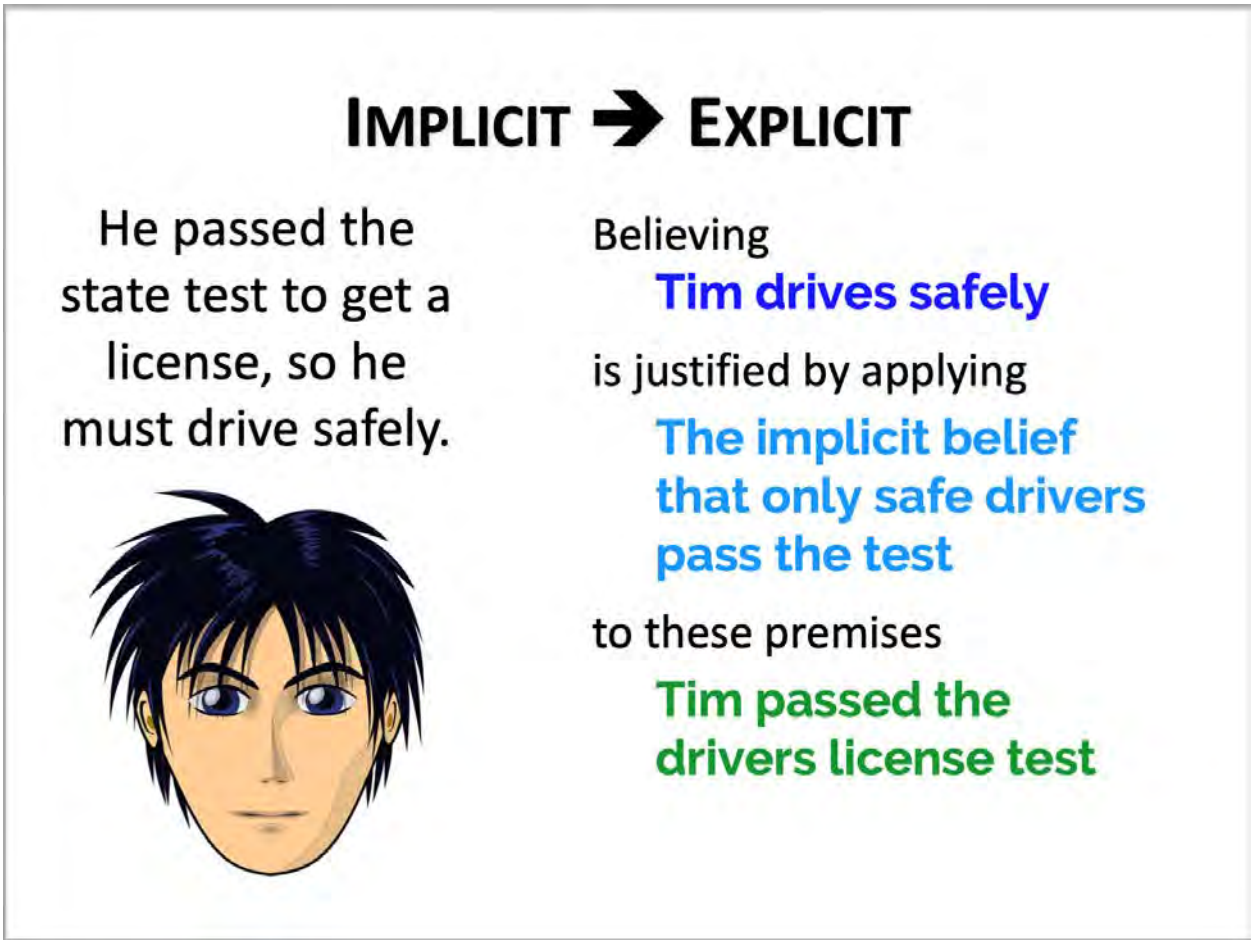
We'll talk more about alternate terms a bit later, and I'll mention my reasons for preferring **premise**, **conclusion**, and **reasoning.**

[Question for participants: Does anyone have any questions about what these three terms mean?]

So far I've concentrated on the 'argument' part of 'explicit argument.'

Real life, however, is full of arguments in which at least one of the parts is implicit rather than explicit. We can see a simple example by returning to our story.

Recall that Jon told his dad Tim had passed the state test to get a license.



IMPLICIT → EXPLICIT

He passed the state test to get a license, so he must drive safely.

Believing

Tim drives safely

is justified by applying

The implicit belief that only safe drivers pass the test

to these premises

Tim passed the drivers license test

In the context of the story, Jon’s statement about Tim passing the state test, can be seen as part of an implicit argument.

The **conclusion**, which Jon wants his dad to believe, is that “*Tim drives safely*.”

In this snippet of the conversation, Jon gives a **premise** for this **conclusion**, namely “*Tim passed the test.”*

He is implicitly applying the **reasoning** “*Only safe drivers pass the test*” to the “*Tim passed the test*” **premise** to justify belief in the **conclusion**.

Jon’s dad is not swayed by this argument, because he (correctly) does not accept the implicit **reasoning**.

That’s an example of the **reasoning** being implicit, which is a situation that is quite common, so common in fact that many approaches to teaching argumentation do not directly address the concept of **reasoning** directly at all, but rather fold it into their discussion of **premises**.

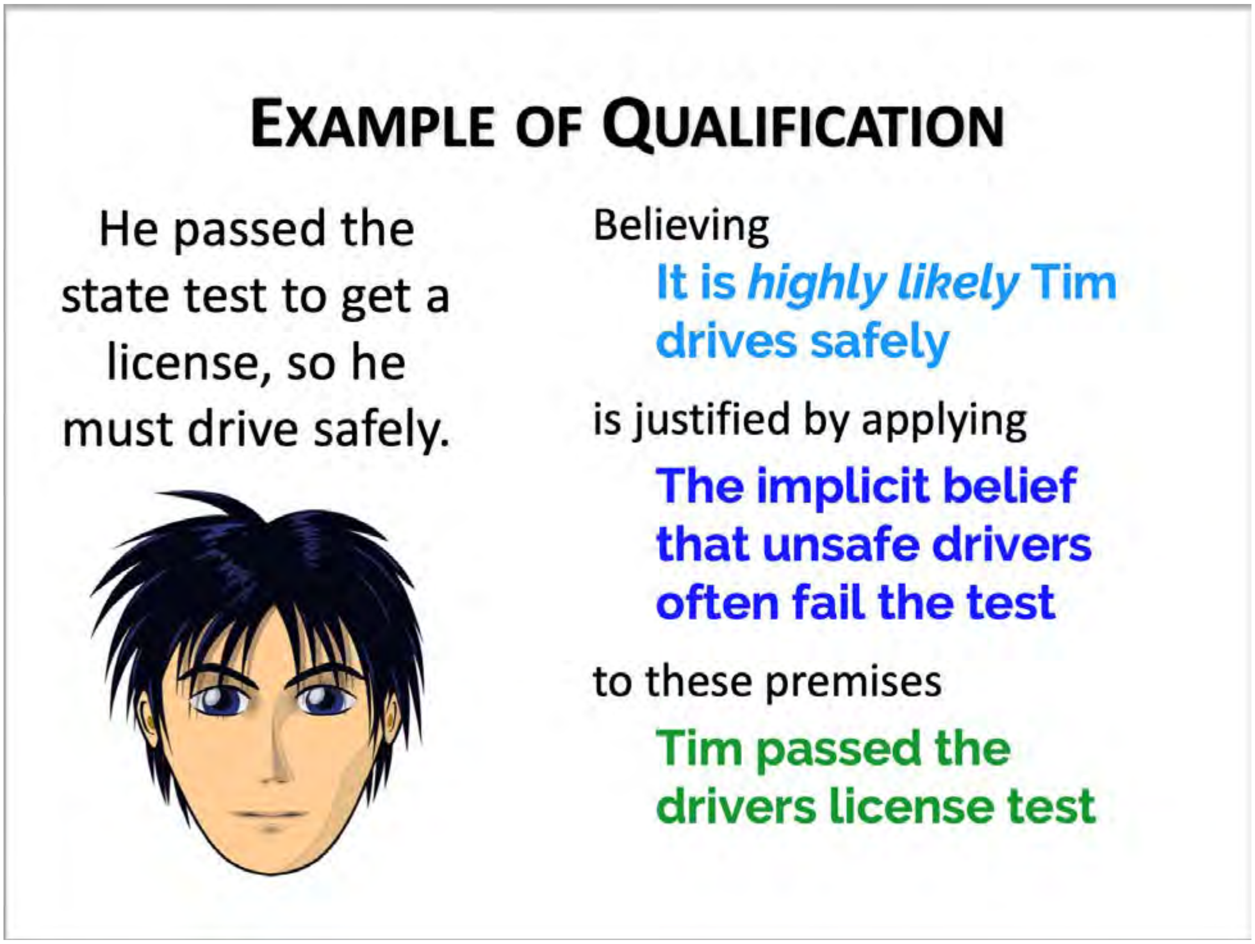
We could also give examples in which one or more of the **premises** is implicit, or even in which the **conclusion** is implicit, or at least not stated specifically. All of these situations of implicitness seem to be fairly common in certain aspects of engineering practice today.

One of the major distinguishing factors of an assurance case approach is the explicit statement of a top level **conclusion**. The explicit statement of a **conclusion** to be established makes it possible for an assurance case to articulate an argument with that same **conclusion**.

As you already may be thinking, the very simple form of argument that we've seen so far may be a bit too simple. You are right.

Often the **conclusion** may need to be expressed in less than absolute terms with **qualification**. That is, the arguer may not be asserting that the **conclusion** is necessarily always and certainly true given the **premises** and the **reasoning**.

Returning to the story, perhaps Jon's implicit argument is really more something like what is shown here on the slide.



Example of Qualification

He passed the state test to get a license, so he must drive safely.

Believing

It is *highly likely* Tim drives safely

is justified by applying

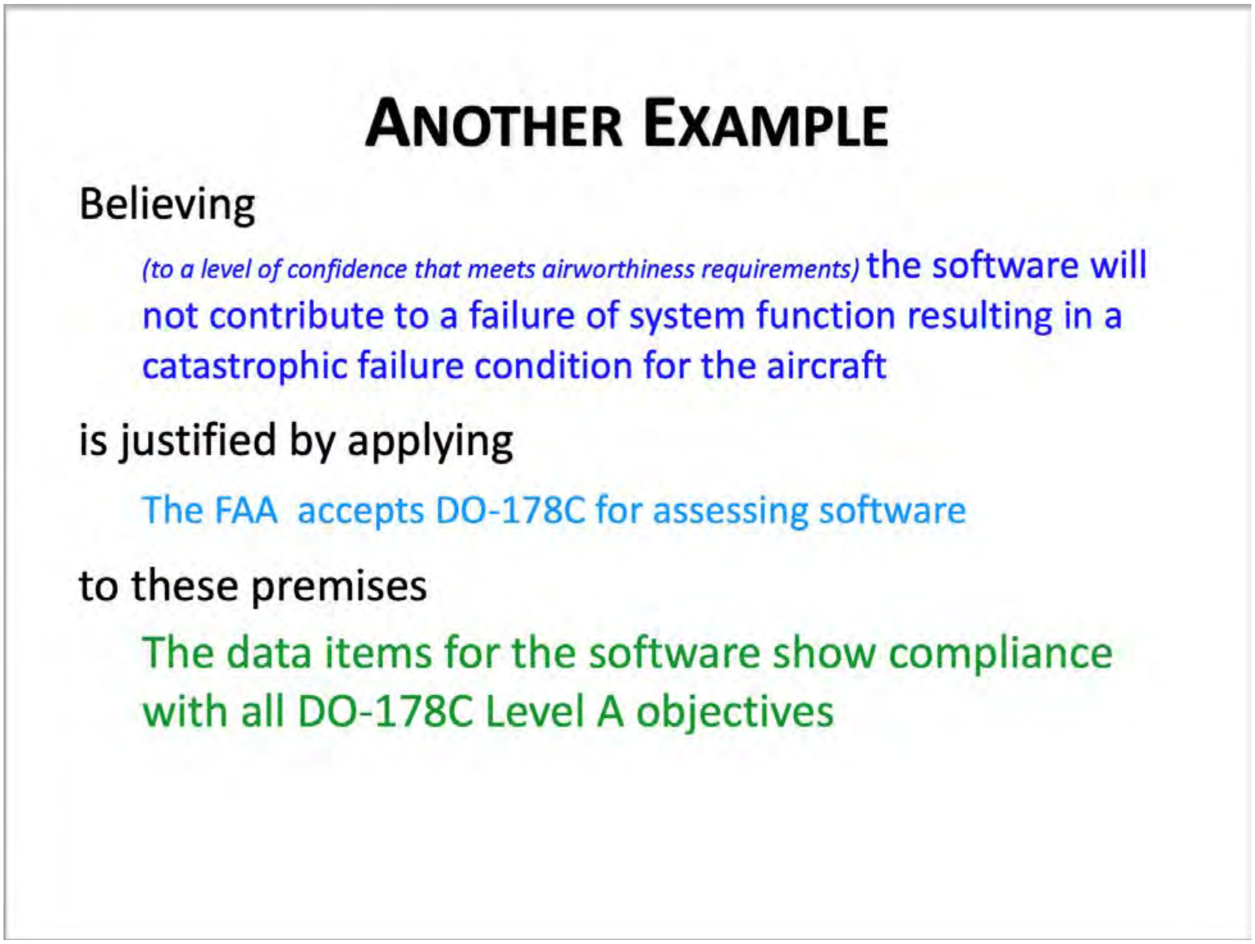
The implicit belief that unsafe drivers often fail the test

to these premises

Tim passed the drivers license test

Given that Tim passed the drivers' license test (the **premise**), and (implicitly) knowing (the **reasoning**) that unsafe drivers often fail the test, then it is highly likely (the **qualification**) (but not necessarily certain) that Tim drives safely (the **conclusion**).

Or here's another example that is a bit more technical. People without knowledge about how software is approved for use on civil aircraft will have to take my word that the example is realistic.



ANOTHER EXAMPLE

Believing

(to a level of confidence that meets airworthiness requirements) the software will not contribute to a failure of system function resulting in a catastrophic failure condition for the aircraft

is justified by applying

The FAA accepts DO-178C for assessing software

to these premises

The data items for the software show compliance with all DO-178C Level A objectives

For level A software on a civil aircraft, one of the **conclusions** that we want to be able to believe (here's the **qualification**) "*to a level of confidence that meets airworthiness requirements*" is that "*The software will not contribute to a failure of system function resulting in a catastrophic failure condition for the aircraft*."

Speaking in fairly simple terms, a primary **premise** that is often used to justify this *conclusion* can be said to be "*The data items for the software show compliance with all DO-178C Level A objectives*."

Why does this **premise** justify the **conclusion**?

Because of the **reasoning**: "*The FAA accepts DO-178C for assessing software*."

I realize that this example oversimplifies reality a bit, so please don't dissect it too much at this point; it is just intended as an illustrative example of how **conclusions** may need **qualification**.

It also may prompt some of you to think that this model of argument may still be incomplete. Once again, you are correct.

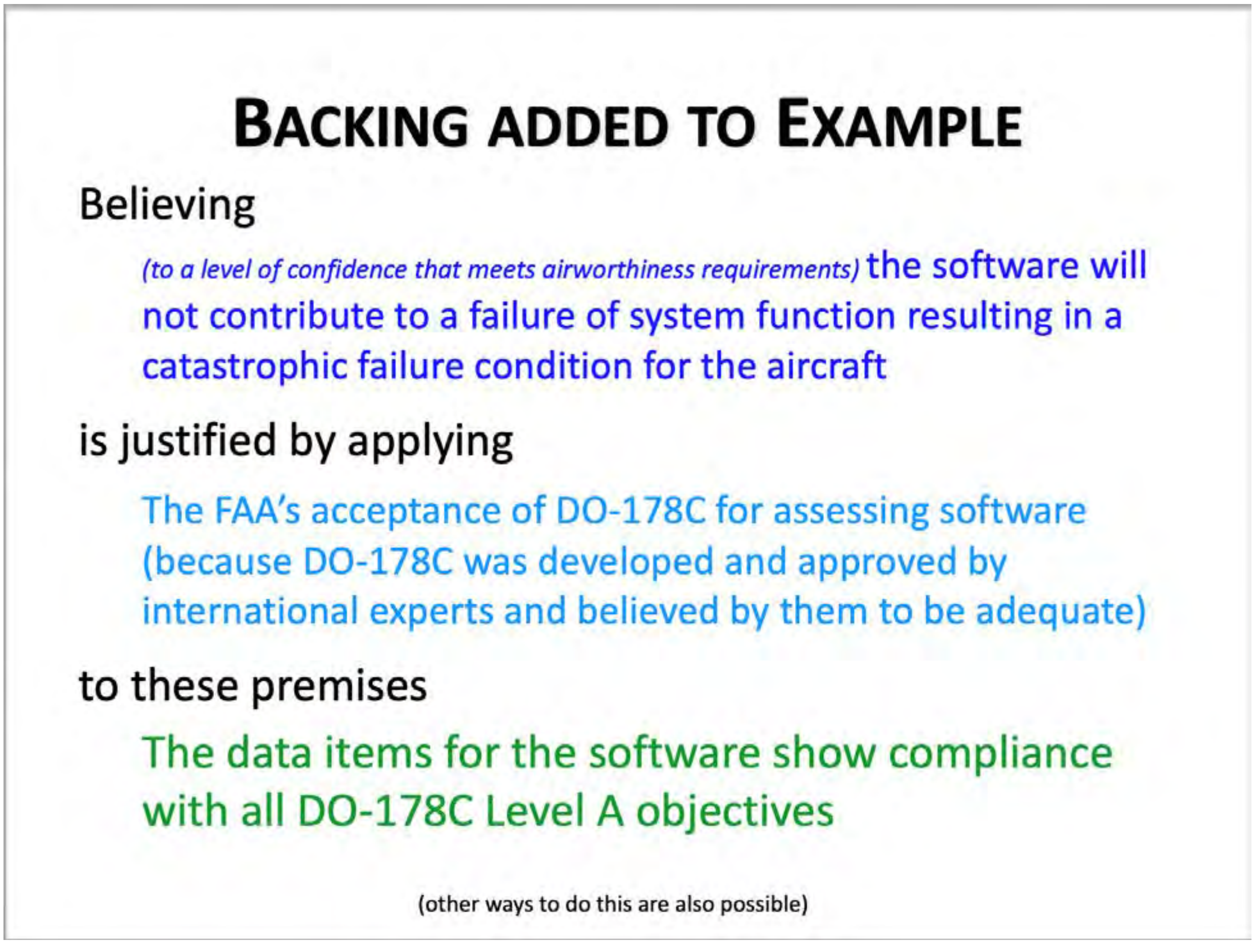
One of Toulmin's insights was the recognition that sometimes it is not possible to state the **reasoning** in such a way as to encapsulate all that's necessary to explain why the **reasoning** justifies accepting the **conclusion** based on the **premises**. Something additional may be needed.

I'll explain this something additional by quoting Toulmin, making minor changes to match our slightly different terminology.

> "In defending a **conclusion**, we may produce our **premises**, our **reasoning**, and the relevant **qualification** and yet find that we have still not satisfied our challenger; for he may be dubious not only about this particular argument but about the more general question whether the **reasoning** is acceptable at all."
>
> "Presuming the general acceptability of this **reasoning** (he may allow) our argument would no doubt be impeccable.... But does not that **reasoning** in its turn rest on something else?"
>
> "Standing behind our **reasoning** there will normally be other assurances, without which the **reasoning** themselves would possess neither authority nor currency. These other things we may refer to as the ***backing*** of the **reasoning**."

We're going to use the term **backing**, too, although we're going to ignore some details and distinctions that Toulmin makes in his book. (Recall that I said that our argument discussion was based on Toulmin's ideas, not that it would be identical to them.)

For our purposes, you can think of **backing** as explaining why the **reasoning** applies or, if you prefer a slightly different wording, reasons for accepting the **reasoning**.



BACKING ADDED TO EXAMPLE

Believing

(to a level of confidence that meets airworthiness requirements) the software will not contribute to a failure of system function resulting in a catastrophic failure condition for the aircraft

is justified by applying

The FAA's acceptance of DO-178C for assessing software (because DO-178C was developed and approved by international experts and believed by them to be adequate)

to these premises

The data items for the software show compliance with all DO-178C Level A objectives

(other ways to do this are also possible)

Here is a statement of **backing** added to the argument you just saw. It asserts that we can accept the **reasoning** on account of the fact that "*DO-178C was developed and approved by international experts and believed by them to be adequate*."

Again, this is just an example. I think it is a fairly realistic example, but I've not been as careful in the wording as would be necessary to turn this into something more than just an example. Of course, there are other (I tend to think, better) ways to incorporate backing an argument, but we'll leave discussing them until another day.

We're almost done with the framework, but not quite. There's one more element of argument that we need to mention.

But before we do that, I'll pause to give you a chance to ask questions.

The final element of argument we will discuss is the notion of **defeaters**, which deals with circumstances in which the general authority of the **reasons** to justify the **conclusion** must be set aside.

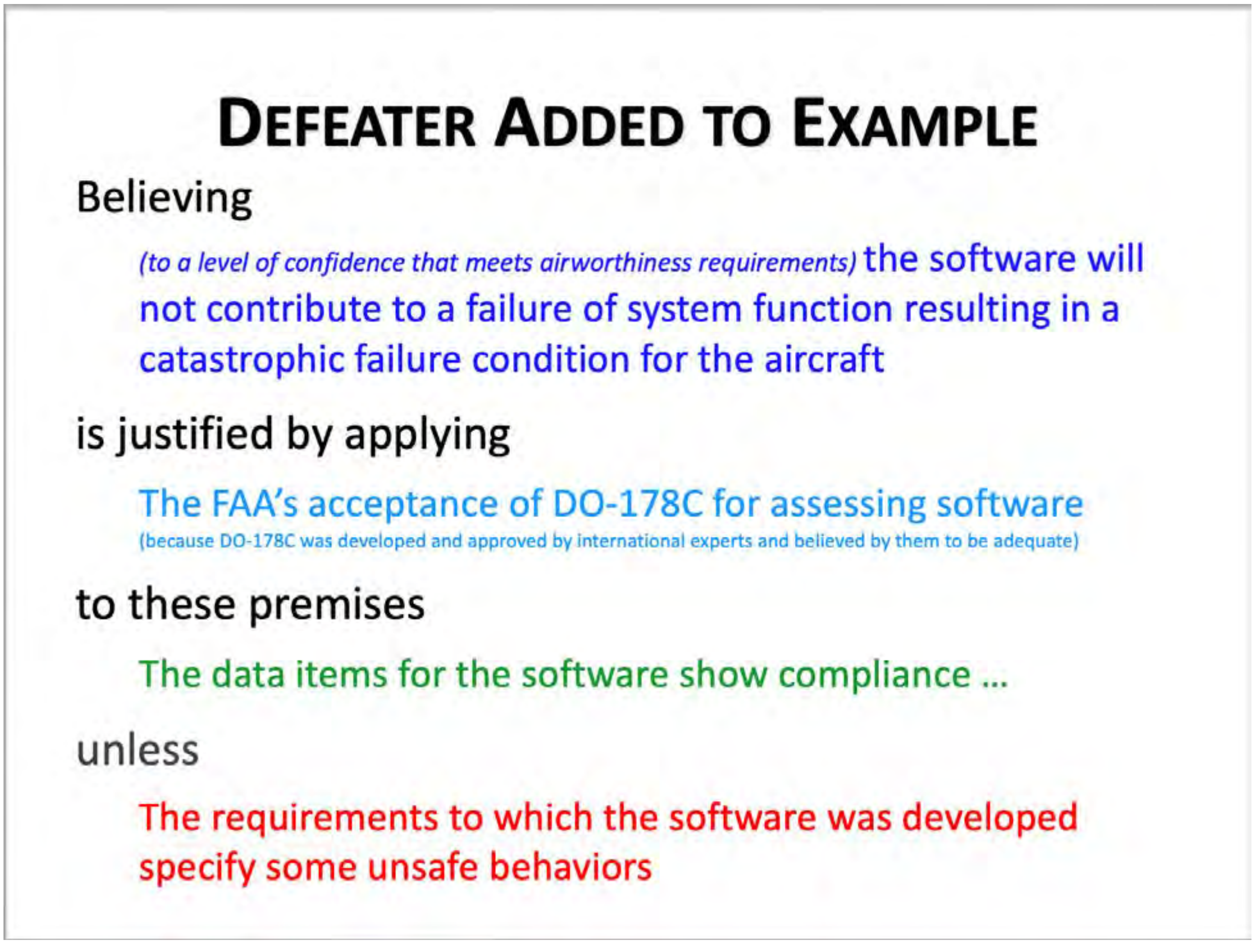
An example or two should help make the concept clear.

Think back to the simple example argument I gave earlier about Angela being eligible to run for President. It had **premises** about her place of birth, her age, and the length of her residency in the US, and the **reasons** referred to the eligibility requirements established in the US Constitution.

A **defeater** is "*Angela has already been elected twice to the office of President*."

In such a case, Section 1 of the 22nd amendment makes her ineligible, despite her meeting the standard eligibility requirements; the **reasoning** that usually justifies the **conclusion** does not do so in this special case.

We can also expand the DO-178 example to include a possible **defeater**.



DEFEATER ADDED TO EXAMPLE

Believing

(to a level of confidence that meets airworthiness requirements) the software will not contribute to a failure of system function resulting in a catastrophic failure condition for the aircraft

is justified by applying

The FAA's acceptance of DO-178C for assessing software

(because DO-178C was developed and approved by international experts and believed by them to be adequate)

to these premises

The data items for the software show compliance …

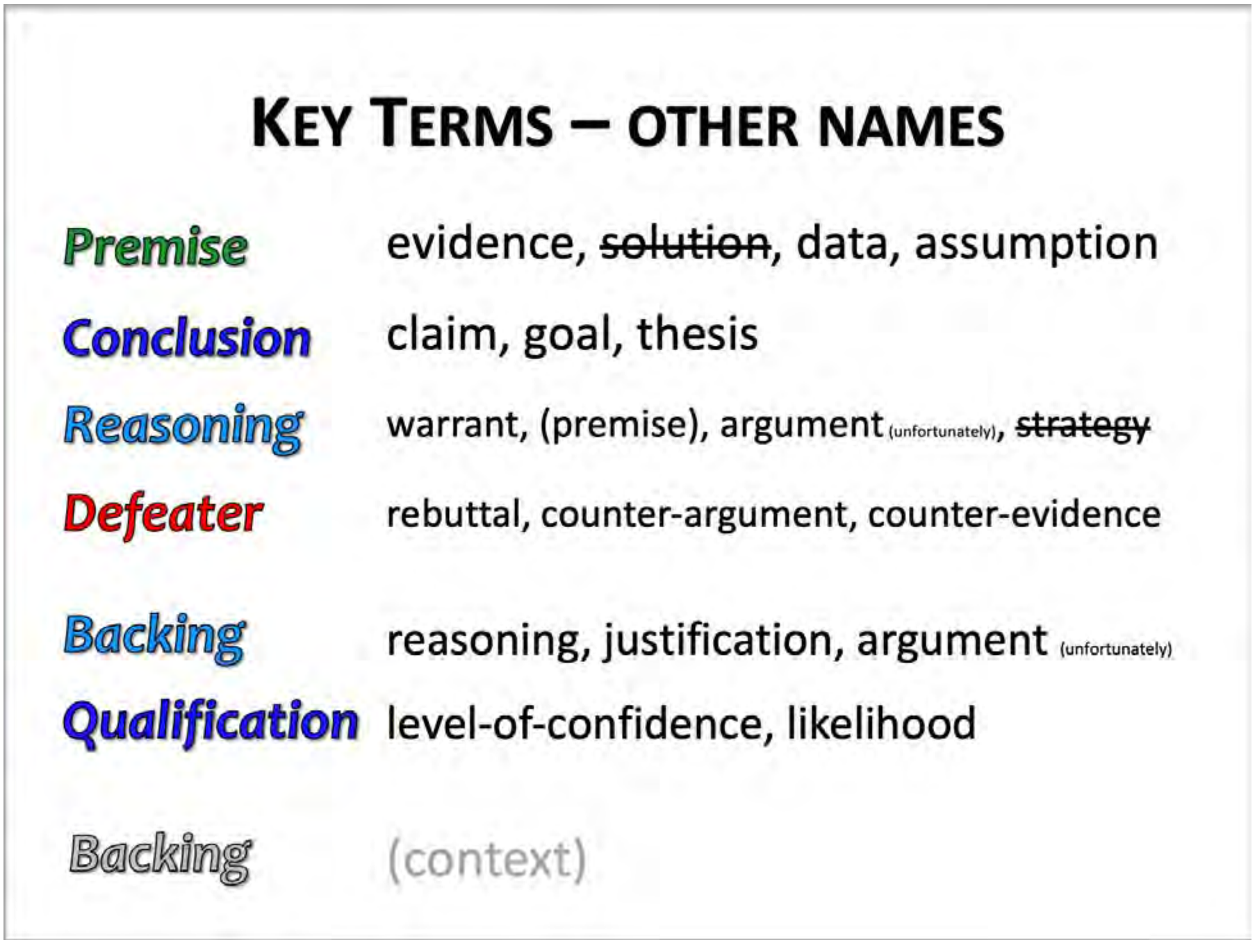
unless

The requirements to which the software was developed specify some unsafe behaviors

Because DO-178's guidance is based on the assumption (which we haven't stated in the argument) that the system safety process has created requirements that, if satisfied, will ensure safety, the argument that we've given so far is also based on that (unstated) assumption.

Thus, if, for a particular instance of software, "*The requirements to which the software was developed specify some unsafe behaviors*", then the argument no longer holds water: it does not establish the truth of the **conclusion**. It has been defeated.

One more slide, and then I'll pause again for questions.



KEY TERMS – OTHER NAMES

Premise	evidence, ~~solution~~, data, assumption
Conclusion	claim, goal, thesis
Reasoning	warrant, (premise), argument (unfortunately), ~~strategy~~
Defeater	rebuttal, counter-argument, counter-evidence
Backing	reasoning, justification, argument (unfortunately)
Qualification	level-of-confidence, likelihood
Backing	(context)

So far, I've specifically introduced six concepts that make up an argument: **premise**, **conclusion**, **reasoning**, **defeater, qualification**, and **backing**.

As we've talked about these concepts, I've mentioned some of the other names used for the concepts; this slide lists the most popular alternative terms.

Within the assurance case community, the most common terms tend to be *evidence* (instead of **premise**), *claim* or *goal* (instead of **conclusion**) and (confusingly) *argument* (instead of **reasoning**).

One of the two most popular notations for expressing assurance cases is called claims-arguments-evidence (or CAE), where *claims* are pretty much the same as **conclusions**, *arguments* are very similar to **reasoning** (or perhaps **reasoning** plus **backing**), and *evidence* is equivalent to certain types of **premises** (more on that later).

The other most popular notation (the Goal-Structuring Notation – GSN) uses the terms *goal*, *strategy*, and *solution*, with *goal* being pretty much equivalent to **conclusion**, *solution* being generally equivalent to certain types of **premises**, and *strategy* serving a role somewhat analogous to **reasoning** and **backing**, though not exactly like it.

The OMG's Structured Assurance Case Metamodel S-A-C-M talks about *claims*, *arguments*, *evidence*, and *reasoning*, among other terms.

I personally think the community is not best served by some of these choices of terminology. Particularly unfortunate in my opinion is the overloading of the term *argument* to refer not only to the overall argument, but also to that ***part*** of the argument that links **premises** with ***conclusions***. It is much clearer and less confusing to use **reasoning** (and, if needed, **backing**) for that part.

Conclusion is a better term than *goal* or *claim* in my opinion because it does not carry the potentially negative connotations that can be associated with those terms. *Claim*, in particular, tends to suggest to some people, myself included, something that is *asserted* to be true, but in reality is most likely *not* true. Consider, for example, the sentence, "My daughter claimed she did her homework last night." Do you think the daughter did her homework?

In saying that I'm not fond of the common terminology, I'm not saying that there are no legitimate reasons that this terminology was chosen and continues to be used; there are reasons (based on analogies to some other disciplines, for example), which are deemed more than adequate, by plenty of folks, so perhaps I've made a bigger deal out of this than I should, but I don't think so.

You see here at the bottom of the slide two additional terms that are often important in practice important but which are not explicitly part of the Toulmin-based argument model: **bindings** (which is the term I prefer) and the somewhat analogous GSN term **context**. For now all you need to do is remember that these terms exist. I'm not going to talk about these anymore in this module.

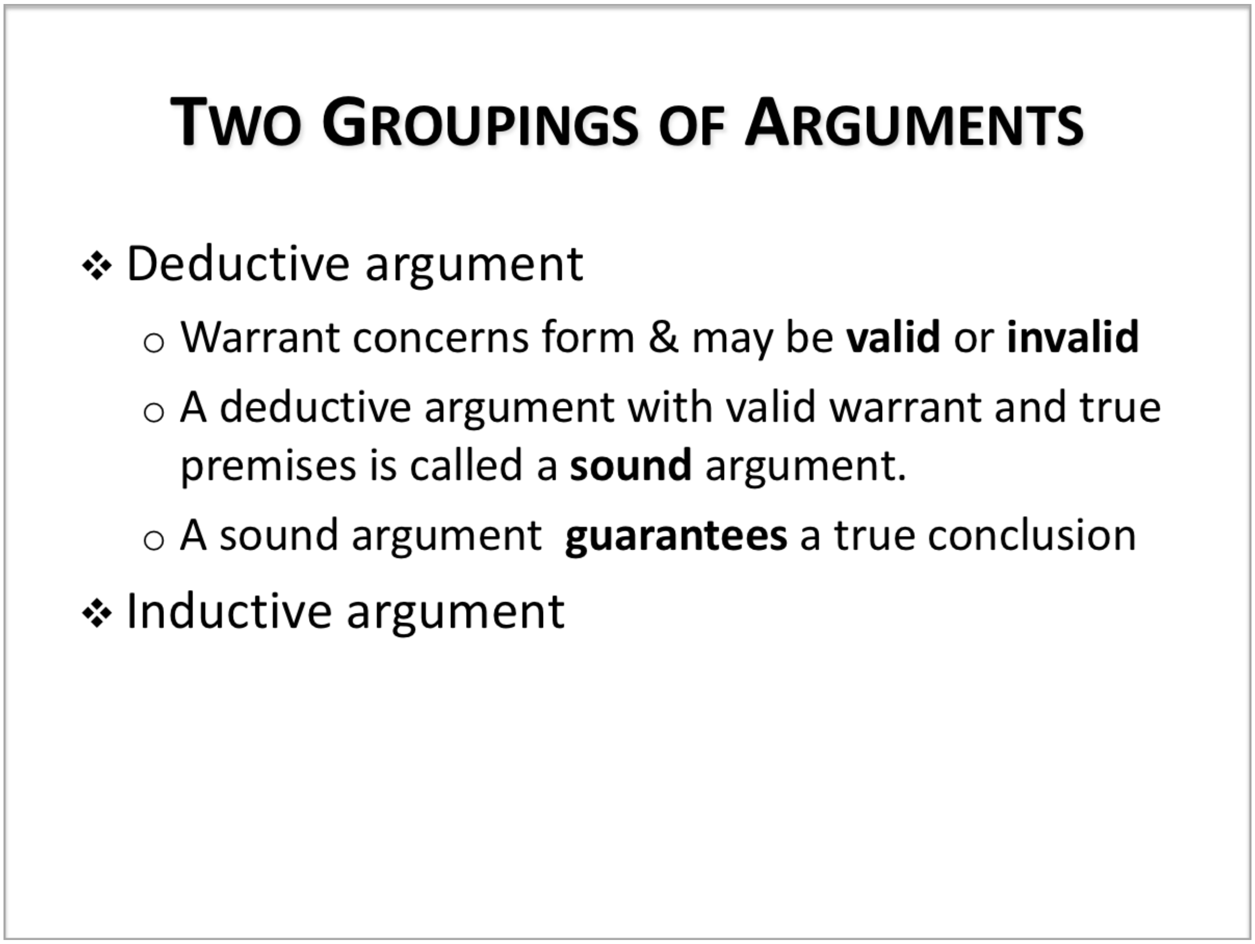
We'll talk about all these terms quite a bit more in the future, particularly in the Evaluation and Creation modules.

Right now, however, I want to stop to take questions, of which I'm sure that are several.

[Question for participants: What are your questions?]

We've looked at the elements that make up arguments. Now we need to talk a little bit about types of arguments.

For our purposes, arguments can be grouped into two categories: *deductive* arguments and *inductive* arguments.



Two Groupings of Arguments

- Deductive argument
  - Warrant concerns form & may be **valid** or **invalid**
  - A deductive argument with valid warrant and true premises is called a **sound** argument.
  - A sound argument **guarantees** a true conclusion
- Inductive argument

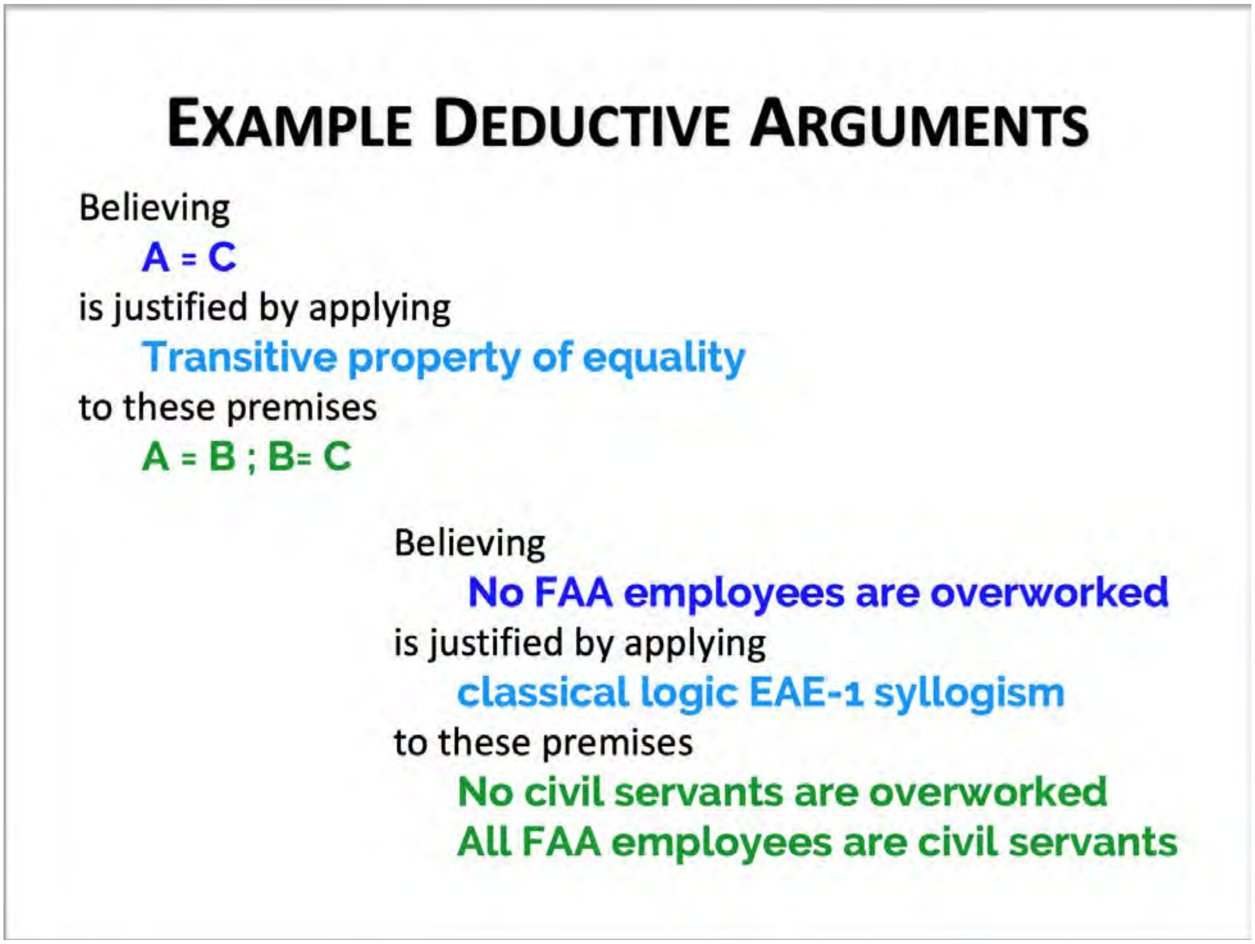
In a deductive argument, the **reasoning** is about the form of the argument. It can be either *valid* or *invalid*.

If a deductive argument has valid **reasoning**, and true **premises**, it is called a *sound* argument.

A sound argument *guarantees* a true **conclusion**.

Or in other words, it is *not possible* for a deductive argument to have valid **reasoning**, true **premises**, and a false **conclusion**.

Here are two examples of deductive arguments.



Example Deductive Arguments

Believing

A = C

is justified by applying

Transitive property of equality

to these premises

A = B ; B= C

Believing

No FAA employees are overworked

is justified by applying

classical logic EAE-1 syllogism

to these premises

No civil servants are overworked

All FAA employees are civil servants

The first one I mentioned early on in our discussion about the elements of argument. It is a simple instantiation of the transitive property of equality.

The second example is new. It asserts the following: Given (**premise** 1) "*No civil servants are overworked*" and (**premise** 2) "*All FAA employees are civil servants*" we should believe the **conclusion** that "*No FAA employees are overworked*" because the form (**reasoning**) is a *"EAE-1 syllogism*" from classical logic, which is known to be one of the valid forms of syllogisms.

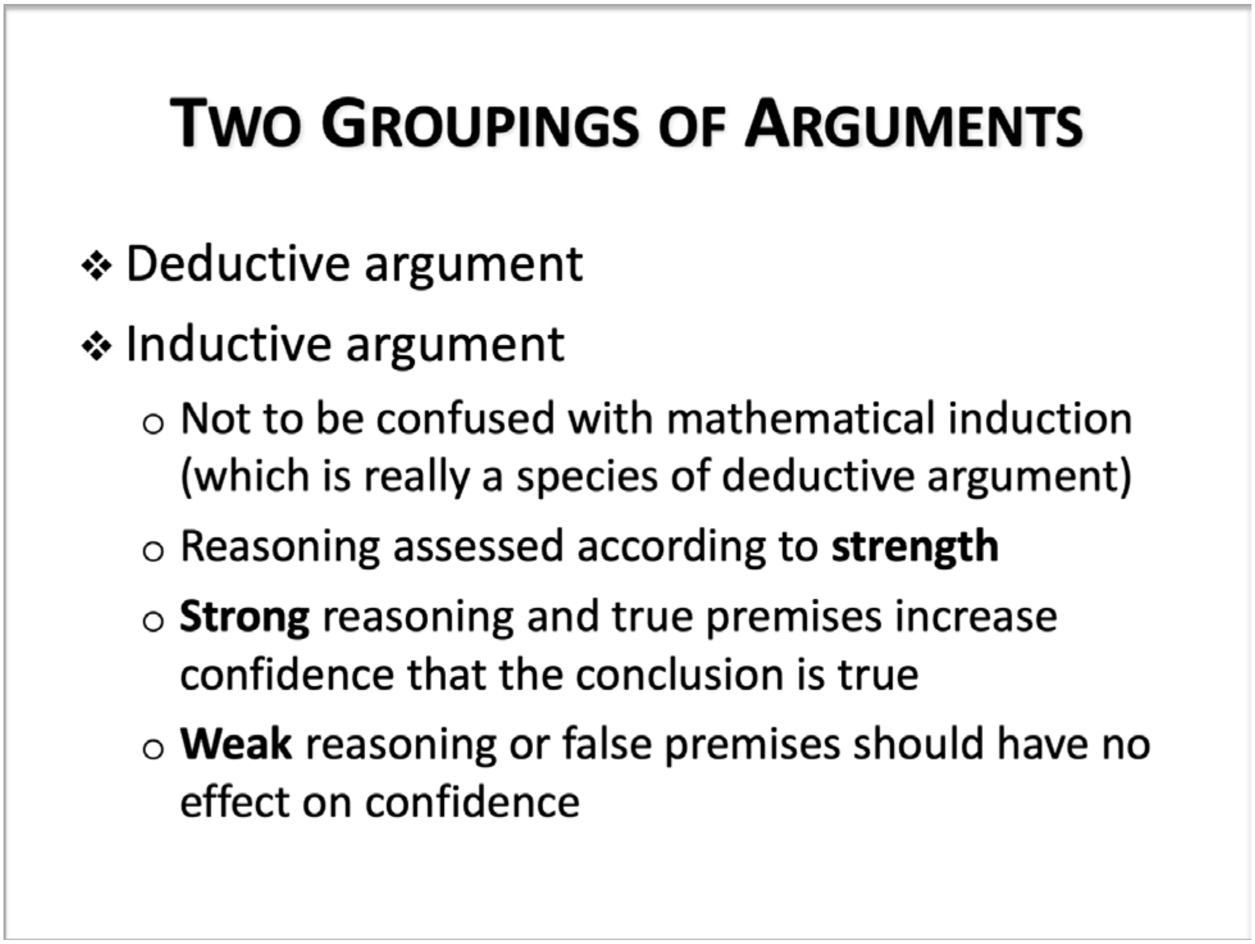
The first of these examples is a sound deductive argument: the form is valid, and (so long as we're talking about mathematical equality) the **premises** are true; hence, the **conclusion** is necessarily also true.

The second example is a valid deductive argument, but it is not sound (and hence the **conclusion** not necessarily true), because one of the **premises** ("*No civil servants are overworked"*) is false.

Please note, and remember always, that just because an unsound argument is given with a particular **conclusion**, does not mean that the **conclusion** is necessarily false.

A sound argument guarantees a true **conclusion**; but an unsound argument by itself tells us *nothing* about the truth of the **conclusion**. It may be false. It may be true (just badly argued for). We do not know.

[Question for participants: Any questions about deductive arguments before I talk a bit about the other main type of argument?]



Two Groupings of Arguments

- Deductive argument
- Inductive argument
  - Not to be confused with mathematical induction (which is really a species of deductive argument)
  - Reasoning assessed according to **strength**
  - **Strong** reasoning and true premises increase confidence that the conclusion is true
  - **Weak** reasoning or false premises should have no effect on confidence

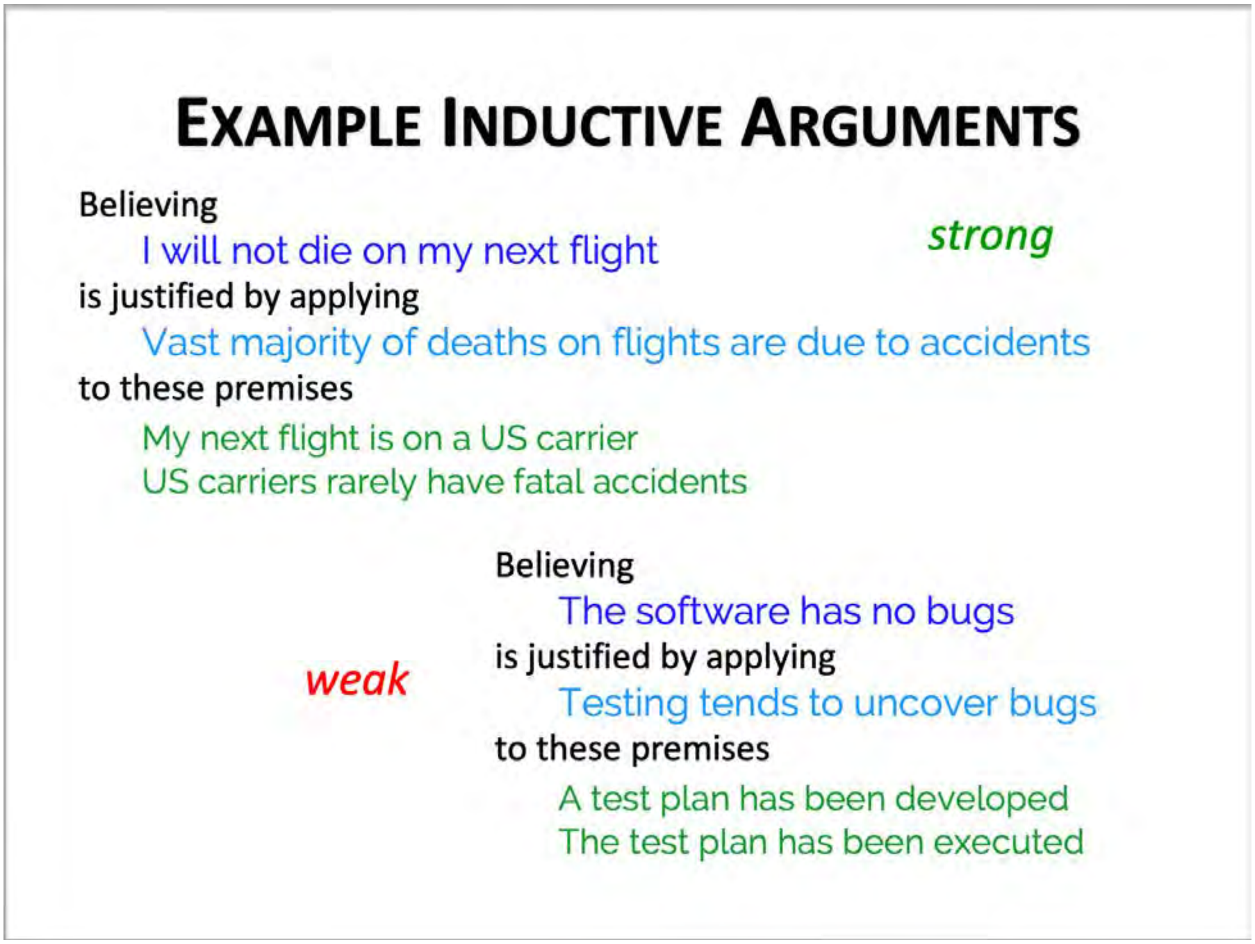
The first thing that everyone needs to remember about inductive arguments is that they are *not* related to mathematical induction, which is really a species of deductive argument.

The terms valid / invalid, sound / unsound don’t really apply to inductive arguments, ‘though you will hear those terms used by some folks.

It is much better to talk in terms of *strength* when it comes to inductive arguments.

An inductive argument with strong **reasoning** and true **premises** should increase confidence that the **conclusion** is really true; whereas weak **reasoning** or false **premises** should (by themselves) have no effect on confidence.

Here are two simple examples of inductive arguments.



EXAMPLE INDUCTIVE ARGUMENTS

Believing

I will not die on my next flight

strong

is justified by applying

Vast majority of deaths on flights are due to accidents

to these premises

My next flight is on a US carrier

US carriers rarely have fatal accidents

Believing

The software has no bugs

weak

is justified by applying

Testing tends to uncover bugs

to these premises

A test plan has been developed

The test plan has been executed

Given *"My next flight is on a US carrier"* and *"US carriers rarely have fatal accidents"*, I believe *"I will not die on my next flight"*, because the *"Vast majority of deaths on flights are due to accidents."*

Of course, as most of you may be thinking, if I was following the Toulmin-based framework more closely, I should include a **qualification** in the **conclusion**, but I've left it out for simplicity, and to enable me to make a point in just a minute.

In the second example, we are arguing that given (**premise** 1) *"A test plan was been developed"* and (**premise** 2) *"The test plan has been executed"*, we should believe (**conclusion**) *"The software has no bugs"* based on the **reasoning** that *"Testing tends to uncover bugs"*.

[Question for participants: What do you think about the strength of these two arguments?]

I'm inclined to say that the first argument is fairly strong, while the second argument is pretty weak.

The first argument could be made even stronger by qualifying the **conclusion**, into something like *"It is very unlikely that I will die on my next flight."*

This illustrates an important point about inductive arguments: the strength of the argument depends on *all* parts of it, not just on (for example) the **reasoning** or the **premises**.

The second example argument as it stands is quite weak, since (among other things) a tendency to uncover bugs does not imply that *all* existing bugs are uncovered. Even if we qualified the **conclusion** a bit, the argument is still going to be rather weak, since (among other things) the **premises** tell us nothing about the quality of the test plan.

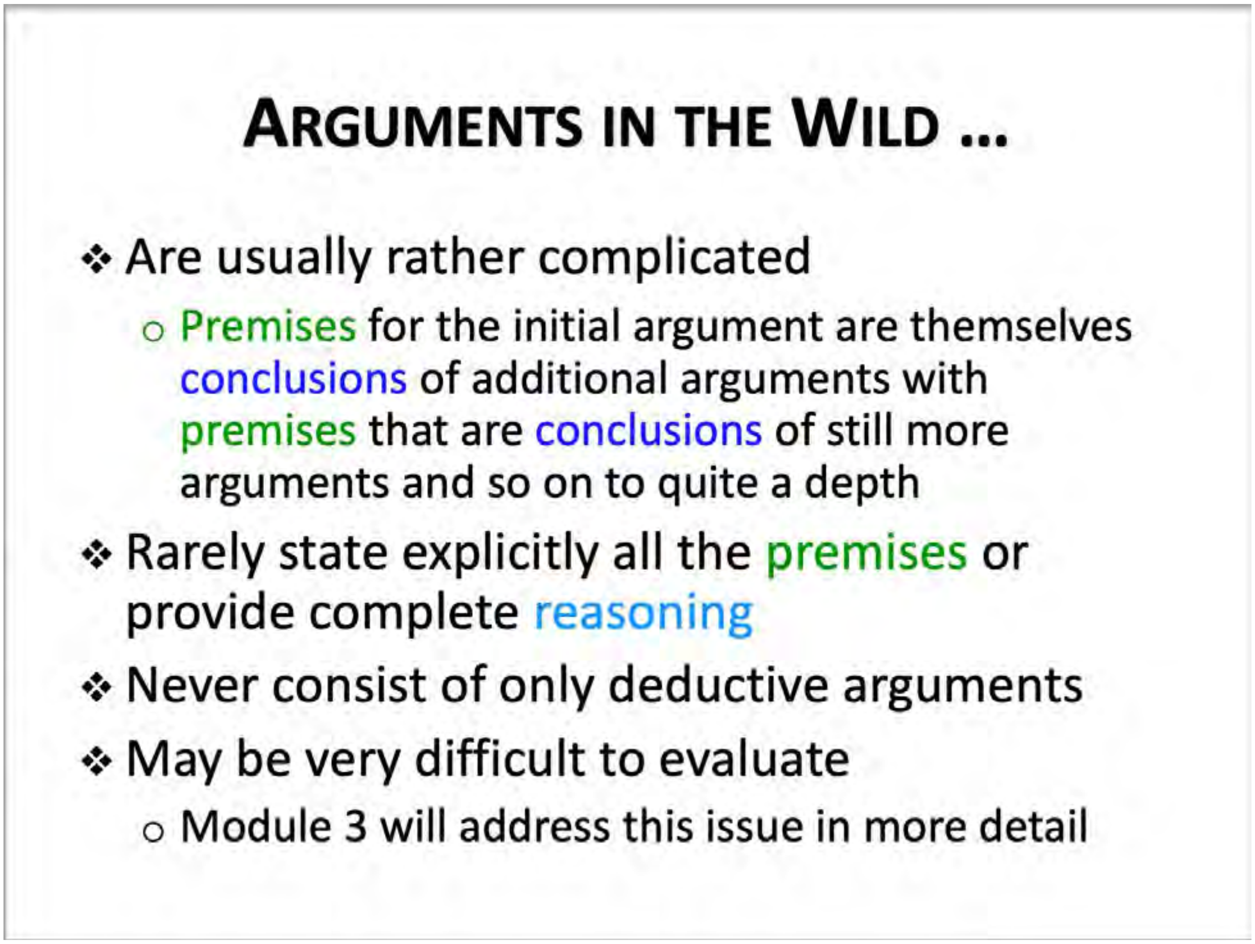
Please always remember that a weak inductive argument does not necessarily mean that the **conclusion** is false. It simply means that *this particular argument* ought not give you confidence that its **conclusion** is true. Perhaps there *is* a strong argument with the same **conclusion** but different other constituent parts.

If, on the other hand, there exists a strong argument with an opposite or contradictory **conclusion**, then that new argument should provide confidence in the falsity of the original **conclusion**.

Similarly, a poor assurance case does not necessarily mean that the system or service is *not* acceptable for its intended use; but it may well indicate some problems.

[Question for participants: Does anyone have questions before we continue?]

So far, we've been talking about arguments mostly in the context of examples contrived to illustrate particular ideas. What about the real world?



ARGUMENTS IN THE WILD ...

- Are usually rather complicated
  - Premises for the initial argument are themselves conclusions of additional arguments with premises that are conclusions of still more arguments and so on to quite a depth
- Rarely state explicitly all the premises or provide complete reasoning
- Never consist of only deductive arguments
- May be very difficult to evaluate
  - Module 3 will address this issue in more detail

Well, arguments in the wild tend to be quite different from the simple examples we've seen in at least four ways.

First, real arguments are usually rather complicated. In particular, **Premises** for the initial argument are themselves **conclusions** of additional arguments with **premises** that are **conclusions** of still more arguments and so on to quite a depth. The argument in any real assurance case for why its top level **conclusion** should be accepted will certainly take such a form. The **premises** for the top level **conclusion** will almost certainly not be obvious truths, but rather statements that will need to be supported by argument themselves.

Eventually the assurance case should stop with sub-arguments with **premises** whose truth can be agreed upon by all relevant parties. A purported 'assurance case' that isn't grounded in such **premises** doesn't deserve to be called an 'assurance case'.

Second, real arguments rarely state explicitly all of the **premises** or provide complete **reasoning**.

Combatting this tendency to leave many things unstated is one of the goals of the assurance case approach.

An assurance case, to be worthy of the name, needs to have sufficiently explicit information (or at least references to information contained elsewhere) to enable evaluators and users of the case to know the intended meaning of all aspects of the argument.

Third, real arguments almost never consist of only deductive arguments.

As I mentioned earlier, this is an area about which there is some controversy. No one disputes that it is true that current assurance cases inevitably contain some inductive arguments. The disputes center around whether there may be advantages to be gained from making deductive as many arguments as possible; or perhaps by using a normalized structure that isolates inductive arguments into specific parts of the overall argument.

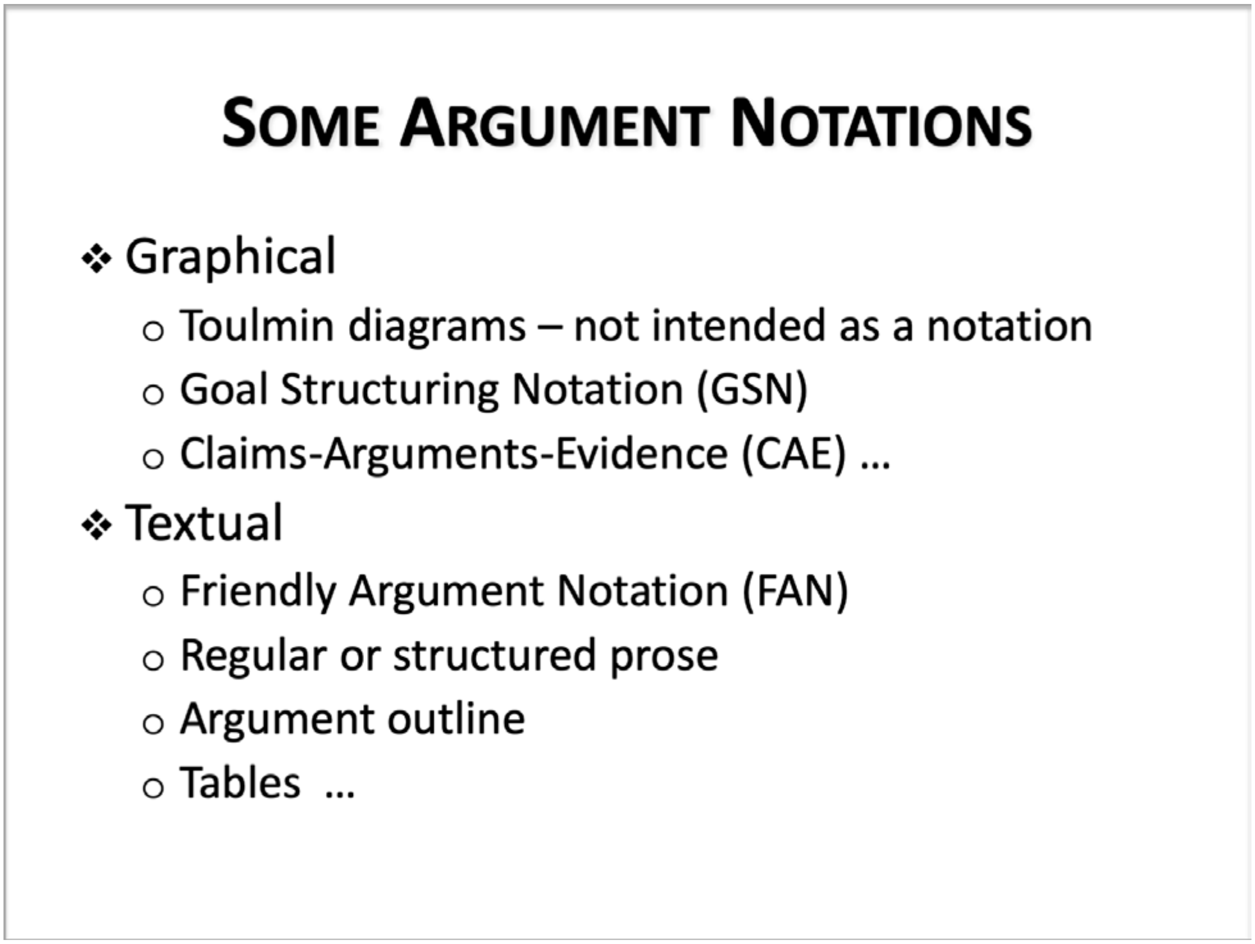
Those who believe that there are advantages to be gained point to (among other things) the simpler evaluation of deductive arguments (much of which could likely be automated).

Those who believe otherwise point to (among other things) the inherent non-formality of many relevant concepts, the likelihood of a huge increase in argument size with a related decrease in human readability, and a skepticism that the sorts of problems solved by formalism are actual problems in real assurance cases.

And finally, as a consequence of these three characteristics, real arguments in the wild may be very difficult to evaluate. Hence the need for a separate module in this series talking about evaluating assurance cases.

We're getting near the end, but this is a good place to stop to ask for questions.

I'm not going to spend a lot of time talking about notations, but I do want to let you know that there are various ones.



SOME ARGUMENT NOTATIONS

- ❖ Graphical
  - o Toulmin diagrams – not intended as a notation
  - o Goal Structuring Notation (GSN)
  - o Claims-Arguments-Evidence (CAE) …
- ❖ Textual
  - o Friendly Argument Notation (FAN)
  - o Regular or structured prose
  - o Argument outline
  - o Tables …

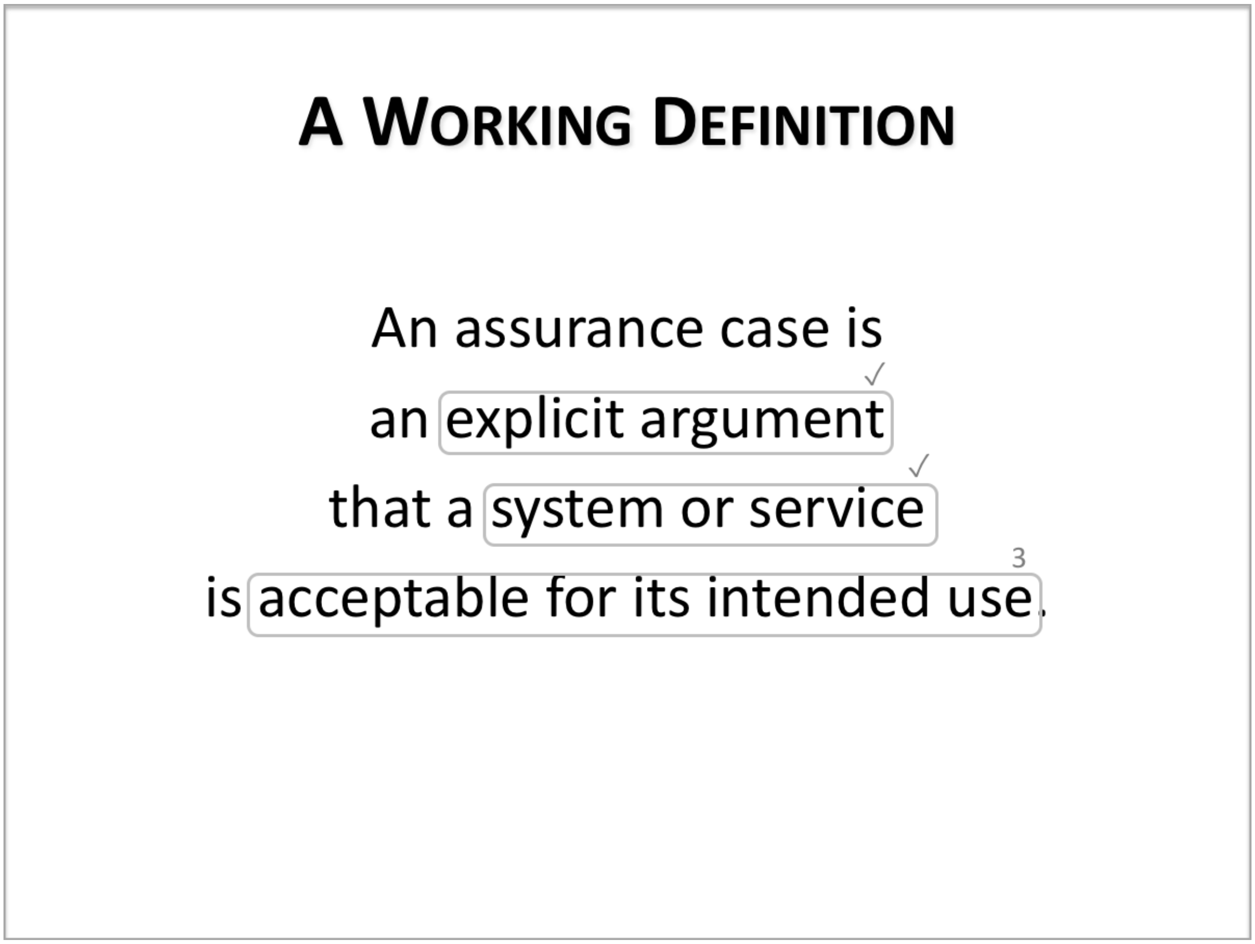
In an earlier version of this model, I used Toulmin diagrams fairly extensively, but I stopped that practice because Toulmin never intended his diagrams to be used that way. Instead I introduced the textual Friendly Argument Notation (FAN).

I've mentioned GSN and CAE, which are the two most common graphical notations used for assurance cases. The website `http://www.goalstructuringnotation.info/` is a good place to visit if you want more information about GSN. To explore CAE further point your favorite browser at `https://www.adelard.com/asce/choosing-asce/cae.html` There are other graphical notations, also, as you may imagine.

There are also or textual ways of representing arguments besides FAN, ranging from unstructured prose, through structured prose, outlines, and tables.

Also, about ten years ago I wrote a conference paper repeating the same example using several notations. It is available at `http://hdl.handle.net/2060/20080042416`. [Holloway, C. M. 2008. "Safety Case Notations: Alternatives for the Non-Graphically Inclined?" *IET 3nd International Conference on System Safety*. 21-23 October 2008, Birmingham, UK.]

That's all that I plan to say about notations, but will be happy to field questions if you have any.



A Working Definition

An assurance case is
an explicit argument ✓
that a system or service ✓
is acceptable for its intended use 3.

Returning to our working definition, we’ve covered two of the three main parts, and while talking about argument we’ve alluded to what needs to be said about the third part: ‘acceptable for its intended use’.

There’s only one more thing that I want to say about it.

The top level **conclusion** is where ‘acceptable for intended use’ is going to be mainly defined; thus a good choice of top level **conclusion** is critical for the success of an assurance case approach. Some critics of the assurance case approach have seemingly missed this point, leveling much of their attacks on badly worded top level **conclusions** as if somehow the approach itself requires people to start with bad ones.

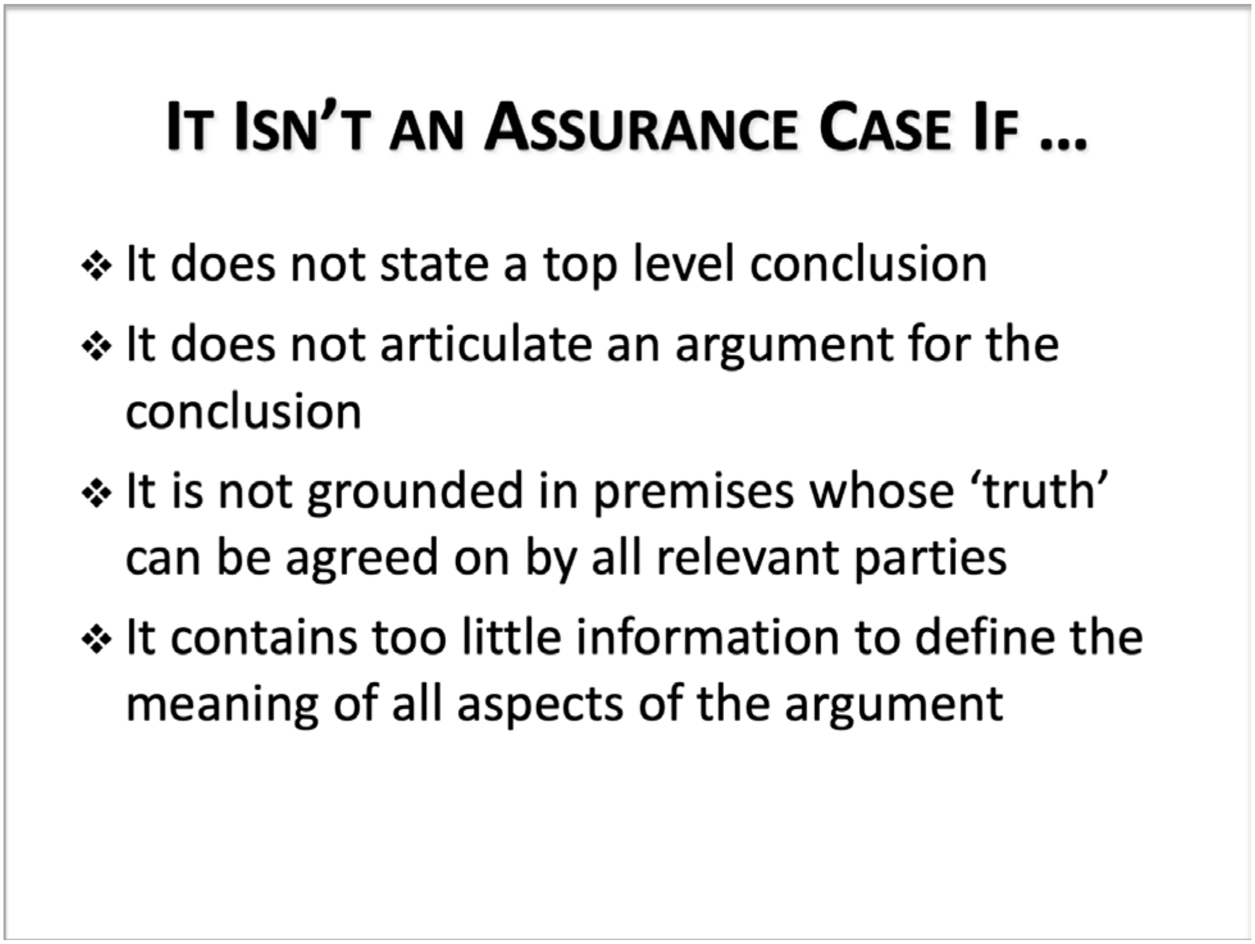
We’ll be talking about how to *recognize* a good top level **conclusion** in some detail in module 3 about assurance case evaluation; and about how to choose a good top level **conclusion** in some detail in module 4 about assurance case creation.

We are almost done with module 1.

Before we quit, however, I want you to think a bit about how you would complete a sentence that begins, “It isn’t an assurance case if ….”

We’ve covered the foundations of assurance cases in sufficient detail that I think you can complete this sentence with several characteristics that distinguish an assurance case from something else.

Here are four things that I think are appropriate completions.



It Isn't an Assurance Case If …

- ❖ It does not state a top level conclusion
- ❖ It does not articulate an argument for the conclusion
- ❖ It is not grounded in premises whose 'truth' can be agreed on by all relevant parties
- ❖ It contains too little information to define the meaning of all aspects of the argument

It isn't an assurance case if … It does not state a top level **conclusion**. If someone claims to have an assurance case but you can't find the **conclusion** the case is supporting, then you're justified in telling them, "This is not an assurance case."

It isn't an assurance case if … It does not articulate an argument for the **conclusion**. If someone claims to have an assurance case but there's no argument, then you're justified in telling them, "This is not an assurance case."

It isn't an assurance case if … It is not grounded in **premises** whose 'truth' can be agreed upon by all relevant parties. If someone claims to have an assurance case but it ends with **premises** whose truth is no more certain than that of higher level conclusions then you're justified in telling them, "This is not an assurance case."

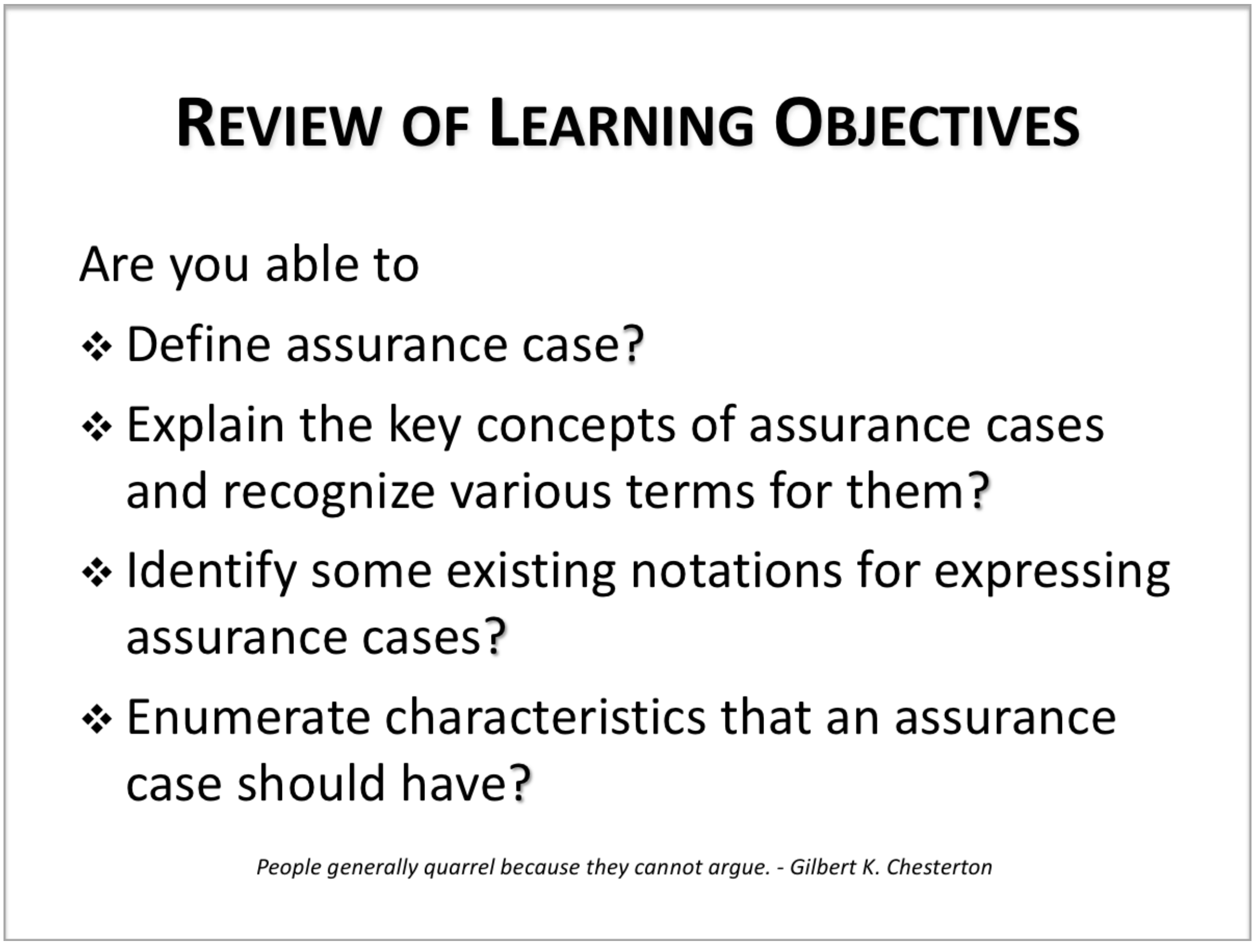
Within the assurance case community, these 'grounded **premises'** tend to be called *evidence*, for a variety of reasons, including analogies to common usage from other fields, and a general belief that there is value in having a specific term for the concept.

So most of my colleagues within the community would probably say something like this: "It isn't an assurance case if … it is not grounded in evidence." If you prefer that formulation from what I've written here, then you'll be in good company.

Finally, it isn't an assurance case if … It contains too little information to define the meaning of all aspects of the argument. If someone claims to have an assurance case but it leaves terms or concepts undefined, then you're justified in telling them, "This is not an assurance case."

At the beginning, I listed four things that I hoped you'd be able to do by the end of this module.

Here are those four things recast in the form of questions.



REVIEW OF LEARNING OBJECTIVES

Are you able to

- ❖ Define assurance case?
- ❖ Explain the key concepts of assurance cases and recognize various terms for them?
- ❖ Identify some existing notations for expressing assurance cases?
- ❖ Enumerate characteristics that an assurance case should have?

People generally quarrel because they cannot argue. - Gilbert K. Chesterton

Think to yourself how you'd answer these questions. If you are not confident in your answers, consider reviewing the materials again.

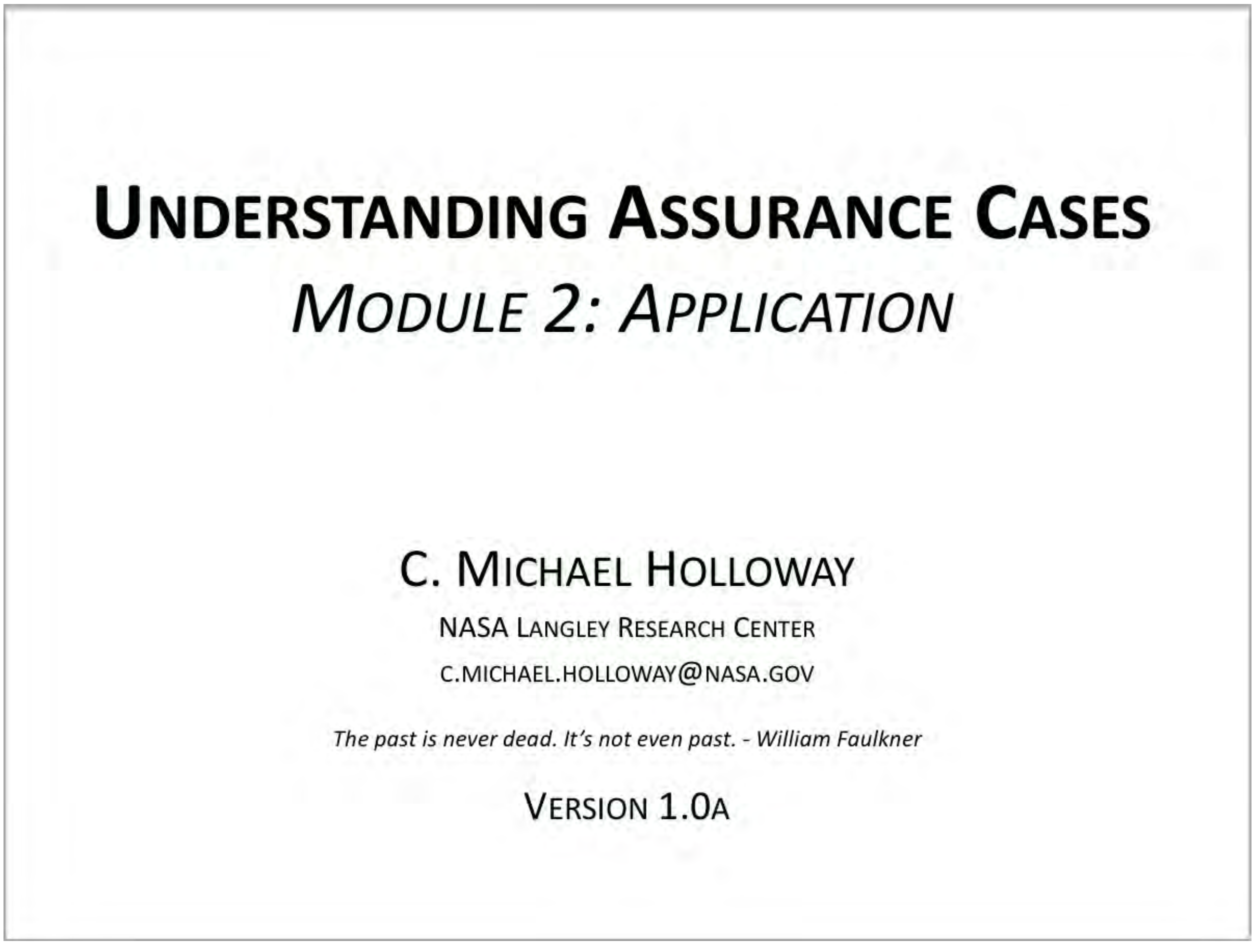
If you have questions or comments about this material, contact its author at `c.michael.holloway@nasa.gov`.

Understanding Assurance Cases: An Educational Presentation in Five Parts

Module 2: Application

C. Michael Holloway
c.michael.holloway@nasa.gov

Senior Research Computer Engineer
Safety-Critical Avionics Systems Branch
NASA Langley Research Center, Hampton, Virginia, U.S.A.



UNDERSTANDING ASSURANCE CASES

MODULE 2: APPLICATION

C. MICHAEL HOLLOWAY

NASA LANGLEY RESEARCH CENTER

C.MICHAEL.HOLLOWAY@NASA.GOV

The past is never dead. It's not even past. - William Faulkner

VERSION 1.0A

This material was created in 2015-16, as part of the Explicate '78 project. The project was supported in substantial part by the Assurance Case Applicability to Digital Systems task under the reimbursable interagency agreement with the Federal Aviation Administration for Design, Verification, and Validation of Advanced Digital Airborne Systems Technology (IAI-1073 Annex 2 for NASA; DTFACT-10-X0008, Modification 0004 for the FAA). The original presentations were delivered to a selected group of FAA civil servants and NASA Langley personnel. The audio was recorded and partial transcripts (containing only the words spoken by the presenter, Mr. Holloway) produced. The intent from the beginning was to collect the material into a form that could be made available publicly. The text adheres closely to the original transcript, with the exception of an occasional insertion of new information that arose since the original presentation dates. The full collection consists of six documents (including this one), which are available electronically through `https://shemesh.larc.nasa.gov/arg/uac.html`.

Hello everybody.

Welcome to the second module in our educational series about Understanding Assurance Cases. In this module, we will examine the ***Application*** of the assurance case concept.

We'll be talking about the past and the present, hence a famous quote from William Faulkner encapsulates our theme:

"The past is never dead. It's not even past."

[Faulkner, William. 1951. *Requiem for a Nun*. act i, scene iii. New York: Random House.]

As with Module 1, there will be several times when I'll stop to give you a chance to ask questions; but feel free to interrupt me at *any* point if you have a burning question. I'll either try to answer it right away, or defer it to a better time a bit later on.

Before going any further, I will repeat verbatim some preliminary remarks I made at the beginning of Module 1.

Within the assurance case community, intramural debates abound about a variety of topics we will discuss. Except in rare instances the existence of these debates is intentionally ignored or mentioned only briefly in this material. Here's why.

Disagreements exist about terms, definitions, notations, philosophy, procedures, tools, and just about everything else.

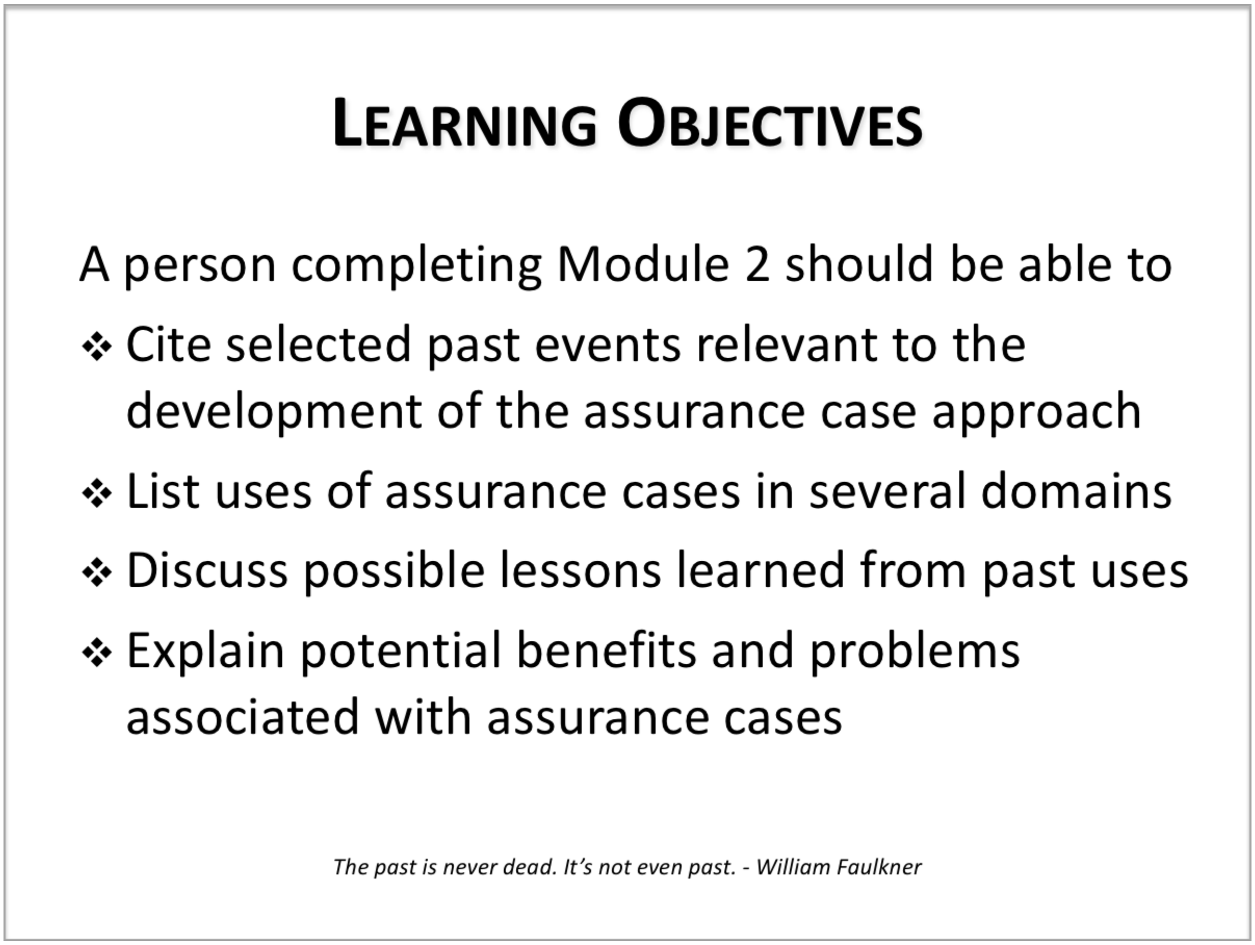
The depth of the disagreements ranges all the way from *shallow* differences in preferences (which term best denotes a particular concept, for example), to rather *deep* philosophical differences (the feasibility and desirability of formalizing assurance arguments, for example).

Spending *too much* time on these disagreements would likely make this material deeply confusing; but spending *too little* time on them might hinder your understanding of some materials you may come across.

In trying to strike a balance, what I've chosen to do is *not* highlight the areas of disagreement on the slides (except occasionally where it is seems essential), but to mention the disagreements where appropriate in my words accompanying the slides.

One other quick note before we proceed: All images you see were either created by me (Michael Holloway) or are in the public domain via CC0 1.0 Universal. For images that do not fall into either category, you will see only links, not the actual image that was used in the original presentation.

Here are the four learning objectives for Module 2.



Learning Objectives

A person completing Module 2 should be able to

- ❖ Cite selected past events relevant to the development of the assurance case approach
- ❖ List uses of assurance cases in several domains
- ❖ Discuss possible lessons learned from past uses
- ❖ Explain potential benefits and problems associated with assurance cases

The past is never dead. It's not even past. - William Faulkner

By the time we're finished today, I hope that you'll be able to do at least these four things:

One, cite selected past events relevant to the development of the assurance case approach.

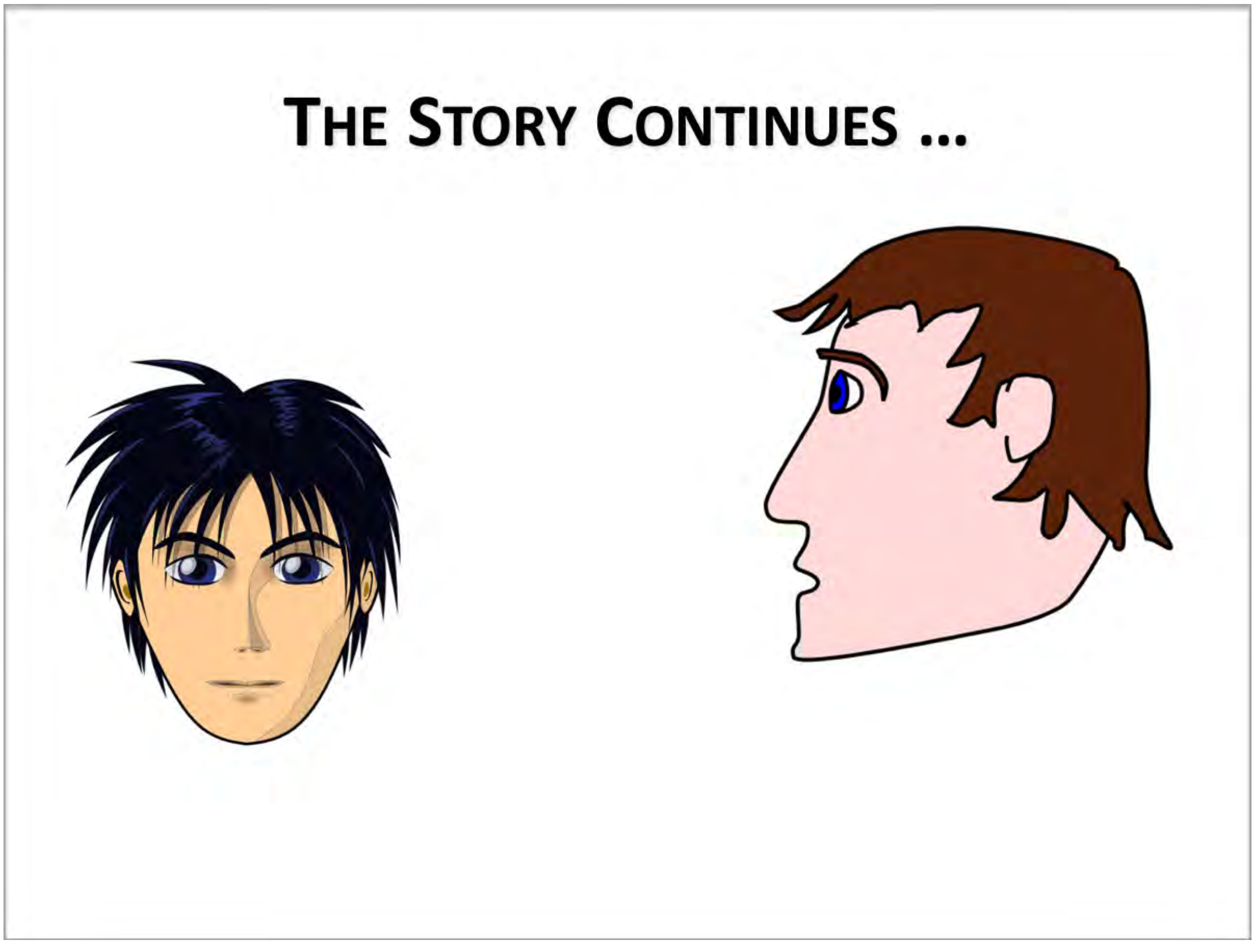
Two, list uses of assurance cases in several domains.

Three: discuss possible lessons learned from past uses.

And four: explain potential benefits and potential problems associated with assurance cases.

[Question to participants: Any questions about these learning objectives?]

As I hope you remember, in Module 1 we introduced Jon (the young fellow on the left) Jon's dad (named Mike, the not-nearly-as young fellow on the right), and Tim (a fellow we've not seen, but with whom Jon wants to ride to a game).



THE STORY CONTINUES ...

You also, I hope, recall that at the end of our story in Module 1 Jon's dad had said that he wanted to see an assurance case for why he should believe that Tim would get Jon to and from the game in one piece.

To this, Jon had replied, "I'll ask Tim if he has one."

A few hours later, the story continues as Jon and his dad get together again.

Jon says, "I asked Tim about an assurance case."

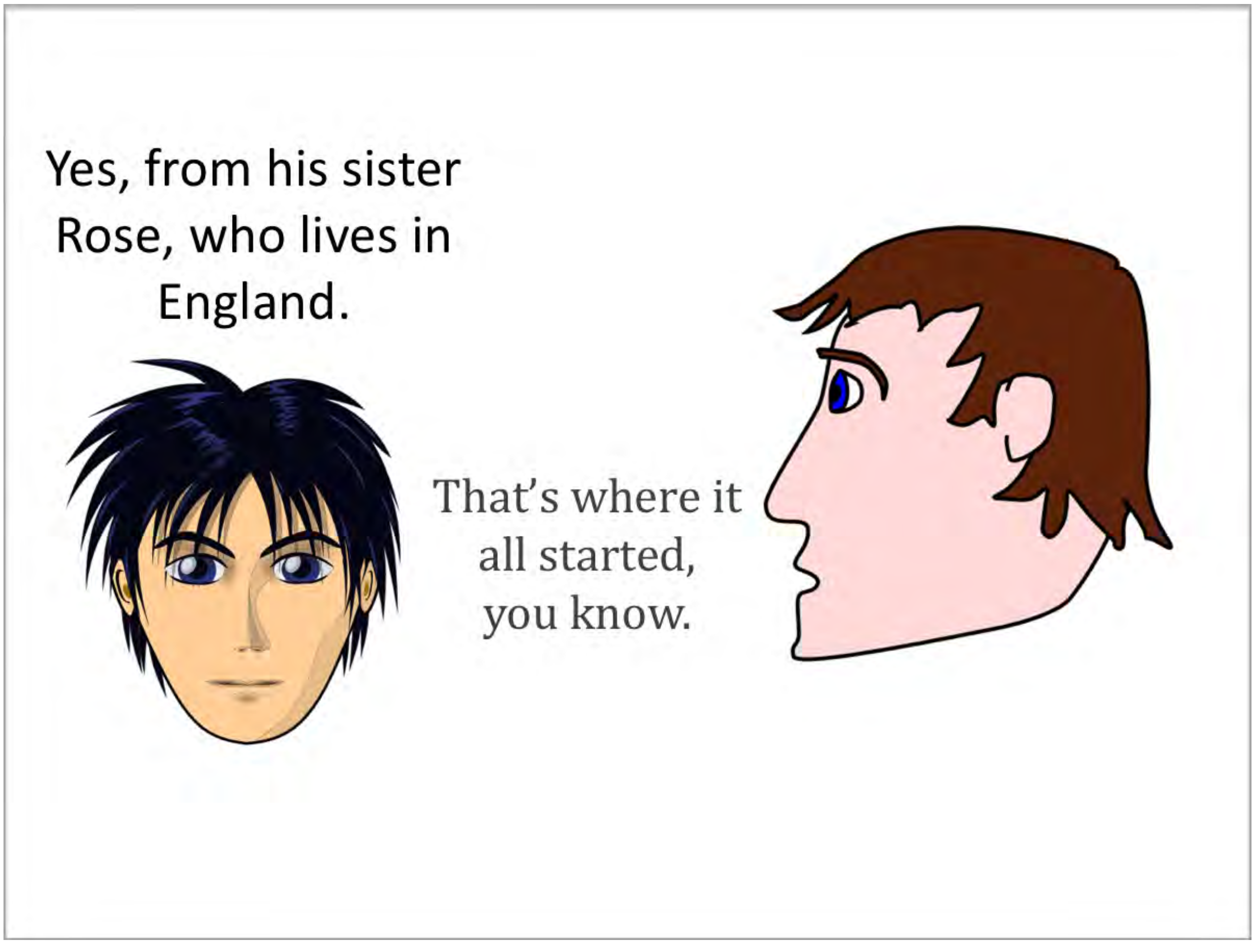
Jon's dad asks expectantly, "What did he have to say for himself?"

"He was a tad bit confused," replies Jon.

Mike, himself a tad bit confused, asks, "Confused? What was he confused about?"

Jon answers: "Whether an assurance case is the same thing as a safety case."

"Then he's heard about safety cases?" Jon's dad replies, once again expectantly.


Yes, from his sister Rose, who lives in England.
That's where it all started, you know.

"Yes, from his sister Rose who lives in England."

"That's where it all started you know."

"Safety cases came first. Assurance cases are more general. All I really need from Tim is just a safety case."

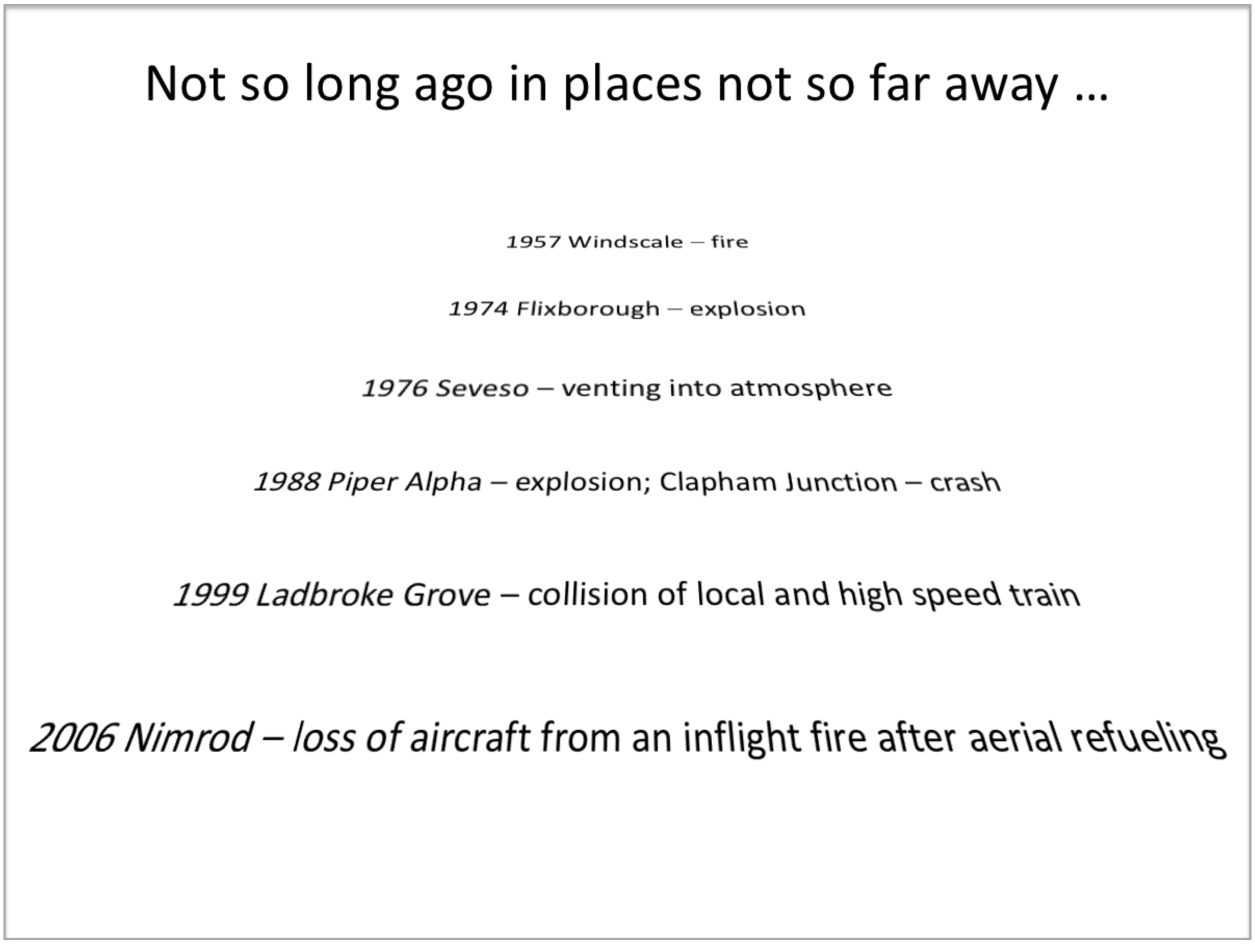
Jon breathes a sigh of relief, and exclaims, "That'll make Tim happy. Thanks dad!"

After a short pause, Jon adds, with a slight smirk on his face, "Oh, Dad, I almost forgot … Tim also wondered … if you want a *brief case*, or a long one."

Mike starts to reply, then the word play registers in his mind, and he simply smiles.

As Mike said, safety cases did come first. So, let's talk a bit about the origins of safety cases. Like many origin stories, the details are a bit murky, and not everyone agrees about when and how things started.

Everyone *does agree*, that the beginnings were not so long ago in places not so far away from us. Back then and over there …



Not so long ago in places not so far away …

1957 Windscale – fire

1974 Flixborough – explosion

1976 Seveso – venting into atmosphere

1988 Piper Alpha – explosion; Clapham Junction – crash

1999 Ladbroke Grove – collision of local and high speed train

2006 Nimrod – loss of aircraft from an inflight fire after aerial refueling

… the accidents you see here were catalysts for changes in the way people thought about and regulated various dangerous activities and systems.

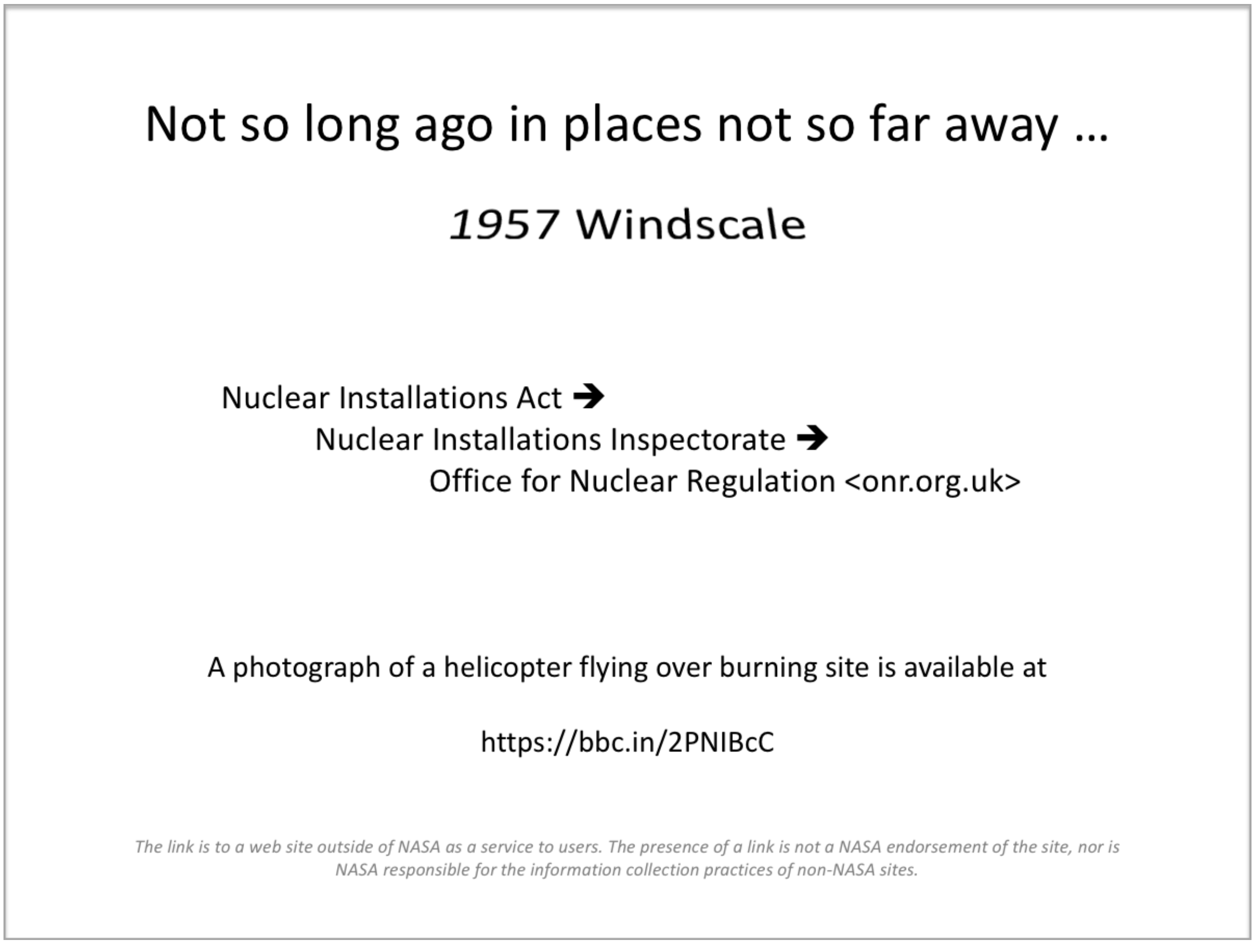
Windscale, Flixborough, Seveso, Piper Alpha, Clapham Junction, Ladbroke Grove, and, Nimrod have all played a part in safety case history and lore.

We could easily spend a whole hour or more talking about any one of these alone, so what I'll present are only incomplete overviews; and as with any overview, I may leave out some things that other folks would include, and include some things that other people would leave out.

Let us begin in northwestern England quite close to the coast, three and one quarter years before I was born

In October 1957, a fire at the Windscale nuclear reactor facility and plutonium-production plant resulted in a major release of radioactive materials. The fire started when a routine heating of the graphite control blocks in the number 1 reactor ran out of control, rupturing adjacent uranium cartridges. The uranium oxidized, causing a fire that burned for 16 hours before it was extinguished, and releasing radioactive Iodine-131 into the atmosphere, and also melting about 10 tons of the reactor core.

The UK government banned, for several weeks, the sale of milk from about a 200 square mile area around the site but generally told the public few details about the accident at the time. Windscale remains today the most serious nuclear power accident in the UK.



Not so long ago in places not so far away …

1957 Windscale

Nuclear Installations Act ➔
Nuclear Installations Inspectorate ➔
Office for Nuclear Regulation <onr.org.uk>

A photograph of a helicopter flying over burning site is available at

https://bbc.in/2PNIBcC

The link is to a web site outside of NASA as a service to users. The presence of a link is not a NASA endorsement of the site, nor is NASA responsible for the information collection practices of non-NASA sites.

Among the actions taken in the wake of the accident was the adoption in 1959 of the Nuclear Installations Act

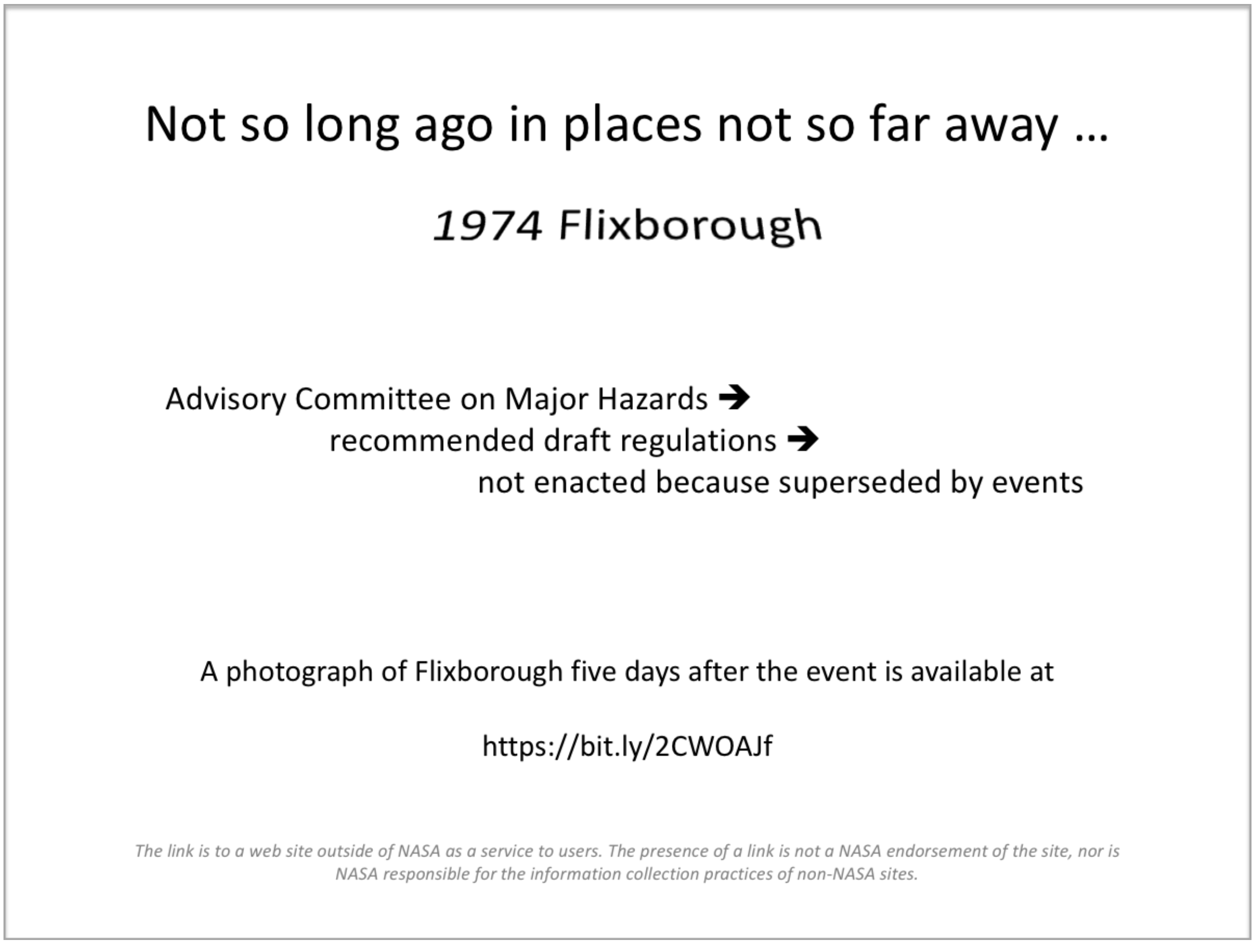
This act established the Nuclear Installations Inspectorate, which in turn required prospective reactor installations to submit a set of reports justifying the safety of the design, construction, and operation of the plant.

Although the term 'safety case' was not used in these early days, many, in retrospect, consider the certification process that was established in the wake of Windscale as the true beginning of the safety case approach.

Over the years, the UK commercial nuclear power regulations became increasingly more clearly safety-case based. Regulation is now the responsibility of the Office for Nuclear Regulation, which is an agency of the Health and Safety Executive. Their web site, O-N-R dot O-R-G dot U-K, is worth visiting, if for no other reason than to see in print an usually well-written mission statement: 'The Office for Nuclear Regulation's mission is to provide efficient and effective regulation of the nuclear industry, holding it to account on behalf of the public.'

They also have several short documents worth reading; we'll come back to at least one of those in later Modules.

Let's now move forward in time nearly 17 years, and in geography about 200 miles south east across England.



Not so long ago in places not so far away …

1974 Flixborough

Advisory Committee on Major Hazards ➔
recommended draft regulations ➔
not enacted because superseded by events

A photograph of Flixborough five days after the event is available at

https://bit.ly/2CWOAJf

The link is to a web site outside of NASA as a service to users. The presence of a link is not a NASA endorsement of the site, nor is NASA responsible for the information collection practices of non-NASA sites.

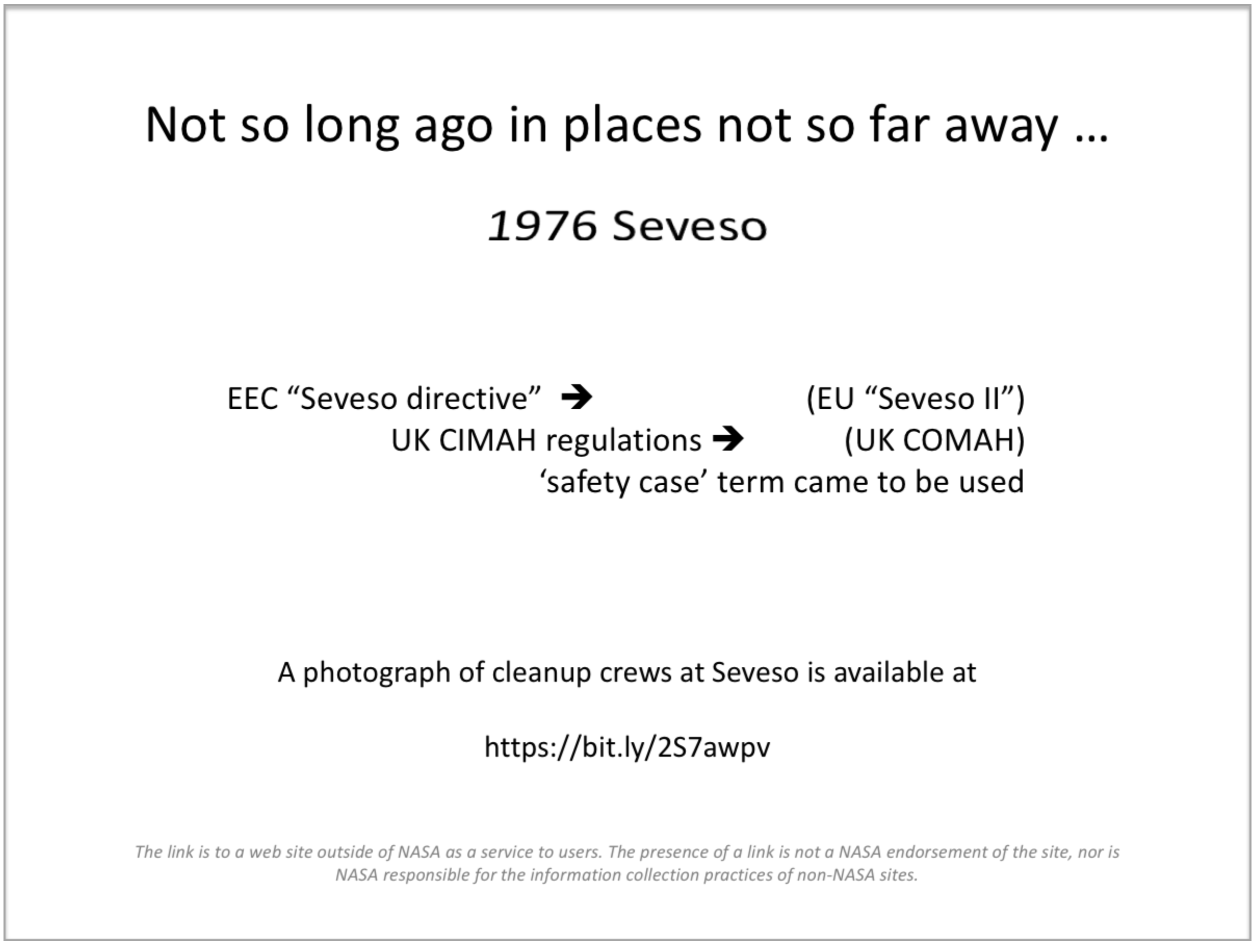
To provide the summary of the Flixborough disaster, I'm going to quote directly from the report produced by the Court of Inquiry, because improving on its words, or at least its first sentence, is impossible.

"At about 4.53 pm on Saturday 1st June 1974 the Flixborough Works of Nypro (UK) Limited were virtually demolished by an explosion of warlike dimensions. Of those working on the site at the time, 28 were killed and 36 others suffered injuries. If the explosion had occurred on an ordinary working day, many more people would have been on the site, and the number of casualties would have been much greater. Outside the Works injuries and damage were widespread but no-one was killed. Fifty-three people were recorded as casualties by the … police; hundreds more suffered relatively minor injuries which were not recorded. … Property damage extended over a wide area."

[Department of Employment. 1975. *The Flixborough disaster: Report of the Court of Inquiry*. London: Her Majesty's Stationery Office.]

Whether the explosion's initiating event was a failure in a 20-inch bypass or in an 8-inch pipe was the subject of much controversy during the court of inquiry and ever since. The inquiry came out in favor of the 20-inch hypothesis but the initiating event is not important for our purposes. What's important is that in response to the accident an Advisory Committee on Major Hazards was created within the Health and Safety Executive.

The Committee recommended that regulations be established to "ensure identification, assessment and management of potential hazards in chemical installations." These recommendations resulted in draft regulations, which were not enacted because other events outside of the UK happened to change the regulatory landscape.



Not so long ago in places not so far away …

1976 Seveso

EEC “Seveso directive” ➔ (EU “Seveso II”)
UK CIMAH regulations ➔ (UK COMAH)
‘safety case’ term came to be used

A photograph of cleanup crews at Seveso is available at

https://bit.ly/2S7awpv

The link is to a web site outside of NASA as a service to users. The presence of a link is not a NASA endorsement of the site, nor is NASA responsible for the information collection practices of non-NASA sites.

These other events began on July 10, 1976, not in the UK, but rather in Seveso, Italy, a few miles north of Milan, when a rupture disc blew on a chemical reactor operated by the Icmesa chemical company. This occurred when a batch process was interrupted before the final step was completed (to conform to Italian law concerning hours that a plant could be operating). The interruption resulted in a spike in steam temperature, which was unseen by the operators because the vessel had no active temperature measurement.

The steam overheated the upper part of the reactor chamber, and with agitation turned off as part of the plant shutdown process, an exothermic runaway reaction began. This reaction produced tetra-chloro-di-ben-zo-p-di-oxin (known as TCDD, and sometimes incorrectly called simply dioxin), which is a highly toxic chemical that the plant *did not* produce during normal operations. No deaths were directly attributed to the TCDD release, but many people got sick, many animals died, and a substantial area had to be evacuated and stripped of soil.

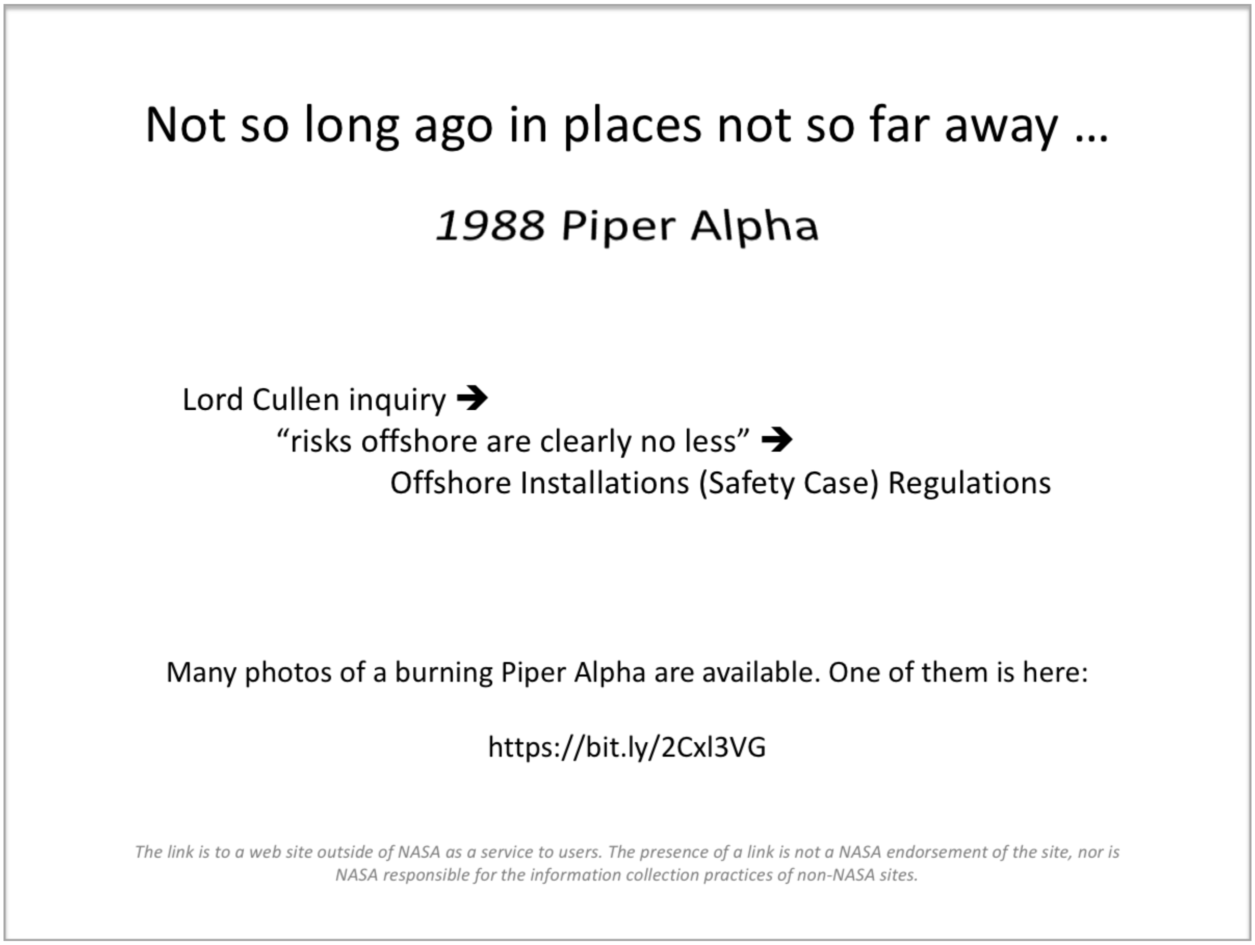
The accident led to the European Economic Community adopting in 1982 what become known as the Seveso directive. The adoption of this directive is cited by some folks as the true origin of safety cases. Among other things the directive required member states to make manufacturers responsible for “tak[ing] all the measures necessary” to prevent “major accidents” and to “prove” that they have done so.

The United Kingdom implemented the directive through the Control of Industrial Major Accident Hazards (CIMAH) regulations in 1984. CIMAH required manufacturers to “provide evidence including documents to show that” they have … “identified the major accident hazards; and … taken adequate steps to … prevent … major accidents and to limit their consequences to persons and the environment, and … provide persons working on the site with the information, training and equipment necessary to ensure their safety.”

Some other people consider the CIMAH regulations to be the true origin of safety cases, perhaps because the term itself came to be used in relation to documents produced by manufacturers to comply with the regulations.

The original Seveso directive has since been superseded by a European Union law generally known as Seveso II; in the UK the CIMAH regulations were replaced by the Control of Major Accident Hazard Regulations (COMAH). The safety case idea is still strong with them.

CIMAH applied to installations on-shore that posed major accident hazards. It did *not* apply to off-shore installations.



Not so long ago in places not so far away …

1988 Piper Alpha

Lord Cullen inquiry ➔
“risks offshore are clearly no less” ➔
Offshore Installations (Safety Case) Regulations

Many photos of a burning Piper Alpha are available. One of them is here:

https://bit.ly/2Cxl3VG

The link is to a web site outside of NASA as a service to users. The presence of a link is not a NASA endorsement of the site, nor is NASA responsible for the information collection practices of non-NASA sites.

Piper Alpha was an off-shore installation (an oil platform to be specific) located in the North Sea about one hundred ten miles from Aberdeen, Scotland. On July 6, 1988, two hundred and twenty-six people were aboard the platform when it experienced a series of catastrophic explosions and fires. One hundred sixty-seven people were killed (including two not from the platform who died in a rescue attempt), and the platform was totally destroyed. Because the platform was destroyed, little physical evidence was available for investigators, so the precise combination of events that led to the disaster is not known for sure.

The public inquiry led by Lord Cullen concluded that most likely the initial explosion occurred when a pump was restarted after maintenance by operators who were unaware that a relief value in the pump discharge had also been removed for maintenance, and a blank loosely installed in its place. This blank leaked, producing a flammable hydrocarbon cloud, which found an ignition source. From that point things spiraled out of control in a variety of ways we won't take time to discuss now.

In addition to determining the likely direct causes of the accident, and discussing specific related recommendations, the inquiry by Lord Cullen also considered more general issues.

One resulting recommendation was that off-shore operations should be required to have a safety case just like on-shore operations. He wrote: "A Safety Case should be required for existing installations. This is the case onshore. The risks offshore are clearly no less. It is not acceptable that installations should be operated without a thorough assessment of what those risks are."

He further wrote that the Safety Case should be primarily "the means by which an operator demonstrated to itself the safety of its activities." Lord Cullen emphasized that the Safety Case should not be a static document, but part of a continuing dialog about safety, including between the operator and the regulatory body, whose role would largely be one of auditor.

As a direct result of Lord Cullen's recommendations, the Offshore Installations (Safety Case) Regulations were introduced in the UK in 1992, making the processing industries onshore and offshore subject to producing and maintaining safety cases.

The story turns now from processing to transportation, particularly rail transportation in the UK.

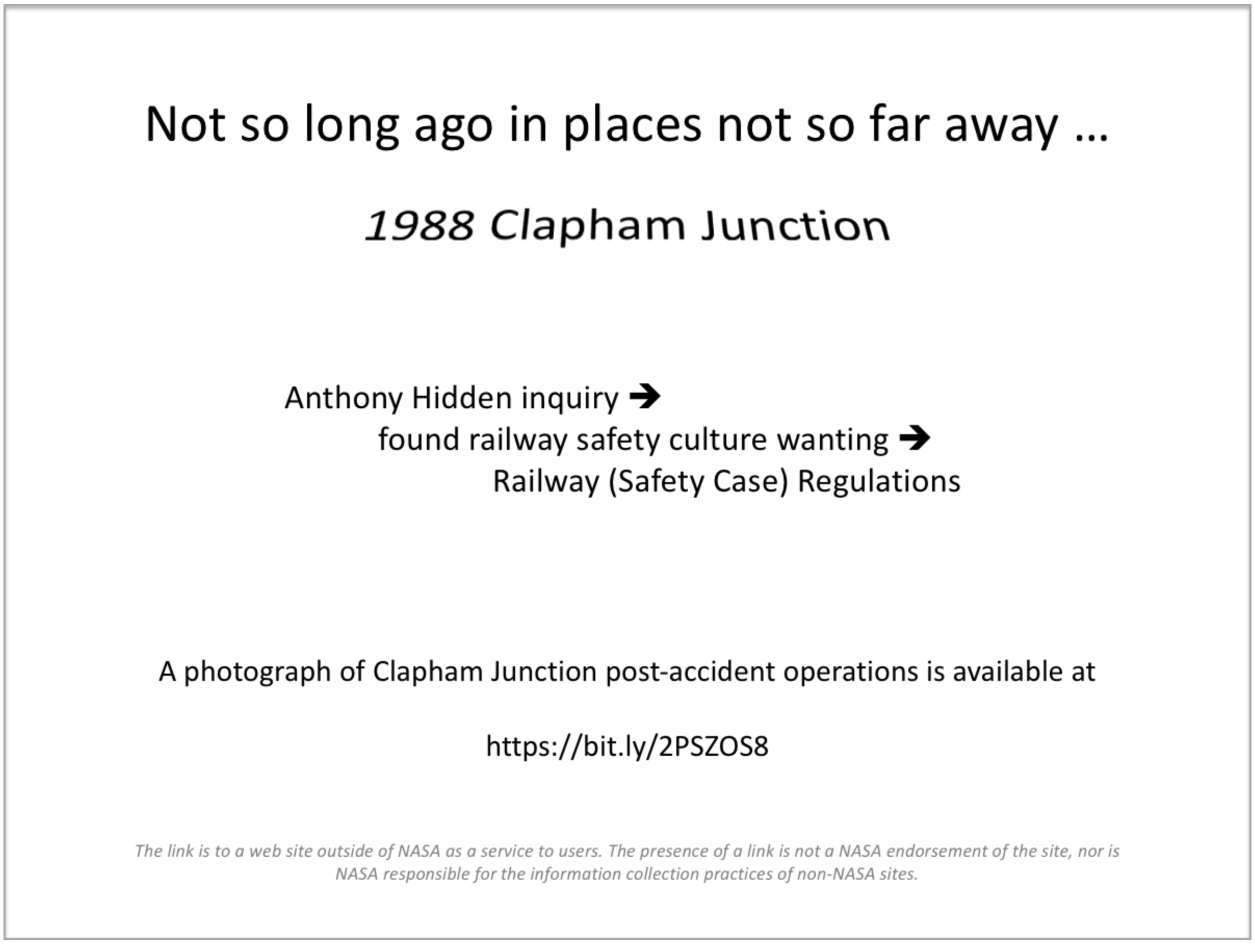
[Question to participants: Before I continue the story, does anyone have any questions?]

There are two pertinent rail accidents for us to discuss.

The first happened in 1988, the same year as the Piper Alpha disaster. I'll tell you about it by quoting some excerpts from report produced by the inquiry into the accident led by Anthony Hidden, because improving on its excellent wording is unlikely.

[Hidden, Anthony. 1989. *Investigation of the Clapham Junction Railway Accident*. London: Her Majesty's Stationery Office.]

"On the railway lines between Waterloo and Wimbledon four tracks run through a cutting a mile or so to the country side of Clapham Junction railway station. … Just after 8 a.m. on Monday, 12 December 1988 three specific trains were running towards that cutting on their normal timetables. Two passenger trains were heading into Waterloo …. One, the 07:18 from Basingstoke, the other, running behind it from the South Coast, the 06:14 'Poole' train. The third train, the 08:03 Waterloo to Haslemere, was running without passengers … on [an] adjoining line."



Not so long ago in places not so far away …

1988 Clapham Junction

Anthony Hidden inquiry ➔
found railway safety culture wanting ➔
Railway (Safety Case) Regulations

A photograph of Clapham Junction post-accident operations is available at

https://bit.ly/2PSZOS8

The link is to a web site outside of NASA as a service to users. The presence of a link is not a NASA endorsement of the site, nor is NASA responsible for the information collection practices of non-NASA sites.

“At about 8:10 … the driver of the ‘Poole’ train, having come into the cutting on his way into Waterloo … and having passed signals in his favour at all stages, cleared the visual obstruction of the steep bank on the left-hand curve. At that moment he must have come upon what was, in signaling and therefore in driving terms, unthinkable and impossible: immediately ahead of him was the Basingstoke train on the same line, stationary, and within a distance in which the ‘Poole’ train could not possibly be stopped.”

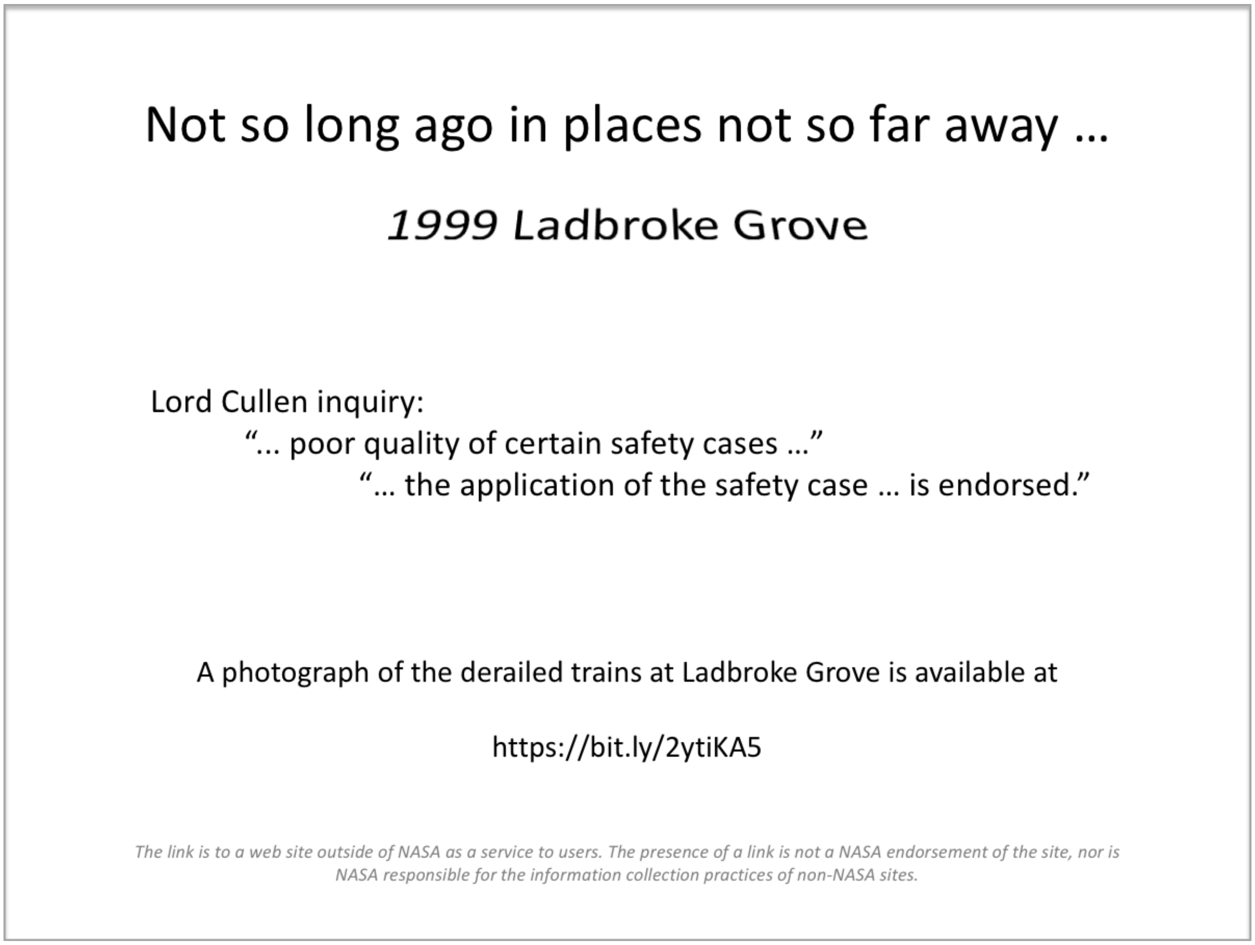
“Despite full emergency braking of the ‘Poole’ train, its leading coach collided head-on with the rear of the Basingstoke train. The collision forced it out to its off-side where it struck the third ‘empty’ train going in the opposite direction. … An appalling accident had happened.”

Thirty-five people died as a result of the accident (Thirty-three on scene, and two a bit later from their injuries. All of them had been carried in the first two coaches of the ‘Poole’ train.

The physical cause of the accident was fairly straightforward to uncover: a signal failure, which had been caused by a maintenance-induced wiring fault.

The inquiry, however, did not stop at finding the physical cause, it also discussed the whole railway safety culture at the time, and found it wanting. The inquiry’s report was one of the catalysts for a wider public consideration of railway safety, which ultimately led to the introduction in 1994 of Railway (Safety Case) Regulations, which required railway infrastructure controllers and all train and station operators to prepare safety cases that demonstrated sufficient thought about and management of all credible hazards.

The Clapham Junction accident led to the requirement for safety cases in the railways; another accident more than a decade later led to deeper consideration of the content and disposition of such cases.



Not so long ago in places not so far away …

1999 Ladbroke Grove

Lord Cullen inquiry:

"... poor quality of certain safety cases …"

"… the application of the safety case … is endorsed."

A photograph of the derailed trains at Ladbroke Grove is available at

https://bit.ly/2ytiKA5

The link is to a web site outside of NASA as a service to users. The presence of a link is not a NASA endorsement of the site, nor is NASA responsible for the information collection practices of non-NASA sites.

"On 5 October 1999 at Ladbroke Grove junction, about two miles west of Paddington Station, London, there was a head on crash at high speed between trains operated by Thames Trains and First Great Western (FGW). This caused the death of [] 31 persons … include[ing] both train drivers, and inflicted injuries, some of them critical, on over 400 other persons."

Lord Cullen, of Piper Alpha fame, conducted a public inquiry into the accident and eventually published a 2-volume report. The first volume of the report dealt mainly with specifics of the accident. The brief summary I gave a moment ago comes directly from words in volume 1.

[The Rt Hon Lord Cullen, PC. 2000. *The Ladbroke Grove Rail Inquiry: Part 1 Report*. Norwich: HSE Books.]

The inquiry discovered that the Thames Train passed a Red danger signal travelling at about 41 mph, leading it to the Main line, on which the First Great Western high speed train was approaching on green signals. Both train drivers applied their brakes, but too late to have any significant effect. The collision occurred at a combined speed of about 130 mph. The inquiry considered it more probable than not that the poor sighting of the signal passed at danger, coupled with bright sunlight at a low angle, were factors that led the train driver to think that he had a proceed aspect.

The second volume produced by Lord Cullen's inquiry "was concerned in regard to the railways, with the management of safety and the regulatory regime." Lord Cullen noted "The general object of a safety case is to ensure that an operator has the will, capabilities, organisation, system and resources to operate safely."

[The Rt Hon Lord Cullen, PC. 2001. *The Ladbroke Grove Rail Inquiry: Part 2 Report*. Norwich: HSE Books.]

He further stated: "The application of the safety case to Great Britain's railways is endorsed. … there is a need for the framework provided by the Safety Case Regulations, within which the duty holder demonstrates, and by reference to which it operates, its arrangements and procedures for the management of safety in a consistent and effective manner."

Lord Cullen also noted "The Inquiry heard evidence from a number of witnesses about the poor quality of certain safety cases, especially the earliest which had been produced."

In discussing poor quality safety cases, he stated "While it is clear that the safety case can become overbureaucratic, it has the potential to be a valuable tool, by, for example, bringing about a systematic approach to safety and providing a record of management's commitments to safety. The evidence showed that it can be a 'living document', part of the direct management of safety. The discipline of producing a safety case has an important value in itself. … The evidence [also] demonstrated the significance of ensuring employees' understanding and knowledge of its substance."

Thus far, we've talked about nuclear power, chemical processing of various sorts, and railways.

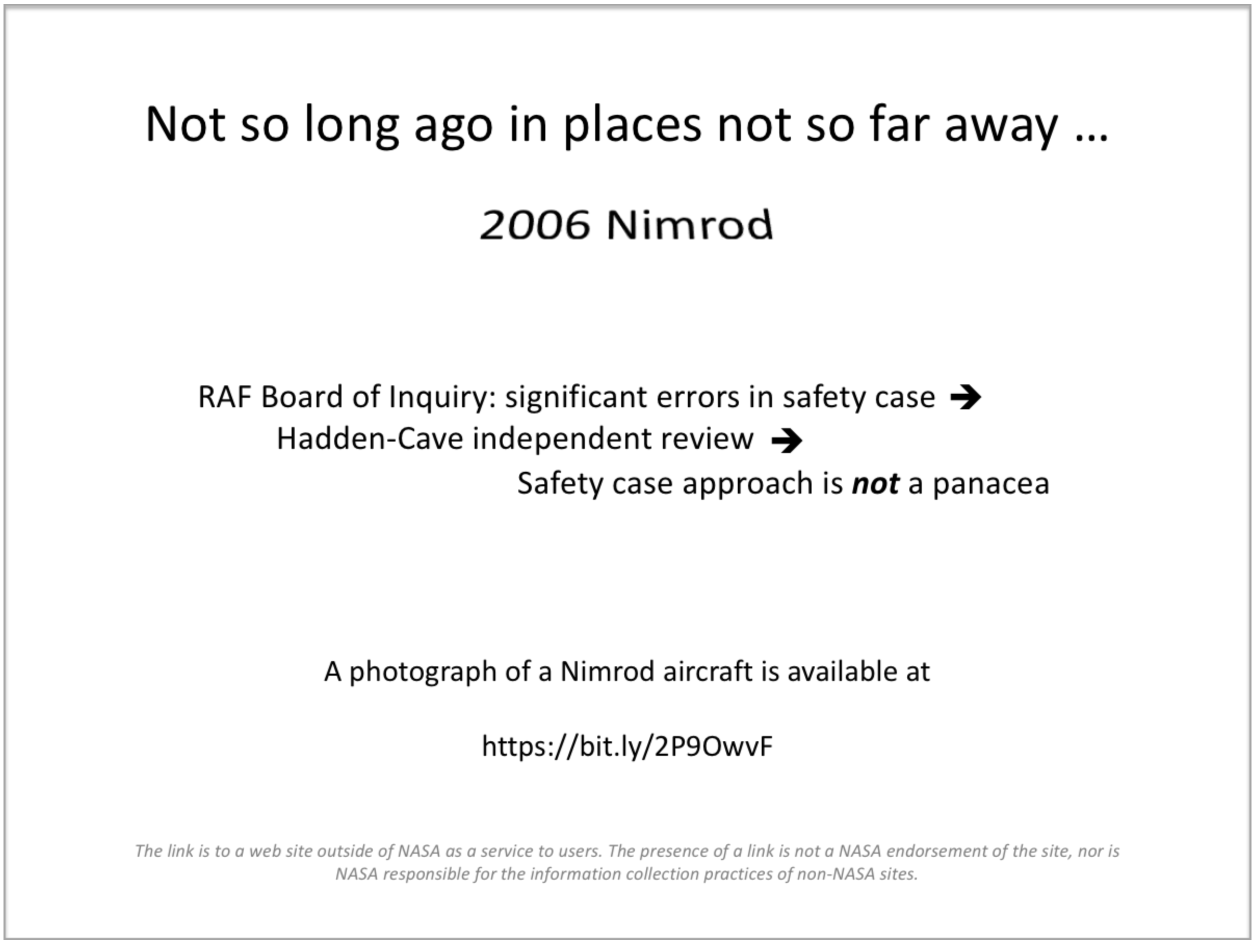
[Question to participants: Any questions or comments at before we continue?]

We now turn to the air.

The last specific accident I'll discuss happened over southern Afghanistan on September 2nd 2006. While on a routine mission in support of NATO and Afghani ground forces, RAF Nimrod X V 230 suffered a catastrophic mid-air fire, leading to the total loss of the aircraft and the death of all twelve on board.

I'll describe what happened borrowing liberally from the Nimod Review report, written by Queens' Counsel, now Sir, Charles Haddon-Cave.

[Haddon-Cave, Charles. 2009. *The Nimrod Review: An independent review into the broader issues surrounding the loss of the RAF Nimrod MR2 Aircraft XV230 in Afghanistan in 2006*. London: The Stationery Office.]



Not so long ago in places not so far away …

2006 Nimrod

RAF Board of Inquiry: significant errors in safety case ➔

Hadden-Cave independent review ➔

Safety case approach is ***not*** a panacea

A photograph of a Nimrod aircraft is available at

https://bit.ly/2P9OwvF

The link is to a web site outside of NASA as a service to users. The presence of a link is not a NASA endorsement of the site, nor is NASA responsible for the information collection practices of non-NASA sites.

"XV230 had taken off … at 09:13 hours that morning. All went according to plan until [about 2 hours later] when, some 1½ minutes after completion of Air-to-Air Refuelling …., the crew were alerted that something was amiss by two almost simultaneous warnings: a fire warning in the bomb bay and a smoke/hydraulic mist warning in the elevator bay." … the camera operator reported `we have flames coming from the rear of the engines on the starboard side'. … the crew immediately commenced emergency drills and … transmitted a MAYDAY whilst diverting to Kandahar airfield."

"Faced with a life-threatening emergency, every member of the crew acted with calmness, bravery and professionalism, and in accordance with their training. They had no chance, however, of controlling the fire."

The aircraft eventually exploded in the air.

An RAF Board of Inquiry presented findings about the causes of the accident in 2007. It concluded that either fuel overflowed from a blow-off value during the refueling or (less likely) fuel leaked from a coupling or pipe; the fuel came into contact with an exposed element of the aircraft's Cross-Feed/Supplementary Cooling Pack duct.

It also found that the Safety Case prepared for the Nimrod between 2002 and 2005 contained significant errors.

Shortly after the Board of Inquiry findings were made public The Secretary of State for Defence announced that an independent review would be conducted to look into the broader issues surrounding the loss of the aircraft. Haddon-Cave was appointed to conduct the review.

His report was published on October 28, 2009, and has been frequently cited in a variety of contexts ever since. I was in London at a System Safety Conference when the report was made public.

Certain critics of safety cases are especially fond of quoting selectively from the report as evidence supporting their negative opinions. Haddon-Cave does indeed identify a number of problems that occurred with the Nimrod safety case: to put it bluntly it was rubbish.

To be a bit more specific, using Haddon-Cave's words for the most part …

"[The] attitude to the [Nimrod Safety Case] was fundamentally affected by the prevailing malaise … that, because the Nimrod had operated safely for over 30 years, it could be assumed that the Nimrod was 'safe anyway' and that, therefore, the [Nimrod Safety Case] exercise did not really matter." "[The contractor's] approach … was flawed and undermined from the outset: it approached the task assuming 'safety' and viewed the [Nimrod Safety Case] task as essentially a documentary or paperwork exercise aimed at proving something that it already knew, i.e. that the Nimrod was safe."

Haddon-Cave noted that the primary purpose of "a 'Safety Case' is to 'identify, assess and mitigate' all potential significant hazards to pieces of equipment, platforms or installations, including hidden, or previously unidentified, hazards. … the drawing up of a 'Safety Case' [is] merely a means to achieving this end, … intended to provide a structure for critical analysis and thinking, or a framework to facilitate a thorough assessment and addressing of serious risks. Unfortunately, in the case of the [Nimrod Safety Case], the production of a 'Report' became an end in itself. Critical analysis descended into a paperwork exercise. "

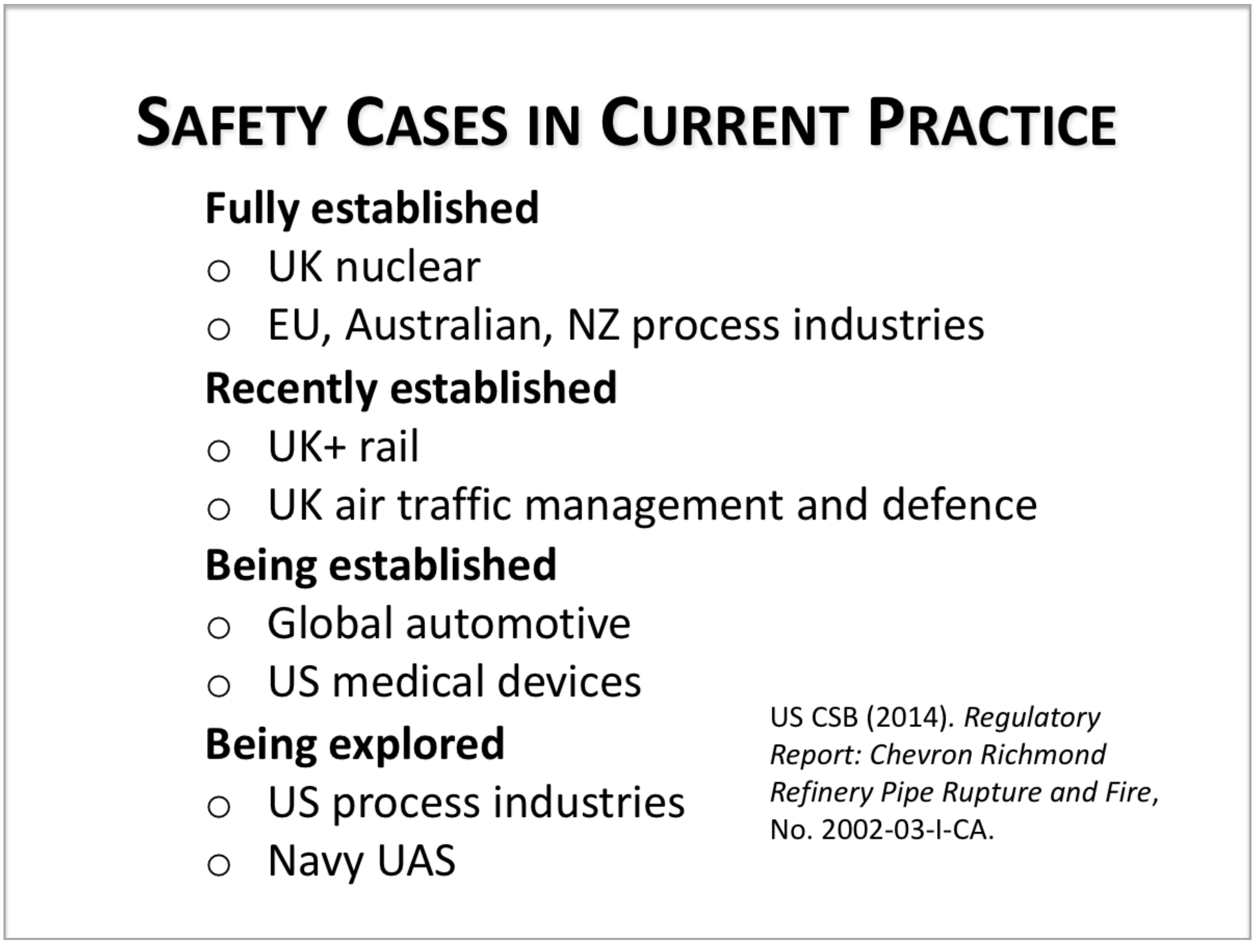
So the real lesson taught by the tragic Nimrod accident is *not* (as some critics would have you to believe) that a safety case approach is a bad idea, but *rather* that a safety case approach is ***not*** a panacea. Creating a document that is called a safety (or an assurance) case does not mean that a good case has been made.

There's a lot more that could be said about the Nimod Review, and about everything else I've mentioned so far, and that are lots of other things that I could mention that I've not mentioned at all, such as the beginnings of research groups at places such as the University of York and City University London, but we'll stop with the history at this point. I'll have more to say about research groups in Module 5.

[Question to participants: Does anyone have any questions about the history?]

Let's move now to talk a bit about current practice.

Discussing current practice is complicated by the paucity of publicly accessible, detailed information about existing industrial cases; Such cases are typically regarded as proprietary information, and thus not available to view. It seems fair to say, however, that the use of safety / assurance cases in real life can be roughly divided in four categories, which I'll now show you.



SAFETY CASES IN CURRENT PRACTICE

Fully established

- UK nuclear
- EU, Australian, NZ process industries

Recently established

- UK+ rail
- UK air traffic management and defence

Being established

- Global automotive
- US medical devices

Being explored

- US process industries
- Navy UAS

US CSB (2014). *Regulatory Report: Chevron Richmond Refinery Pipe Rupture and Fire,* No. 2002-03-I-CA.

Among domains in which (safety) cases are fully established are the UK nuclear industry (as I mentioned already in the discussion of Windscale), many of the EU process industries (think Seveso), and also process industries in Australia and New Zealand.

Recently established domains include rail in the UK (and much of the EU), UK air traffic management, and various aspects of UK defence.

Domains in which the use of cases is in the process of being established include the global automotive industry, and certain aspects of US medical devices, particularly infusion pumps.

Finally, domains that are exploring use include Some US process industries, and the US Navy, at least in respect to UAS.

The US Chemical Safety and Hazard Investigation Board published in 2014 what they call a “Regulatory Report” concerning the 2012 Chevron Richmond Refinery pipe rupture and fire. Nearly all of the report deals with whether a “safety case regulatory regime” might be appropriate, reaching the conclusion that the CSB believes that adopting attributes of “more robust safety management regulatory regimes from around the world” “would greatly enhance existing federal and California process safety regulations.” The report is available at

https://www.csb.gov/assets/1/20/chevron_regulatory_report_06272014.pdf

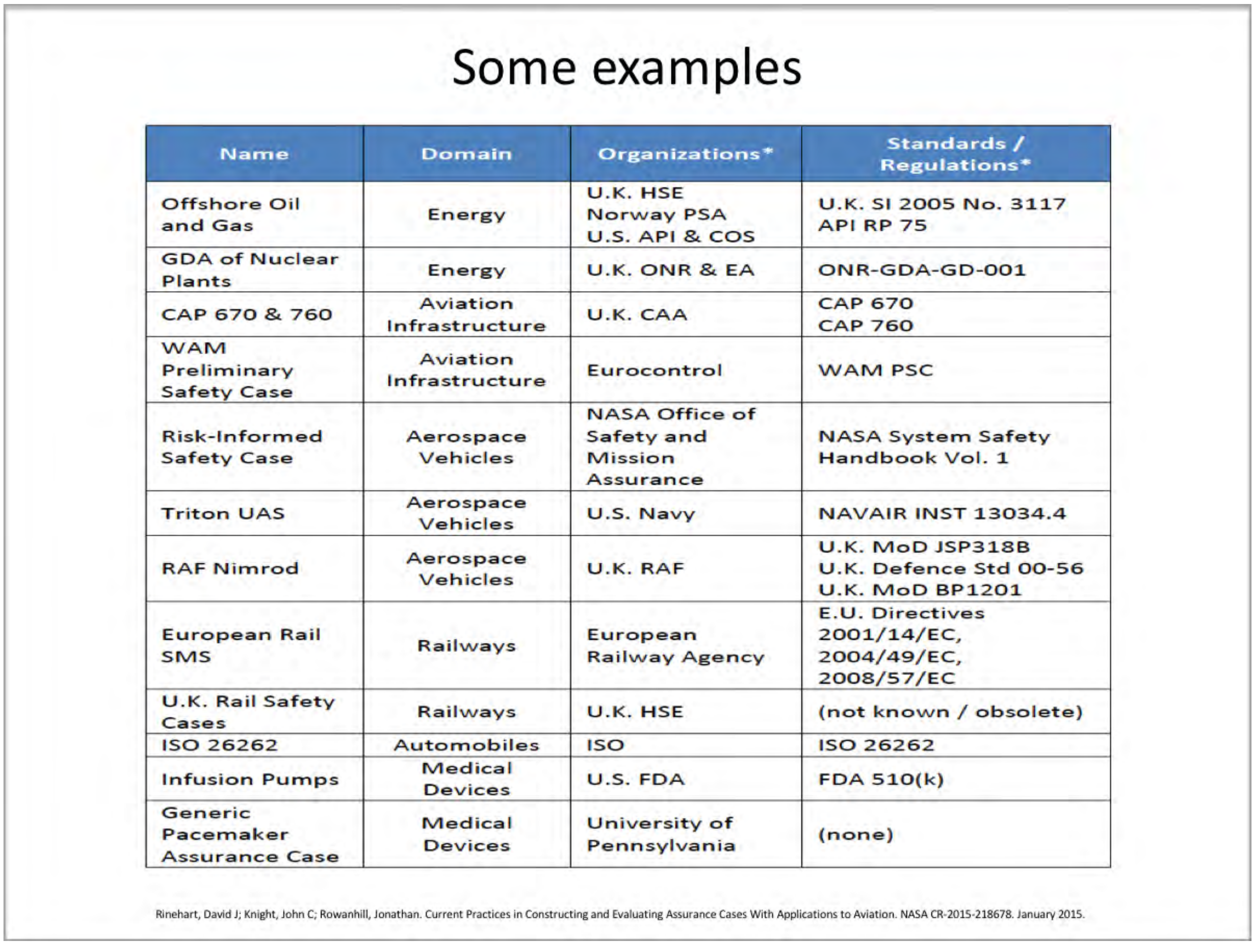
I believe it is important to note that the majority of existing experience in using safety cases has tended to involve 'services' rather than 'systems'. It's been more about how a plant is operated than about specifics of the design of a particular system within the plant.

Please don't look at this slide as a definitive, all-inclusive breakdown of current practice; it is simply a rough breakdown, which I believe to be mostly accurate at the current time. Some other folks may dispute the categorizations in some areas, and may have additions, also.

One could certainly suggest that much more could be included in the "being explored" category (FAA and NASA, for example), but I've tried to restrict this listing to domains in which there exists evidence of active, real-life, practical activity of some sort, and not just research efforts. Concerning research efforts, we at NASA Langley published earlier this year (that is, 2015) a contractor report developed by folks from Saab Sensis and Dependable Computing that, among other things, discusses the results of a literature search looking for examples of published assurance cases.

[Rinehart, David J., Knight, John C., Rowanhill, Jonathan. 2015. *Current Practices in Constructing and Evaluating Assurance Cases with Applications to Aviation*. NASA CR-2015-218678.]

Here is an excerpt from a table in the report. I won't go into details, but I will note the column that mentions some of the relevant standards or regulations that exist in certain domains.



Some examples

Name	Domain	Organizations*	Standards / Regulations*
Offshore Oil and Gas	Energy	U.K. HSE Norway PSA U.S. API & COS	U.K. SI 2005 No. 3117 API RP 75
GDA of Nuclear Plants	Energy	U.K. ONR & EA	ONR-GDA-GD-001
CAP 670 & 760	Aviation Infrastructure	U.K. CAA	CAP 670 CAP 760
WAM Preliminary Safety Case	Aviation Infrastructure	Eurocontrol	WAM PSC
Risk-Informed Safety Case	Aerospace Vehicles	NASA Office of Safety and Mission Assurance	NASA System Safety Handbook Vol. 1
Triton UAS	Aerospace Vehicles	U.S. Navy	NAVAIR INST 13034.4
RAF Nimrod	Aerospace Vehicles	U.K. RAF	U.K. MoD JSP318B U.K. Defence Std 00-56 U.K. MoD BP1201
European Rail SMS	Railways	European Railway Agency	E.U. Directives 2001/14/EC, 2004/49/EC, 2008/57/EC
U.K. Rail Safety Cases	Railways	U.K. HSE	(not known / obsolete)
ISO 26262	Automobiles	ISO	ISO 26262
Infusion Pumps	Medical Devices	U.S. FDA	FDA 510(k)
Generic Pacemaker Assurance Case	Medical Devices	University of Pennsylvania	(none)

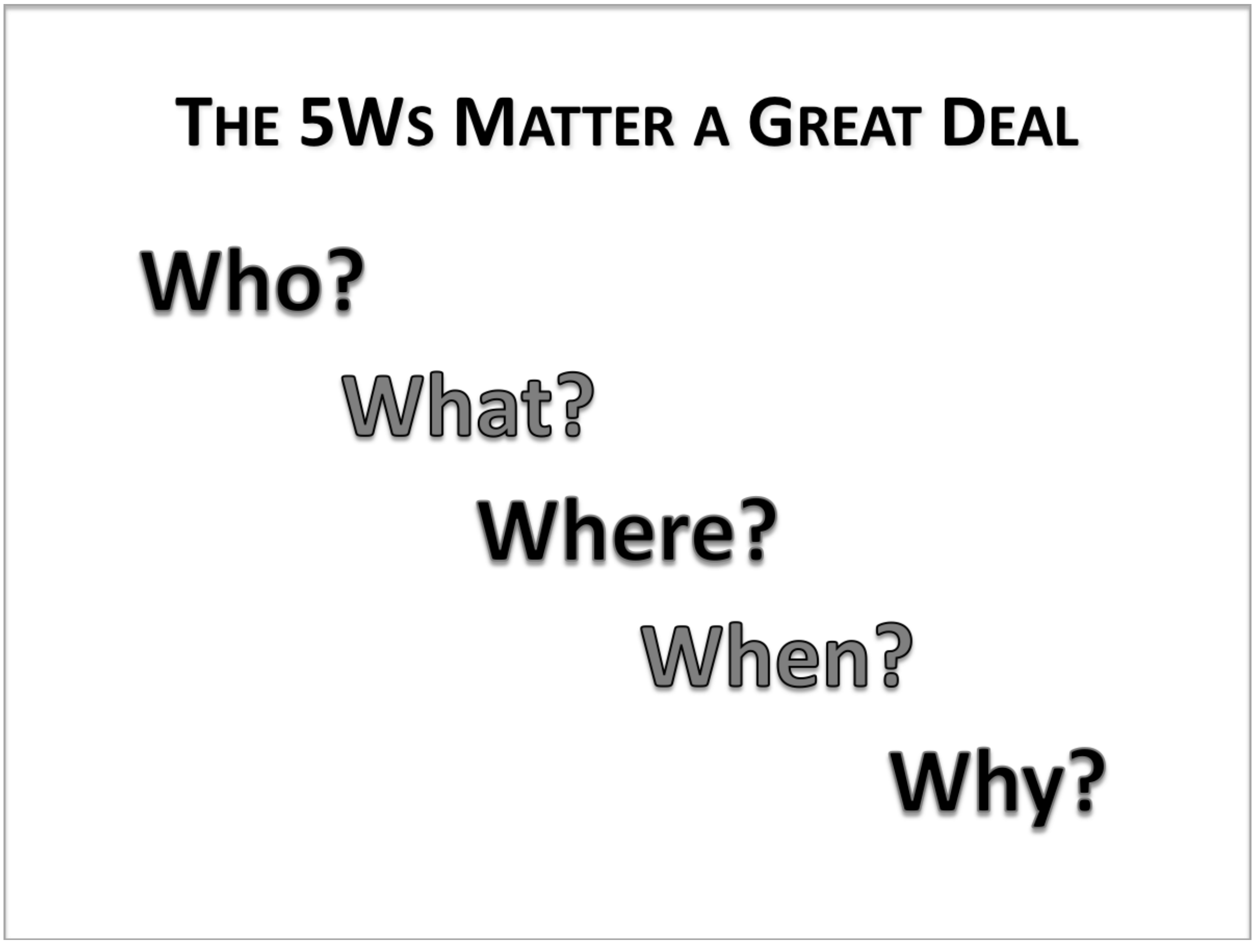
Rinehart, David J; Knight, John C; Rowanhill, Jonathan. Current Practices in Constructing and Evaluating Assurance Cases With Applications to Aviation. NASA CR-2015-218678. January 2015.

In particular, there seems to be some evidence that the voluntary automotive standard ISO 26262 is growing in influence[1], which is a major reason that I included the global automotive industry in the *being established* category.

Concerning the FDA, based on conversations that I and others have had with some folks within that organization, they seem to be experiencing some of the same things that were highlighted in the Ladbroke Grove and Nimrod discussions earlier: namely, that some assurance cases are quite badly done.

I'm ready now to move on to discuss matters directly related to our 3rd and 4th learning objectives for this module, but before I do so, does anyone have a question?

Concerning possible lessons taught from the past, I think that a helpful way to think about such lessons is by using the famous five Ws, because history and practice seems to suggest that the answers to the questions "Who? What? Where? When? Why?" matter a great deal when it comes to applying assurance cases.



THE 5WS MATTER A GREAT DEAL

Who?

What?

Where?

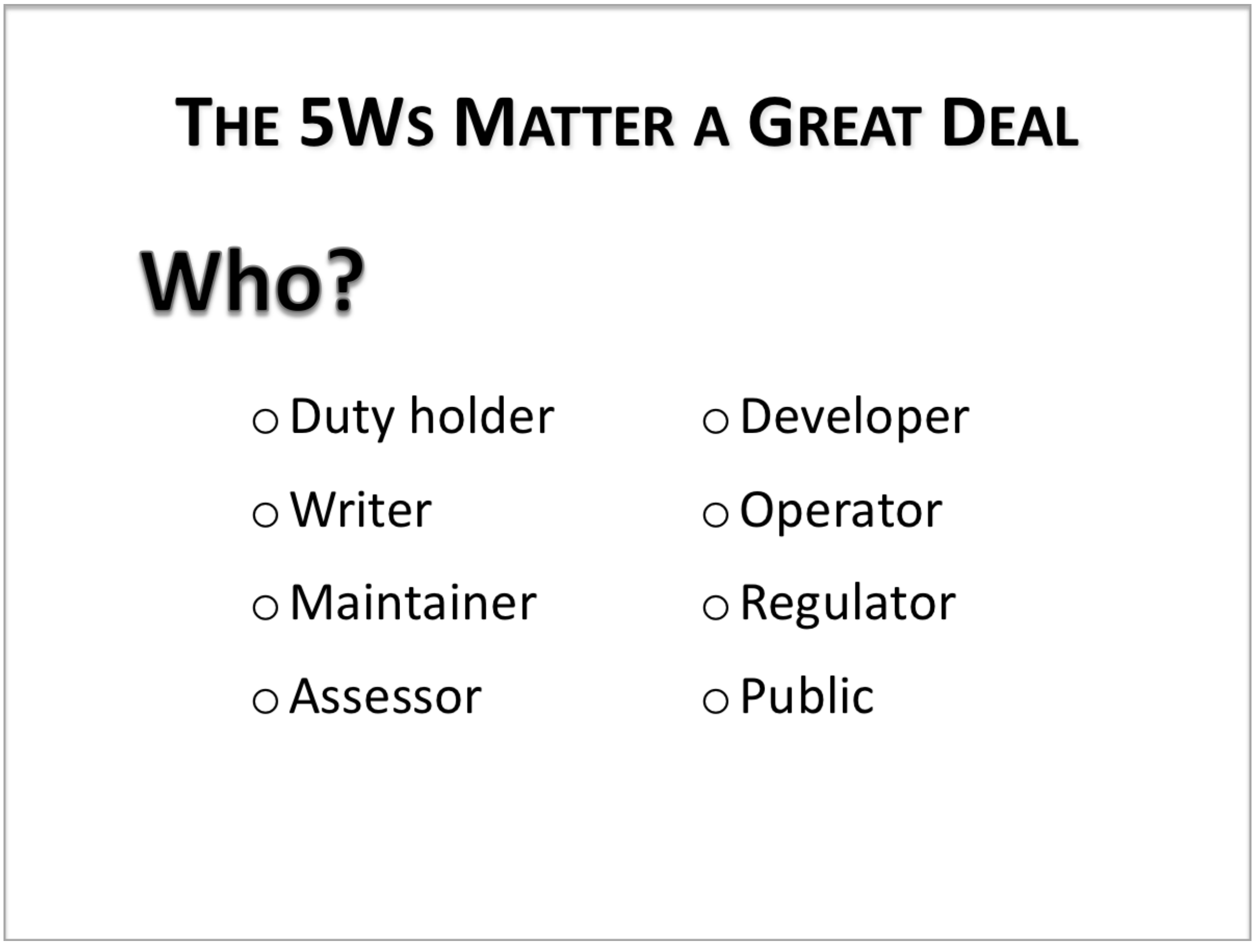
When?

Why?

In fact, they matter so much that one might perhaps accurately say that the overarching lesson taught thus far is "Carefully ask and answer the 5Ws."

[1] Since this module was developed in 2015, the influence of the standard has continued to grow.

Let’s look at each of these questions briefly.



THE 5WS MATTER A GREAT DEAL

Who?

- ○ Duty holder
- ○ Writer
- ○ Maintainer
- ○ Assessor
- ○ Developer
- ○ Operator
- ○ Regulator
- ○ Public

To successfully apply an assurance case approach, you need to understand well *who* is necessarily involved in the application: eight categories of *who*s are listed here.

The *duty holder* refers to the people or organizations obliged to preserve the properties we’re concerned about in the system or service (so, for example, safety for a safety case).

For simplicity, let’s assume for the rest of this discussion that we’re interested in safety of a system, so we don’t have to say ‘system or service’ and we can talk about a ‘safety case’. Let’s further assume that the term ‘organization’ is a short hand for the longer phrase meaning something like “person, people, organization, or organizations.”

The *developer* refers to the organization that will make design decisions and implement the system.

The *writer* refers to the organization that writes the safety case. One can argue that the Nimrod example suggests that it may not be a good idea to have an organization involved whose sole role is that of *writer*; though it is certainly true that the writing of safety cases has sometimes been contracted out to third parties without dire consequences.

The *operator* refers to the organization who will operate the system. In the process industries, the operator is often also the duty holder. This need not be the case. For example, consider pilots and drivers.

The *maintainer* refers to the organization who will service the system during its operational life. The maintainers might be disjoint with the duty holders, developers, and operators. For example, an operator might delegate maintenance to a contractor.

The *regulator* refers to the organization explicitly legally tasked with supervision of a particular system. Not all systems have regulators.

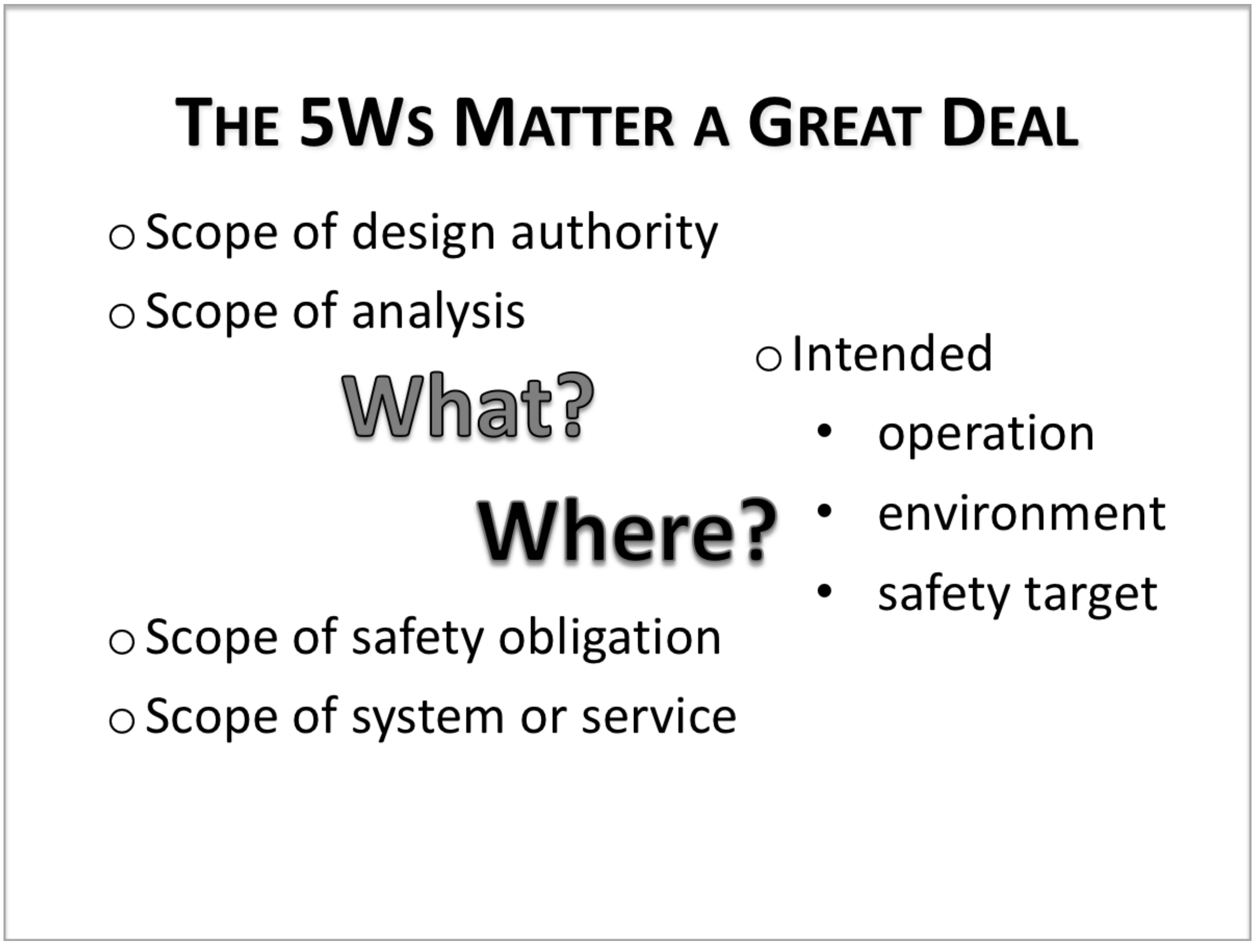
The *assessor* refers to the organization that explicitly audits or assesses the safety case. An assessor might or might not be a regulator. For example, an automobile manufacturer conforming to ISO 26262 might hire an outside organization to assess conformance even though law or regulation does not require conformance.

Finally, the *public* refers to anyone who may use or in any way be affected by the system in question and who is not in one of the other seven categories.

Those are the eight *who*s that are necessarily involved in the application of an assurance case approach. Trying to apply assurance cases without thinking about these *who*s is not a good idea.

[Question to participants: Are there any questions about “Who?”]

We’ll consider “What?” and “Where?” together.



The 5Ws Matter a Great Deal

○ Scope of design authority

○ Scope of analysis

What?

Where?

○ Intended
- operation
- environment
- safety target

○ Scope of safety obligation

○ Scope of system or service

These two questions encompass issues about scope and intent.

Let’s talk about intent first.

What and where is the *intended operation* of the system? (Again we’re using system as a shorthand for system or service). What’s the system supposed to do? What’s its mission? What is it *not* supposed to do? What are foreseeable types of misuse?

What and where is the *intended environment* of the system?

This includes any and all features of the (intended) time or place of operation. For example, will an aircraft perform taxi, takeoff, and landing in conditions of extreme cold, or extreme heat, or sandstorms?

What is the *intended safety (or assurance) target?*

This question might be answered By identifying an appropriate standard that will be followed (for example, software contributions to system risk should be controlled by applying DO-178C) or by identifying an appropriate risk acceptance test (for example, risks should be managed As Low as Reasonably Practicable - ALARP, So Far As Is Reasonably Practicable - SFAIRP, or Globally at least as good - GAMAB) As an aside: We could spend a lot of time talking about differences among these, but we won’t (ALARP is hazard based. SFAIRP is precaution based. GAMAB is comparison based.)

“What?” and “Where?” also involve issues concerning scope.

Scope of design authority refers to what the developers are able to control.

Scope of analysis refers to what part of the system is being considered in the assurance case. Perhaps it is not the entire system but only certain aspects of it.

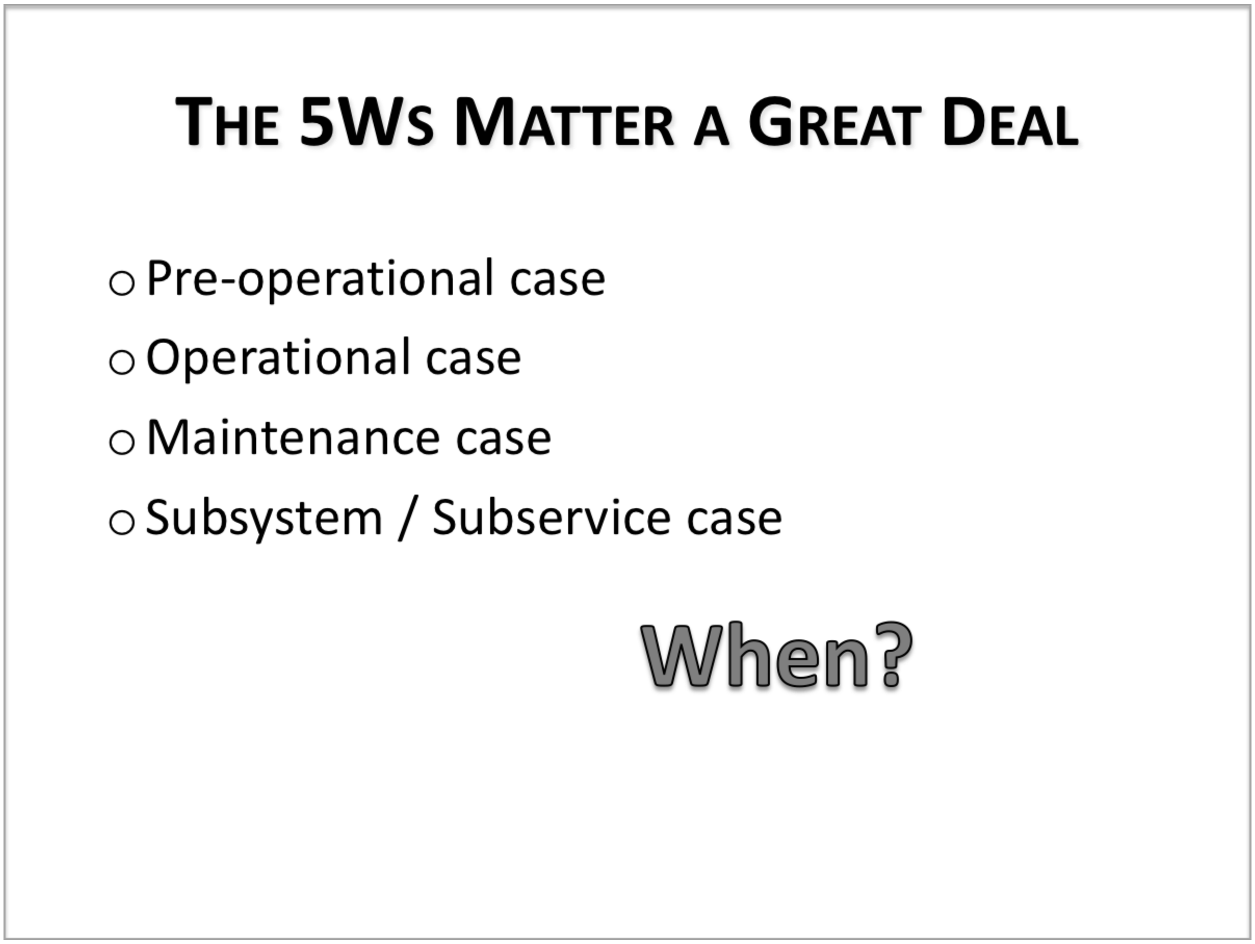
Scope of safety obligation refers to what the duty holder is obliged to consider.

Scope of system or service refers to the full extent of what we need to consider the safety implications of. This is very much related to intended operation.

Trying to apply assurance cases without thinking about these *what*s and *where*s is not a good idea.

[Question to participants: Are there any questions about “What?” and “Where?”]

The “When” question concerns the timing of the creation of an assurance case or cases. Some possibilities are shown here on the slide.



The 5Ws Matter a Great Deal

- Pre-operational case
- Operational case
- Maintenance case
- Subsystem / Subservice case

When?

A *Pre-operational case* comes from the system developers. The scope is limited to the system design and implementation, with operations assumed for the purpose of safety analysis. It is used to make release-to-service decisions. There can and should be several early versions of the pre-operational case. Early versions will describe the system as it will be (as far as is known at the time of writing); the final pre-operational case should describe the system as actually built. Note that the pre-operational cases necessarily lack evidence from experience of operation, and thus are based on assumptions about operation.

In an *operational case* the scope is the actual operation of the system in real life. System design issues are excluded (except for modification and monitoring). It addresses safety of operators and (if relevant) the public. It relies on a pre-operational safety case for claims about what the system does and (if appropriate) supports that case with information that shows that assumptions made in the pre-operational case about operations are correct.

Note that if the developer is not the operator the writer of the operational safety case might not the same as the writer of the pre-operational case.

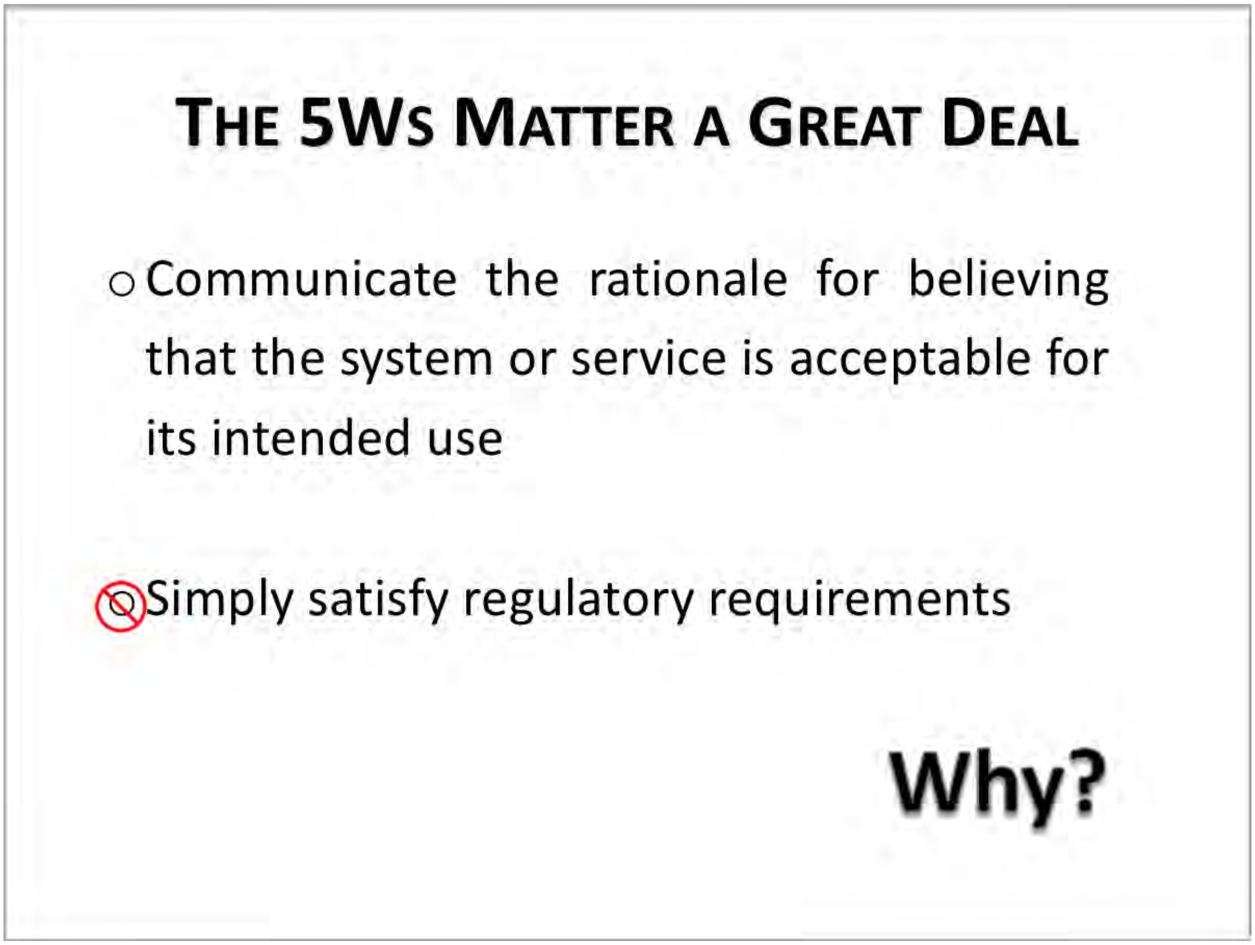
A *maintenance case* concerns, as you may suspect, how the system is being maintained, and should include discussion about the safety of the maintainers, and arguments concerning the maintenance assumptions made in the pre-operational case.

For some systems there may be separate cases for components or *subsystems* These might be cited by the overall pre-operational, operational, or maintenance cases to justify conclusions about the component or subsystem contribution to system risks or their mitigation.

In applying assurance cases the “When” question must be asked and answered. There’s some evidence to suggest that answering it with a single time may tend to be unwise. That is, writing a pre-operational case only, while ignoring operational and maintenance cases may fail to ensure the level of safety that is desired.

[Question to participants: Are there any questions about “When?”]

The final W question in this discussion, but almost certainly the first in a temporal sense, is “Why are you doing it?”



The 5Ws Matter a Great Deal

○ Communicate the rationale for believing that the system or service is acceptable for its intended use

🚫 Simply satisfy regulatory requirements

Why?

There’s a good answer and a bad answer to this question. Creating a case simply to satisfy regulatory requirements is the bad answer. Creating an assurance case to communicate the rationale for believing that the system or service is acceptable for its intended use is the good answer.

Each of the “Who” parties we talked about earlier should gain something from this communication. For example, Consider a pre-operational safety case written by the developer, who is also the duty holder.

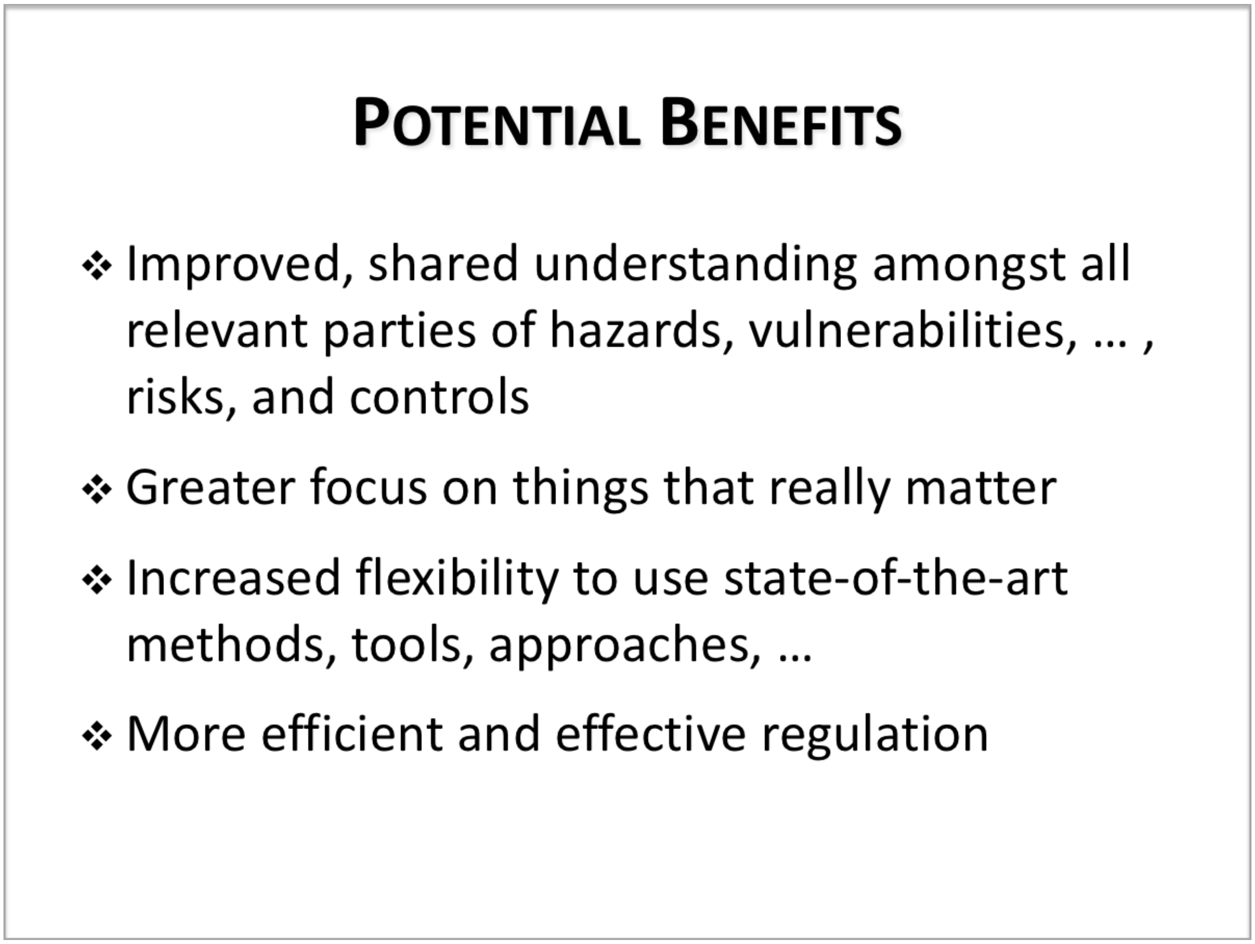
The writer / developer / duty holder, who must articulate the rationale, might gain a more detailed understanding of that rationale, and recognize possible deficiencies.

A regulator might gain insight into whether applicable law or regulation has been complied with. An auditor might gain insight into what the duty holder considers adequately safe, what hazards they think are most in need of attention, what options were considered, and how they have gone about implementing the chosen options.

An operator might gain a better understanding of what a system or service is meant to be or do in order to be safe, thus putting that operator in a better position to notice operational realities that would make the system or service less safe than intended. A maintainer might gain a better understanding of which hazards a system's developers considered most in need of addressing and how they intended to address them.

Finally, if given access to the safety case, The public, whom might be harmed by the system or service, might gain a better understanding of how safe 'adequately safe' actually is.

This discussion leads us naturally into talking a bit about potential benefits which I've summarized here on this slide in four points.



POTENTIAL BENEFITS

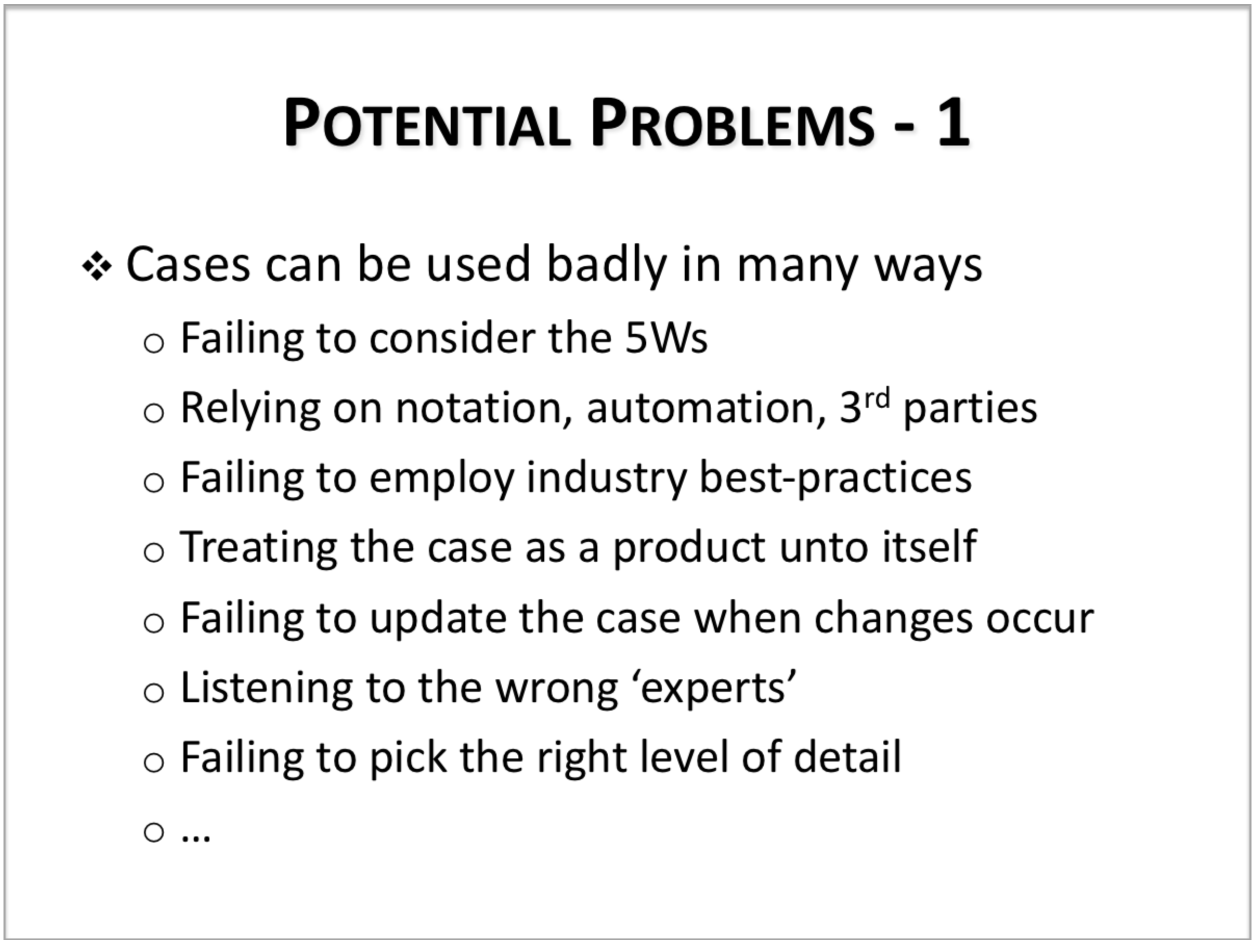
- Improved, shared understanding amongst all relevant parties of hazards, vulnerabilities, … , risks, and controls
- Greater focus on things that really matter
- Increased flexibility to use state-of-the-art methods, tools, approaches, …
- More efficient and effective regulation

The first of these is directly related to what we've just discussed: An improved, shared understanding amongst all relevant parties of hazards, vulnerabilities, … , risks, controls (and other things you might want to put here.)

There is also the potential for a greater focus on things that really matter, and for increased flexibility to use state-of-the-art methods, tools, approaches, and whatever else can be state-of-the-art.

Consequently, these things possibly could lead to more efficient and effective regulation.

These benefits are not givens, however, as we saw in several of the examples from history we discussed earlier.



Potential Problems - 1

- Cases can be used badly in many ways
  - Failing to consider the 5Ws
  - Relying on notation, automation, 3rd parties
  - Failing to employ industry best-practices
  - Treating the case as a product unto itself
  - Failing to update the case when changes occur
  - Listening to the wrong ‘experts’
  - Failing to pick the right level of detail
  - …

Cases can be used badly in many ways; I’ve listed seven of them on this slide.

Failing to consider the 5Ws.

Relying on notation, automation, or third parties.

Failing to employ industry best practices.

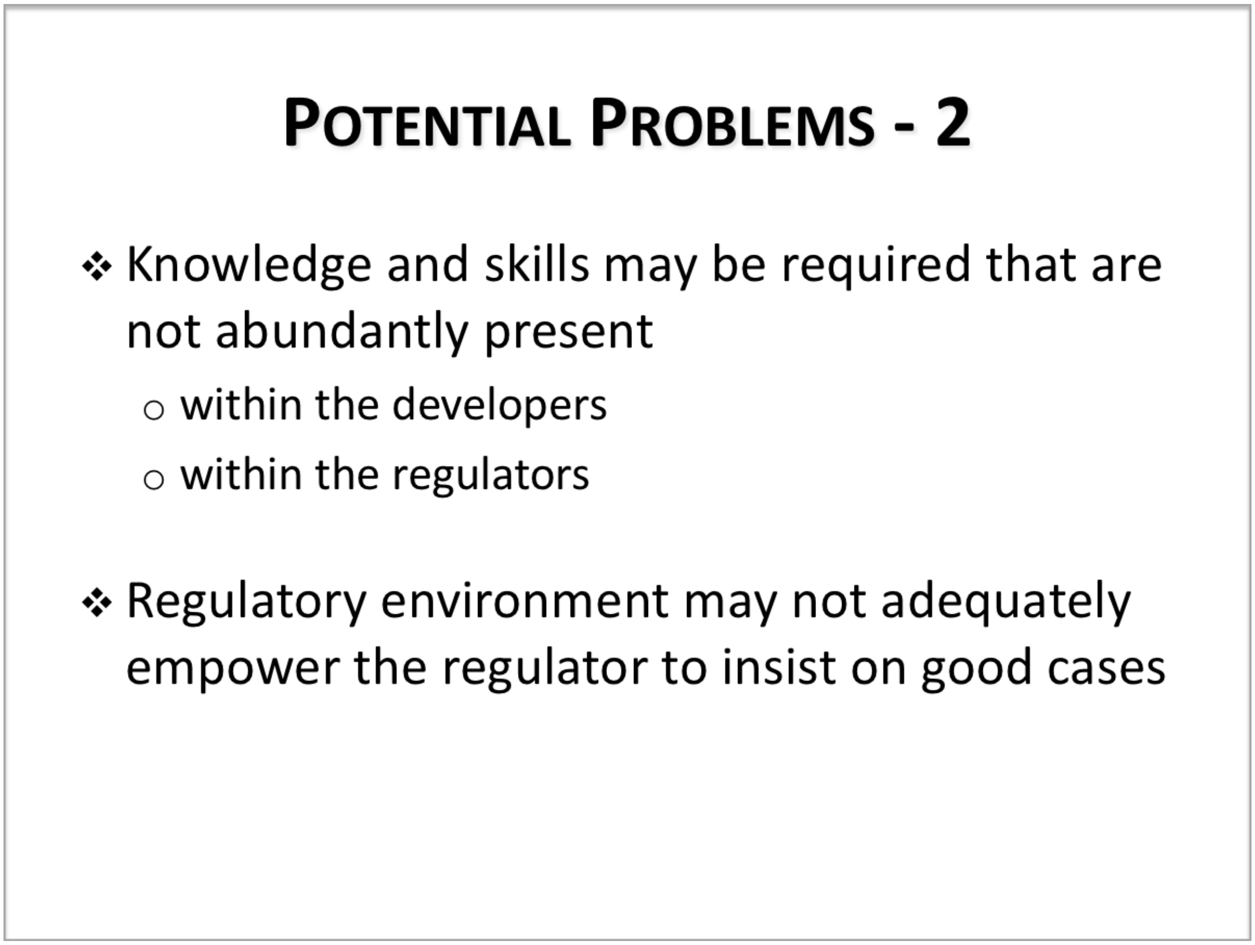
Treating the case as a product unto itself.

Failing to update the case when changes occur.

Listening to the wrong ‘experts’ (with the growing popularity of assurance cases, there’s also a growing number of folks who style themselves as experts, but not all of them know what they’re talking about).

Failing to pick the right level of detail.

Doing cases badly are not the only potential problems.



Potential Problems - 2

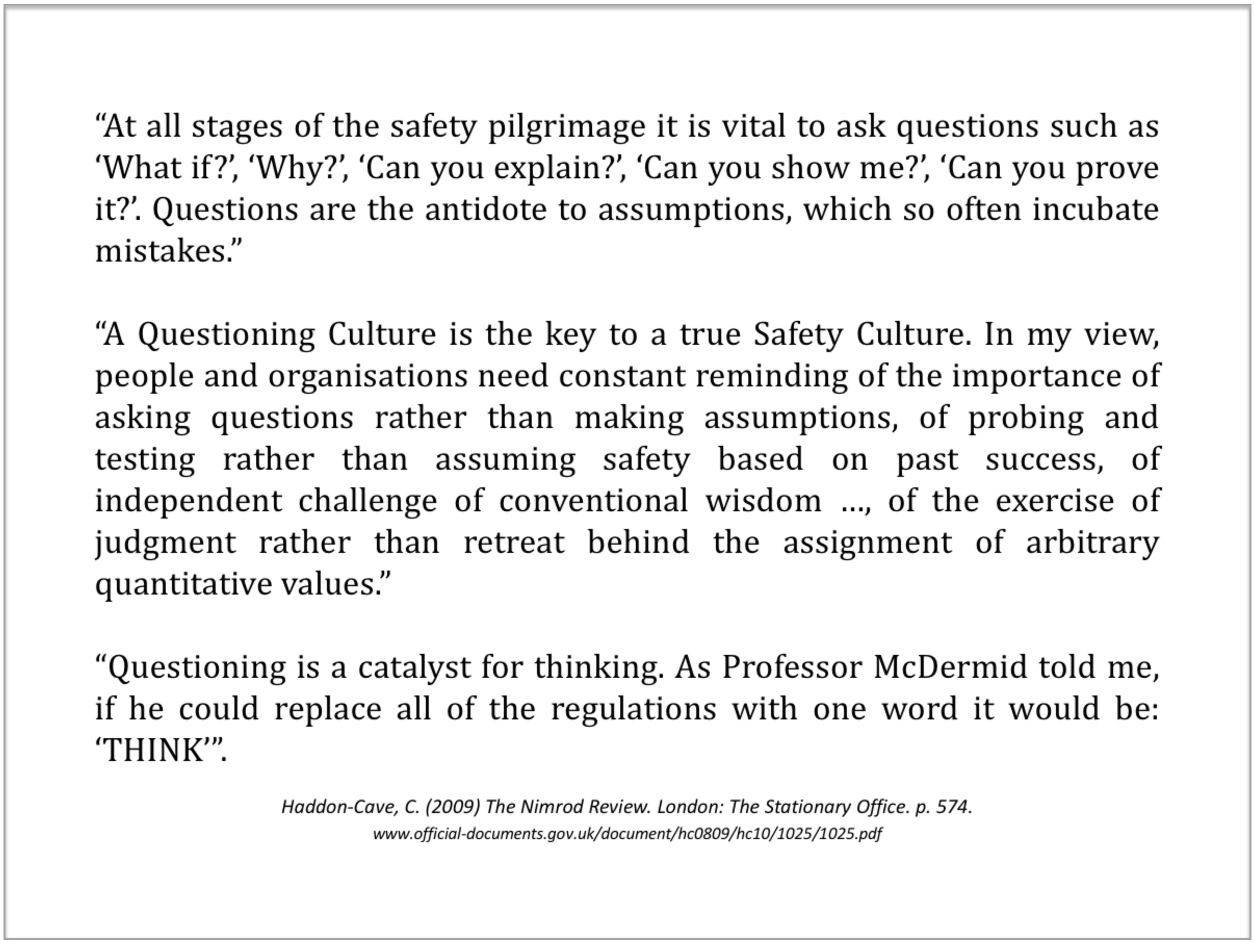
- Knowledge and skills may be required that are not abundantly present
  - within the developers
  - within the regulators
- Regulatory environment may not adequately empower the regulator to insist on good cases

Two others include the possibility that knowledge and skills may be required that are not abundantly present within the developers, or within the regulators; and, the possibility that the regulatory environment may not adequately empower the regulator to insist on good cases.

Even if a regulator has adequate skills, if the regulatory environment does not allow them to reject poor assurance cases, problems will certainly occur[2].

We're almost done, but before taking questions and comments, I want to show and read to you a superb quotation from the Haddon-Cave report.

[2] In the three years since this module was first presented, the importance of this particular problem relative to the other problems listed seems to have increased.



"At all stages of the safety pilgrimage it is vital to ask questions such as 'What if?', 'Why?', 'Can you explain?', 'Can you show me?', 'Can you prove it?'. Questions are the antidote to assumptions, which so often incubate mistakes."

"A Questioning Culture is the key to a true Safety Culture. In my view, people and organisations need constant reminding of the importance of asking questions rather than making assumptions, of probing and testing rather than assuming safety based on past success, of independent challenge of conventional wisdom …, of the exercise of judgment rather than retreat behind the assignment of arbitrary quantitative values."

"Questioning is a catalyst for thinking. As Professor McDermid told me, if he could replace all of the regulations with one word it would be: 'THINK'".

Haddon-Cave, C. (2009) The Nimrod Review. London: The Stationary Office. p. 574.
www.official-documents.gov.uk/document/hc0809/hc10/1025/1025.pdf

At all stages of the safety pilgrimage it is vital to ask questions such as "What if?", "Why?", "Can you explain?", "Can you show me?", "Can you prove it?". Questions are the antidote to assumptions, which so often incubate mistakes.

A Questioning Culture is the key to a true Safety Culture. In my view, people and organisations need constant reminding of the importance of ***asking questions*** rather than making assumptions, of ***probing and testing*** rather than assuming safety based on past success, of ***independent challenge*** of conventional wisdom or shibboleths, of the ***exercise of judgment*** rather than retreat behind the assignment of arbitrary quantitative values.

Questioning is a catalyst for thinking. As Professor McDermid told me, if he could replace all of the regulations with one word it would be: "THINK".

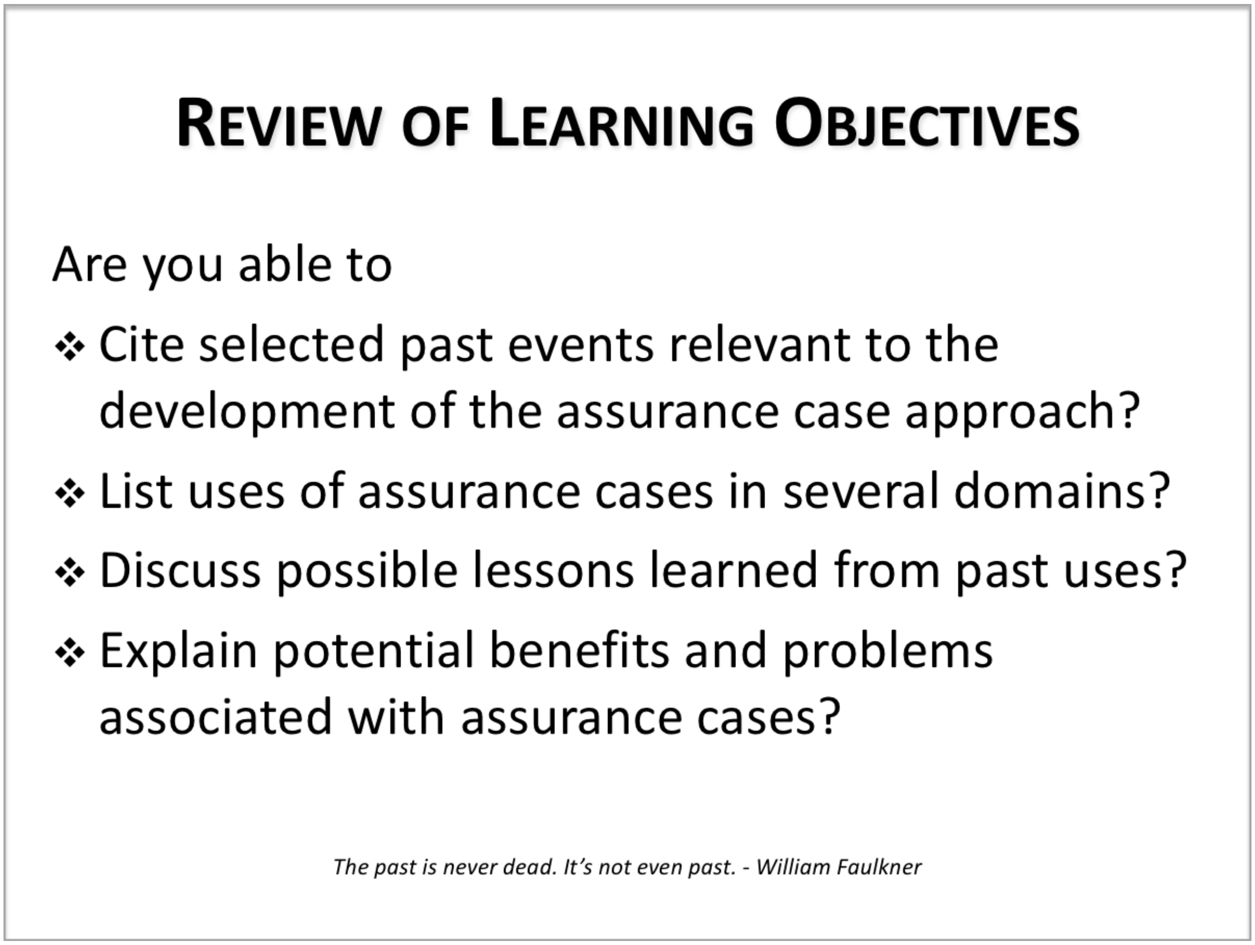
In my opinion the greatest potential benefit of the assurance case approach is that, used properly, it can force people to think more deeply than they otherwise would.

The greatest potential problem of the assurance case approach is that, if used improperly, it can cover up shoddy thinking.

[Question for participants: Any questions or comments before we end by reviewing the learning objectives?]

At the beginning, I listed four things that I hoped you'd be able to do by the end of this module.

Here are those four things recast in the form of questions. Think to yourself how you'd answer these questions.



REVIEW OF LEARNING OBJECTIVES

Are you able to

❖ Cite selected past events relevant to the development of the assurance case approach?

❖ List uses of assurance cases in several domains?

❖ Discuss possible lessons learned from past uses?

❖ Explain potential benefits and problems associated with assurance cases?

The past is never dead. It's not even past. - William Faulkner

After you've thought about the questions for a little bit, please ask me any questions that you still have for me.

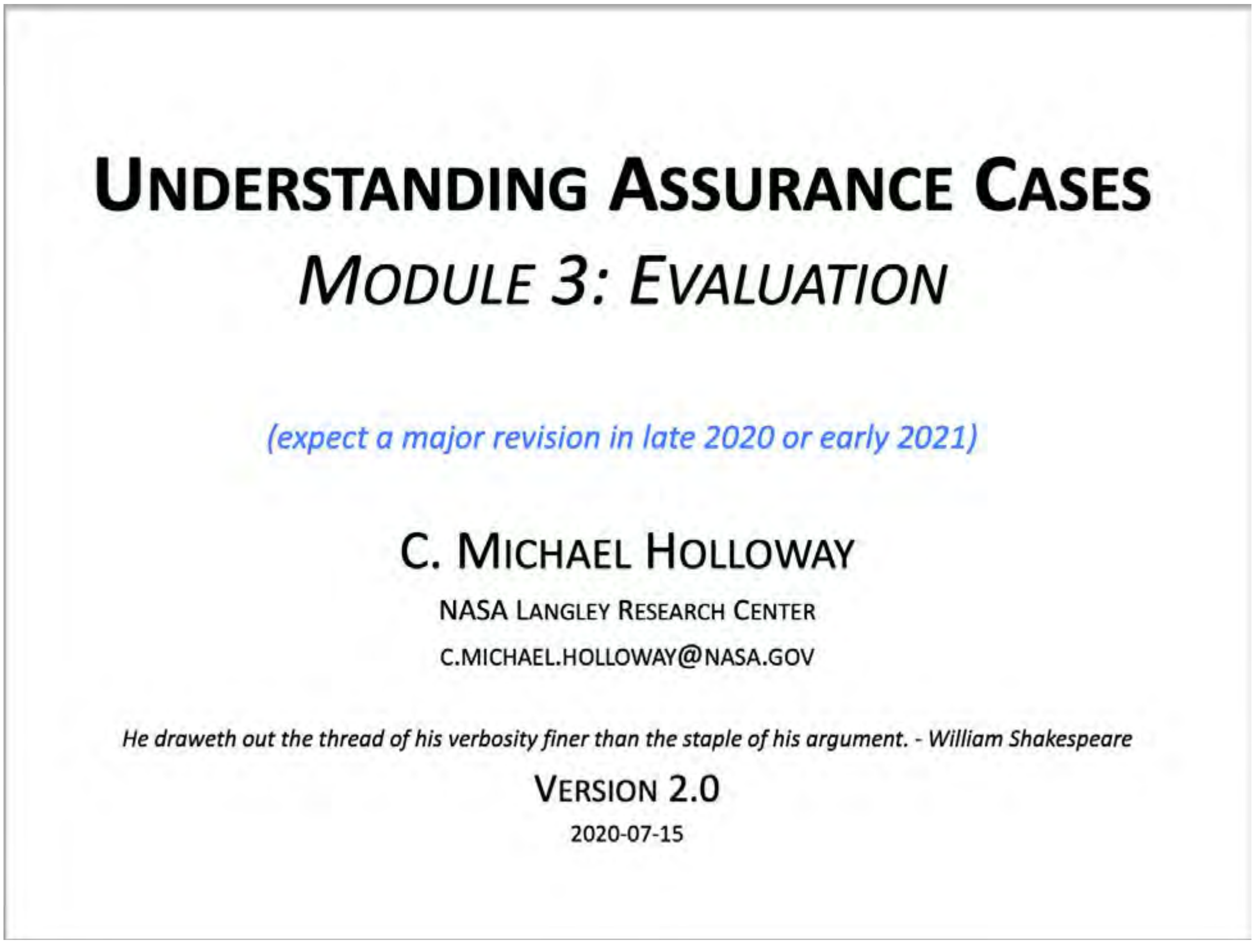
If you have questions or comments about this material, contact its author at `c.michael.holloway@nasa.gov`.

Understanding Assurance Cases: An Educational Presentation in Five Parts

Module 3: Evaluation

C. Michael Holloway
c.michael.holloway@nasa.gov

Senior Research Computer Engineer
Safety-Critical Avionics Systems Branch
NASA Langley Research Center, Hampton, Virginia, U.S.A.



UNDERSTANDING ASSURANCE CASES

MODULE 3: EVALUATION

(expect a major revision in late 2020 or early 2021)

C. MICHAEL HOLLOWAY

NASA LANGLEY RESEARCH CENTER

C.MICHAEL.HOLLOWAY@NASA.GOV

He draweth out the thread of his verbosity finer than the staple of his argument. - William Shakespeare

VERSION 2.0

2020-07-15

This material was originally created in 2015-16, as part of the Explicate '78 project. The project was supported in substantial part by the Assurance Case Applicability to Digital Systems task under the reimbursable interagency agreement with the Federal Aviation Administration for Design, Verification, and Validation of Advanced Digital Airborne Systems Technology (IAI-1073 Annex 2 for NASA; DTFACT-10-X0008, Modification 0004 for the FAA). The original presentations were delivered to a selected group of FAA civil servants and NASA Langley personnel. The audio was recorded and partial transcripts (containing only the words spoken by the presenter, Mr. Holloway) produced. The intent from the beginning was to collect the material into a form that could be made available publicly. The text adheres closely to the original transcript, except where changes have been made to the original presentation since it was first given, as part of work for for NASA IA-303333/FAA IA NO 692M15-19-T-00029 Annex 1/TO 1. The full collection consists of six documents (including this one), which are available electronically through `https://shemesh.larc.nasa.gov/arg/uac.html`.

Welcome to the third module in an educational series about Understanding Assurance Cases. [Significant changes will be made to this module by mid 2021.]

In this module, we will examine the ***Evaluation*** of assurance cases. If you have not already completed Modules 1 and 2 (Foundation and Application respectively), please stop reading this document, and complete both Foundation and Application before continuing[1].

In evaluating an assurance case one hopes the occasion will not arise to say of the writer of the case what one Shakespeare character said of another in *Love's Labour's Lost:* "He draweth out the thread of his verbosity finer than the staple of his argument."

[Shakespeare, William. *Love's Labour's Lost*, act v, scene i, lines 1750-51.]

As with all the modules, feel free to interrupt me at *any* point if you have a burning question. I reserve the right to defer the answer to later on that's appropriate, but otherwise I'll do my best to answer it.

In today's module, there will be a few times when I'll ask you to do a bit of work on your own --- nothing substantial or time-consuming, but I hope it'll help improve your understanding of the material.

[Question to participants: Does anyone have any questions or comments that you want to make now, before we proceed further?]

Let's list our learning objectives.

By the time we're finished today, I hope that you'll be able to do at least these four things.

One, identify *positive* properties that an assurance case *should* have.

Two, identify *negative* properties that an assurance case *should not* have.

Three, you should also be able to enumerate steps for evaluating an assurance case.

Four, I expect you to be able suggest potential corrections for selected deficiencies.

As I'm sure you realize, when we're done with this module, you're not going to be an expert in evaluating assurance cases (unless you're one already), but you should be fairly well acquainted with much of what's involved in evaluating them.

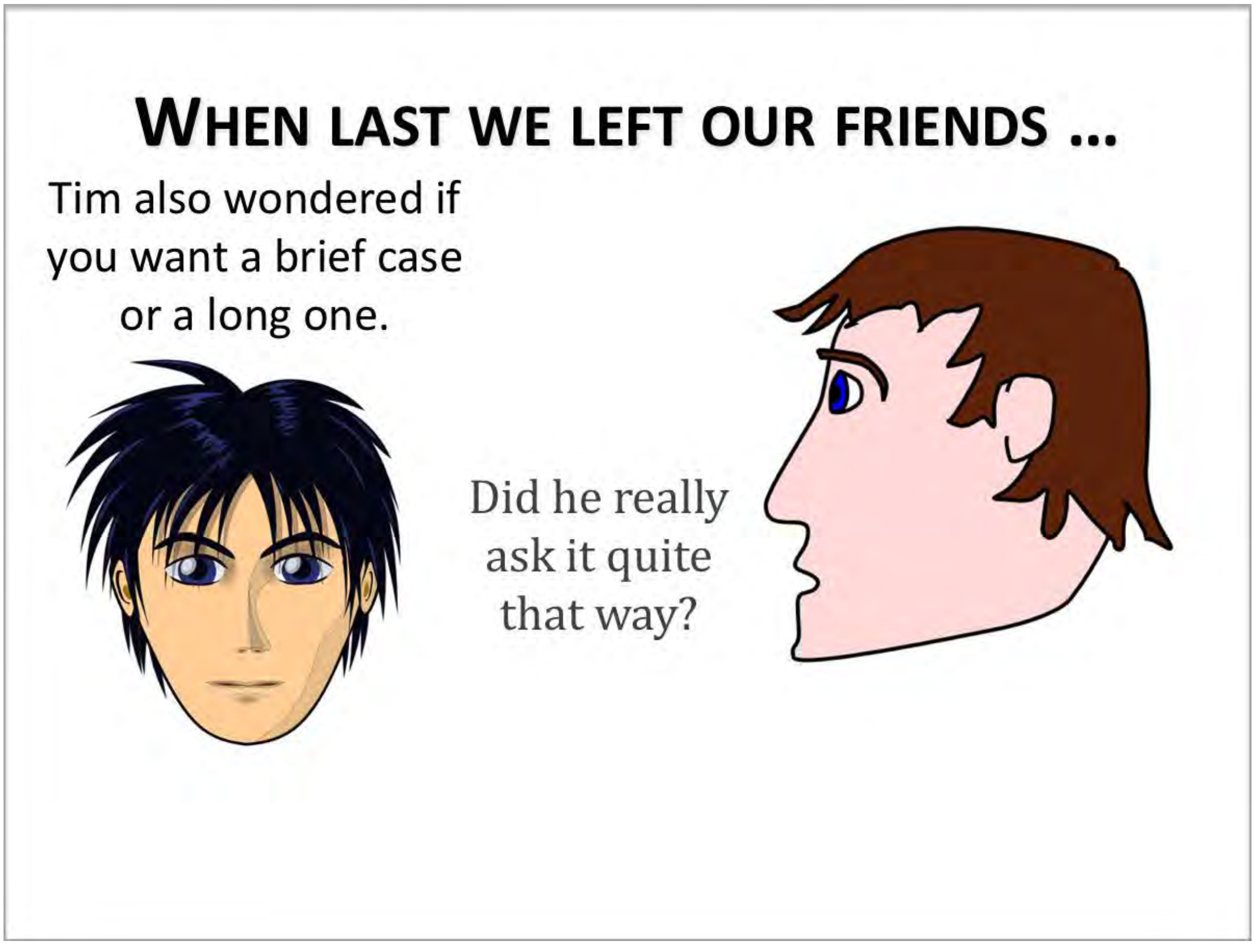
[Question to participants: Any questions about these learning objectives?]

[1] Just in case someone does not follow the suggestion, and thus misses the preliminary information first expounded in Module 1 and repeated verbatim in Module 2, here is that information in simplified form: Within the assurance case community, intramural debates abound about a variety of topics we will discuss. Except in rare instances the existence of these debates is intentionally ignored or mentioned only briefly in this material. (See Module 1 or 2 for an explanation of why). Also, all images you see were either created by me (Michael Holloway) or are in the public domain via CC0 1.0 Universal.

When last we left our friends Jon (the teenager on the left), his dad Mike (the fellow on the right), and Tim (the unseen fellow who may or may not drive Jon to a game) Jon had told his dad that …

“Tim also wondered if you want a brief case or a long one.”

We left Jon’s dad smiling, but we know pick up the conversation a few seconds later.



WHEN LAST WE LEFT OUR FRIENDS ...

Tim also wondered if you want a brief case or a long one.
Did he really ask it quite that way?

Mike asks Jon, “Did he really ask it quite that way?”

“Well, no, not exactly,” says Jon, “He’s not quite as funny as I am.”

“So, what did he really want to know?” asks Mike.

“He wanted more details about what you’re expecting,” replies Jon.

“That’s simple,” says Jon’s dad.

“I want a cogent argument.”

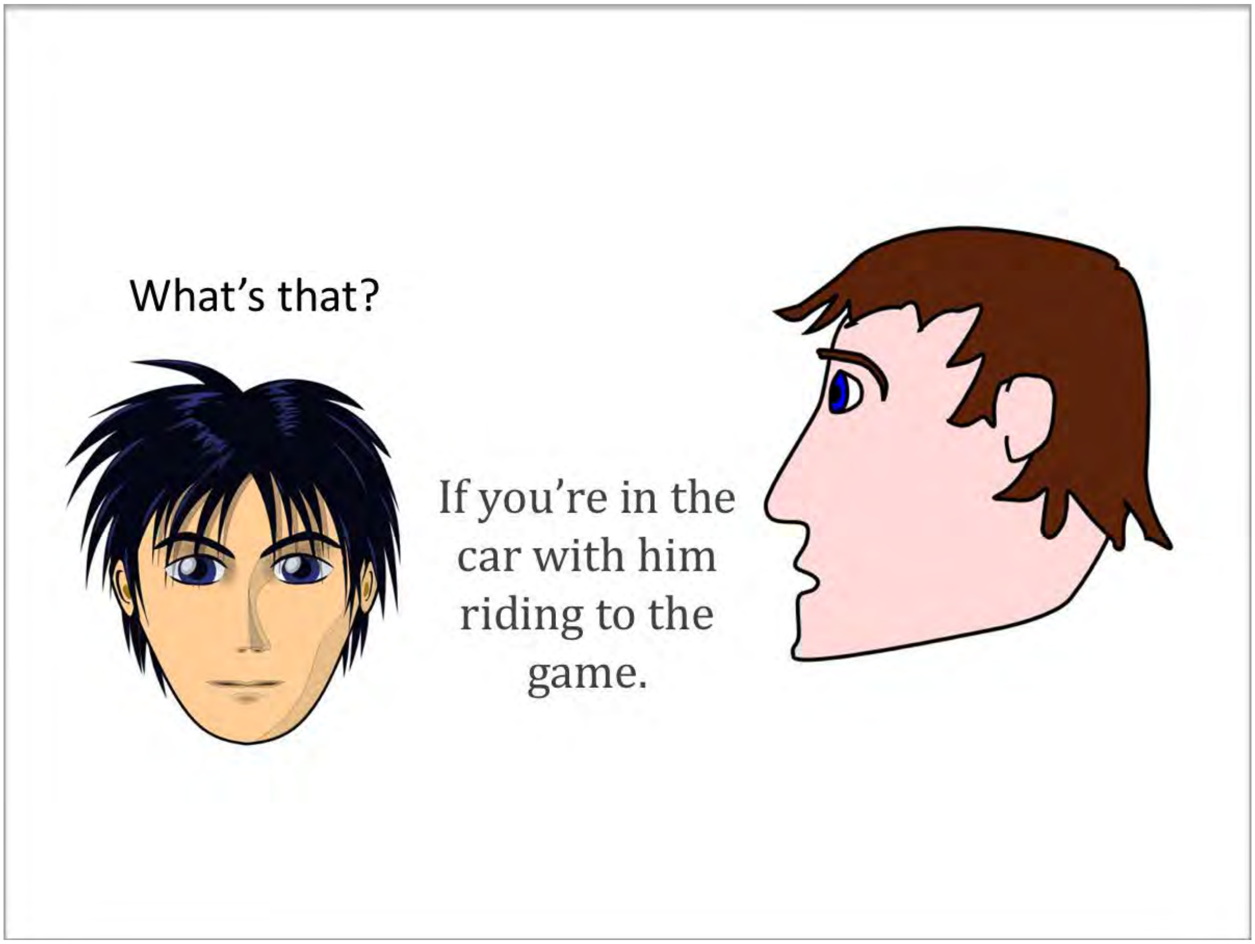
Jon is not thrilled by that answer, and exclaims,

“Simple? … How will Tim know what you’ll think is cogent?”

"Well … there's one surefire way he'll know …"

"What's that?" asks Jon.

"If you're in the car with him riding to the game."

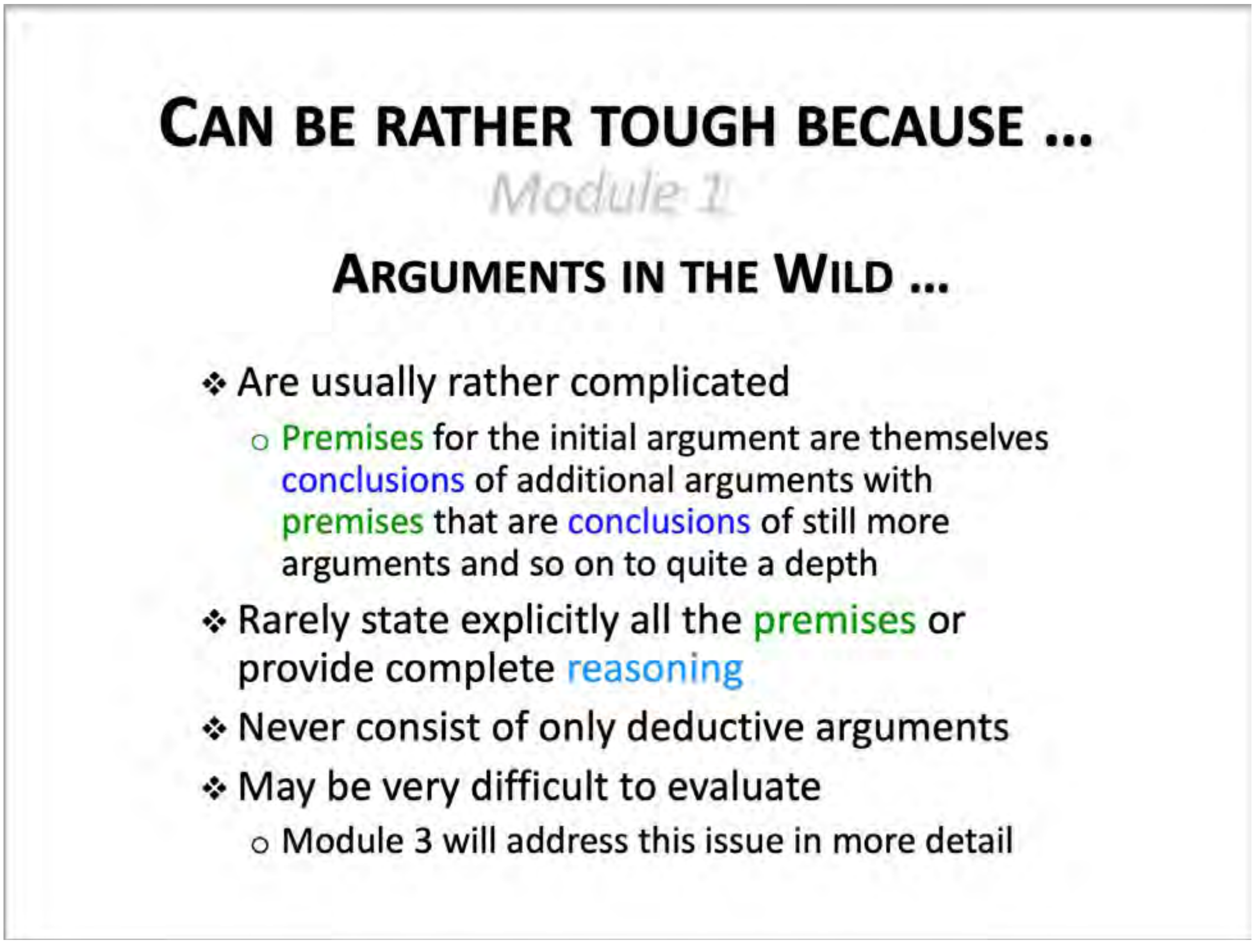


With a sigh, Jon replies, "Cute Dad … but that's a bit late to find out, don't ya think?"

"Yeah, sorry … there really isn't a simple answer. Deciding if a case is good enough can be rather tough."

Mike is spot on: evaluating an assurance case can be rather tough, whether you're a writer evaluating your own case, an auditor evaluating someone else's case, or just an inquisitive learner wondering about the matter.

It can be rather tough for a variety of reasons, beginning with some of the observations we made in Module 1 concerning the nature of arguments in the wild.



CAN BE RATHER TOUGH BECAUSE ...

Module 1

ARGUMENTS IN THE WILD ...

- Are usually rather complicated
  - Premises for the initial argument are themselves conclusions of additional arguments with premises that are conclusions of still more arguments and so on to quite a depth
- Rarely state explicitly all the premises or provide complete reasoning
- Never consist of only deductive arguments
- May be very difficult to evaluate
  - Module 3 will address this issue in more detail

Recall then that we said that real arguments are usually rather complicated. We noted in particular, the premises for the initial argument are themselves usually conclusions of additional arguments with premises that are conclusions of still more arguments and so on to quite a depth.

In any real assurance case, the *premises* for the top level *conclusion* will almost certainly not be obvious truths, but rather statements that will need to be supported by argument themselves. Eventually the assurance case should stop with sub-arguments with *premises* whose truth can be agreed upon by all relevant parties; such premises are sometimes called evidence, 'though, as I've mentioned, I am not particularly fond of that term.

Second, real arguments rarely state explicitly all of the premises or provide complete reasoning. This should be less true of assurance case arguments than is generally true of generic arguments in the wild, but deciding whether it's true is one of the evaluation activities, and it is not necessarily an easy one.

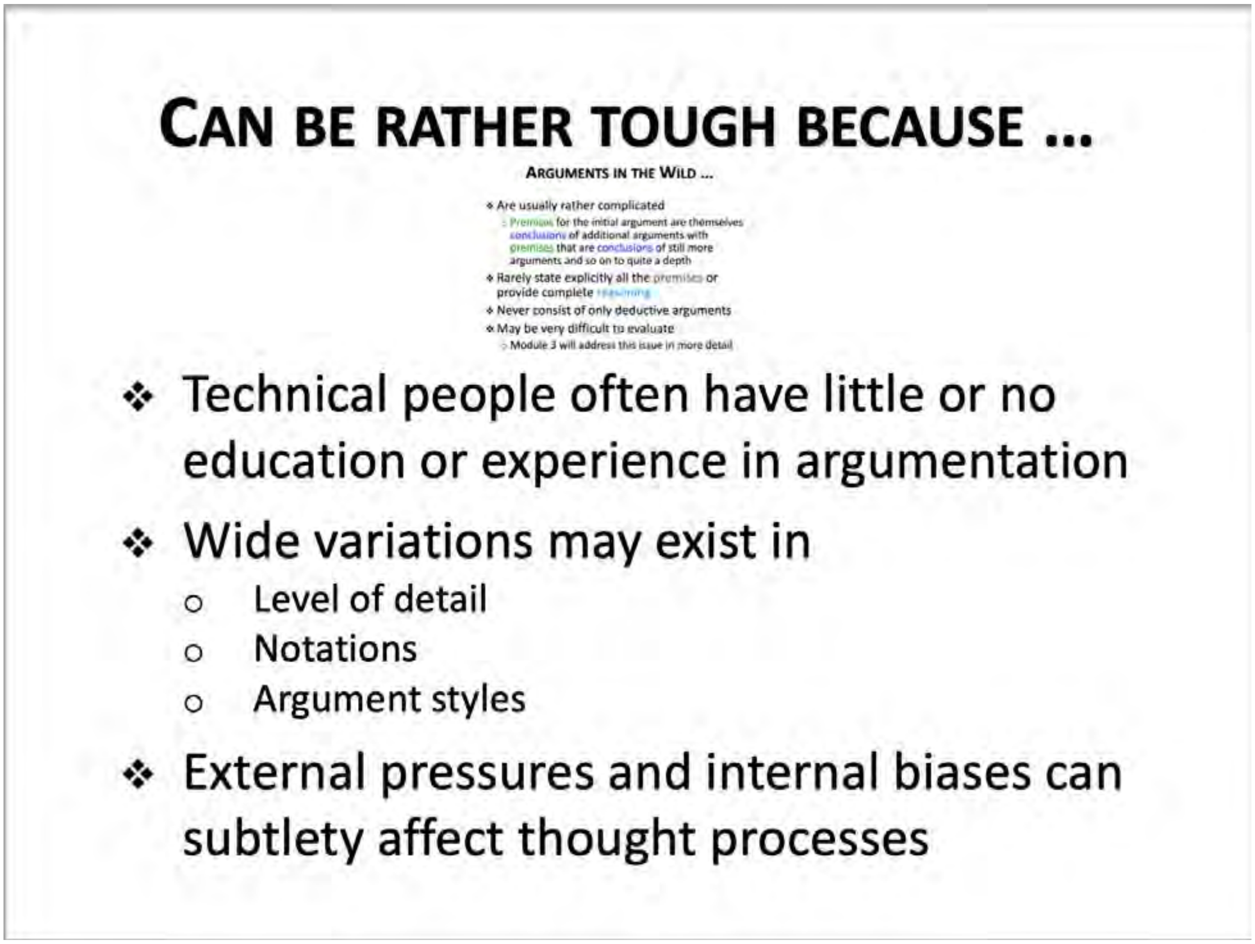
Third, real arguments, both in the generic wild, and in the assurance case context, almost never consist of only deductive arguments.

You'll recall from Module 1 (or from prior knowledge) that deductive arguments are ones in which true premises and valid reasoning *guarantee* the truth of the conclusion.

Inductive arguments, on the other hand, do not provide guarantees, only increases in confidence. An inductive argument with true premises and strong reasoning should improve our confidence in the truth of the conclusion, but ought not provide us with certainty.

We talked a bit in Module 1 about the controversy that exists within the assurance case community over whether there may be advantages to be gained from making deductive as many arguments as possible; or perhaps by using a normalized structure that isolates inductive arguments into specific parts of the overall argument. That controversy is currently an academic one, because everyone, even the most zealous formalist, recognizes that the current state of the practice involves mostly inductive arguments.

These three facts aren't the only things that can make it rather tough to evaluate an assurance case.



CAN BE RATHER TOUGH BECAUSE ...

ARGUMENTS IN THE WILD ...

- Are usually rather complicated
  - Premises for the initial argument are themselves conclusions of additional arguments with premises that are conclusions of still more arguments and so on to quite a depth
- Rarely state explicitly all the premises or provide complete reasoning
- Never consist of only deductive arguments
- May be very difficult to evaluate
  - Module 3 will address this issue in more detail

- Technical people often have little or no education or experience in argumentation
- Wide variations may exist in
  - Level of detail
  - Notations
  - Argument styles
- External pressures and internal biases can subtlety affect thought processes

Other toughness inducing-aspects include the things you see here. Technical people often have little or no education or experience in argumentation. This lack of knowledge and practiced ability can lead to poorly written assurance cases, and perhaps to an inability by auditors to recognize them as such.

You may recall from Module 2 the poor quality of some safety cases was identified in several accident inquiries including Ladbroke Grove and Nimrod, and has been identified by the FDA as a problem they are experiencing as they use assurance cases in infusion pump approvals.

Evaluating assurance cases can be tough also because in practice cases may vary widely in the level of detail provided (some cases may be really just argument sketches, while others may delve deeply into the tiniest details of a system).

They may also differ widely in the notations used, ranging (as we saw in Module 1) from unstructured prose to highly structured, but not necessarily easy to understand, graphical notations.

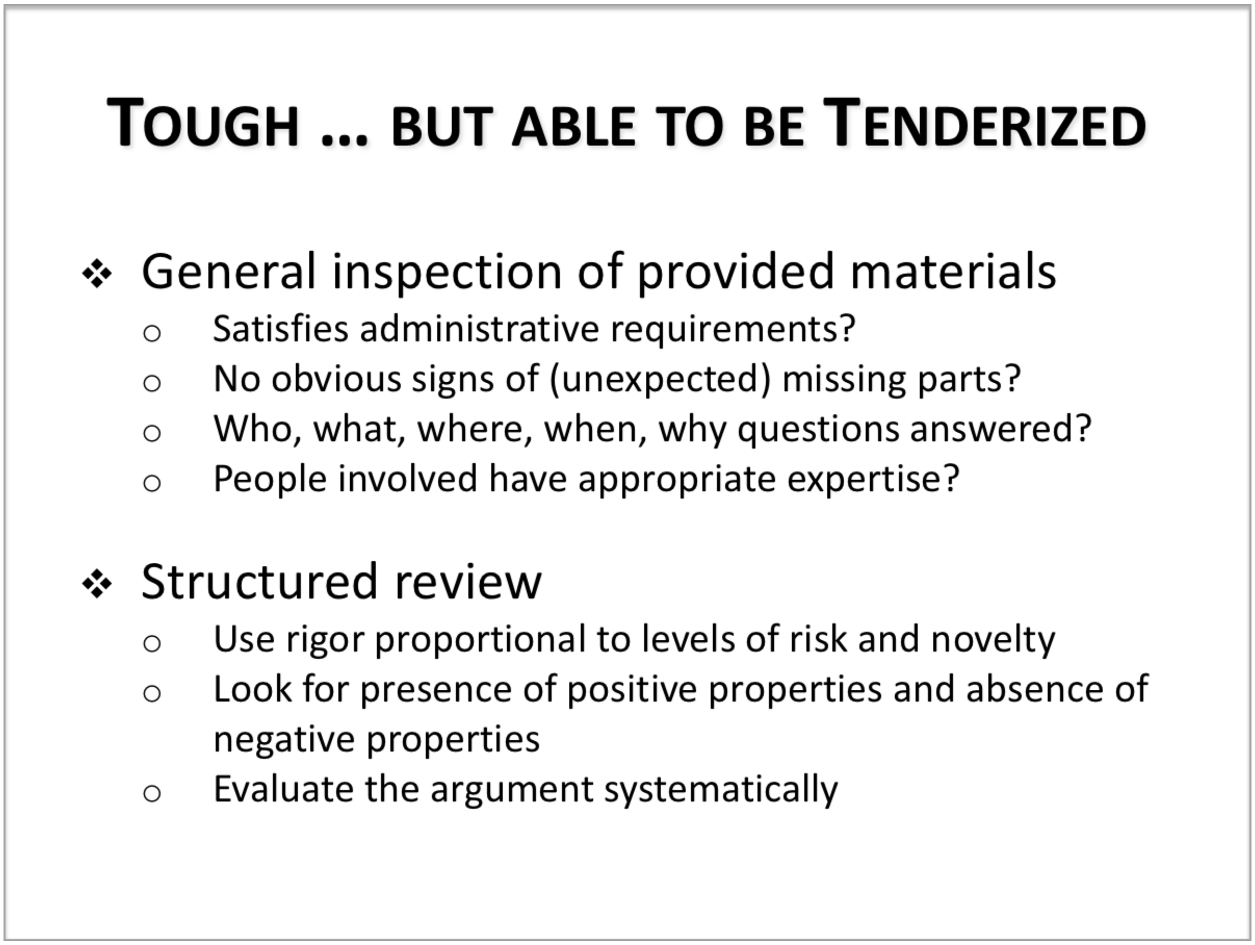
If you're asked to evaluate an assurance case in a notation you don't already know, you may find it quite hard to distinguish between problems in the assurance case itself and problems in your own understanding of the notation.

There can also be wide variations in argument styles, which can make consistent evaluation hard.

Finally, evaluating an assurance case can be made tough by external pressures and internal biases that can affect your thought processes, even if you try to block out the effects. We'll talk some more about these things a bit later on. All these things, and probably others we've not discussed, make evaluating assurance cases tough.

[Question to participants: Before I talk a bit about how this toughness may be tenderized, does anyone have a question they'd like to ask now?]

Evaluating assurances is tough, but it can be tenderized in some very helpful ways.



TOUGH … BUT ABLE TO BE TENDERIZED

- ❖ General inspection of provided materials
  - ○ Satisfies administrative requirements?
  - ○ No obvious signs of (unexpected) missing parts?
  - ○ Who, what, where, when, why questions answered?
  - ○ People involved have appropriate expertise?
- ❖ Structured review
  - ○ Use rigor proportional to levels of risk and novelty
  - ○ Look for presence of positive properties and absence of negative properties
  - ○ Evaluate the argument systematically

First, there are various steps that can be taken by way of the general inspection of provided materials.

Before starting evaluation of the assurance argument, you should look everything over to see if (first) it satisfies administrative requirements. For example, if the argument is required to be expressed in a particular notation or style, is it?

You should be sure that there are no obvious signs of (unexpected) missing parts. Does the argument have a top level conclusion, for example.

As we discussed in Module 2, answers to the who, what, where, when, why questions are important. Does the case make clear who wrote it, what its scope is, and what assurance target is applicable, for example?

Finally, by way of general inspection, does the information you have available show that the people involved in designing the system or service, have appropriate expertise?

If an assurance case that you've been asked to evaluate does not pass even a general inspection, there is no good reason to attempt a more extensive, structured review.

We'll talk about the structured review in much more detail shortly, but here on the slide are three important aspects of it.

First, the rigor of the review should likely be tailored to the levels of risk and of novelty in the system or service for which the assurance case has been developed. Generally, the greater the risk the more rigorous the review should be, and the greater the novelty of the system, the more rigorous the review should be.

Second, you should be continually looking for presence of positive properties and absence of negative properties (both of which we'll talk about a bit more shortly).

This looking for properties will be going on while you evaluate the argument systematically. We'll go through a procedure for this systematic review shortly.

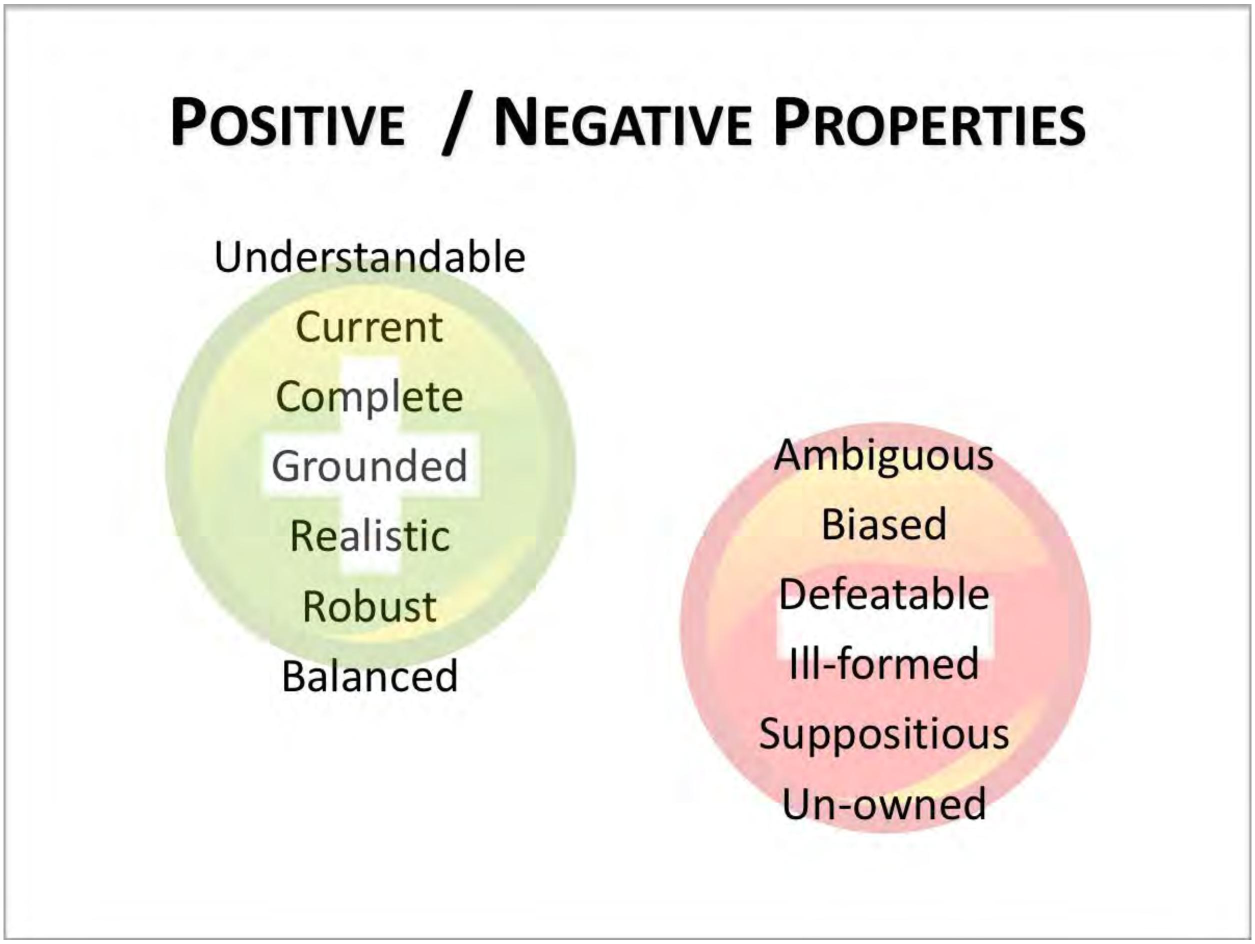
Now, I want to enumerate briefly some positive properties and some negative properties that an assurance case may possess.

The next slide lists seven positive properties and six negative properties; the meaning of some of these is probably self-evident; while the meaning of some others … not so much.

Understandable is pretty self-evident: the assurance case needs to provide enough information, in a clear way, so that everyone who will use it knows what it means, and to what it applies.

Current means that the case accurately represents the current status of the system or service in all relevant aspects.

Complete is a relative term, which depends on the life cycle stage(s) covered by the case, but relative to that stage, the assurance case should cover in an appropriate way all aspects of the system or service.



Positive / Negative Properties

Understandable
Current
Complete
Grounded
Realistic
Robust
Balanced

Ambiguous
Biased
Defeatable
Ill-formed
Suppositious
Un-owned

By *Grounded* I am referring back to the concept we introduced in Module 1, namely that the argument structure terminates in premises whose 'truth' can be agreed by all relevant parties.

Realistic means that the case identifies its assumptions and that these assumptions correspond well to what will happen (or is happening) in the actual world.

Robust refers to an assurance case incorporating good engineering practice and known sound safety principles.

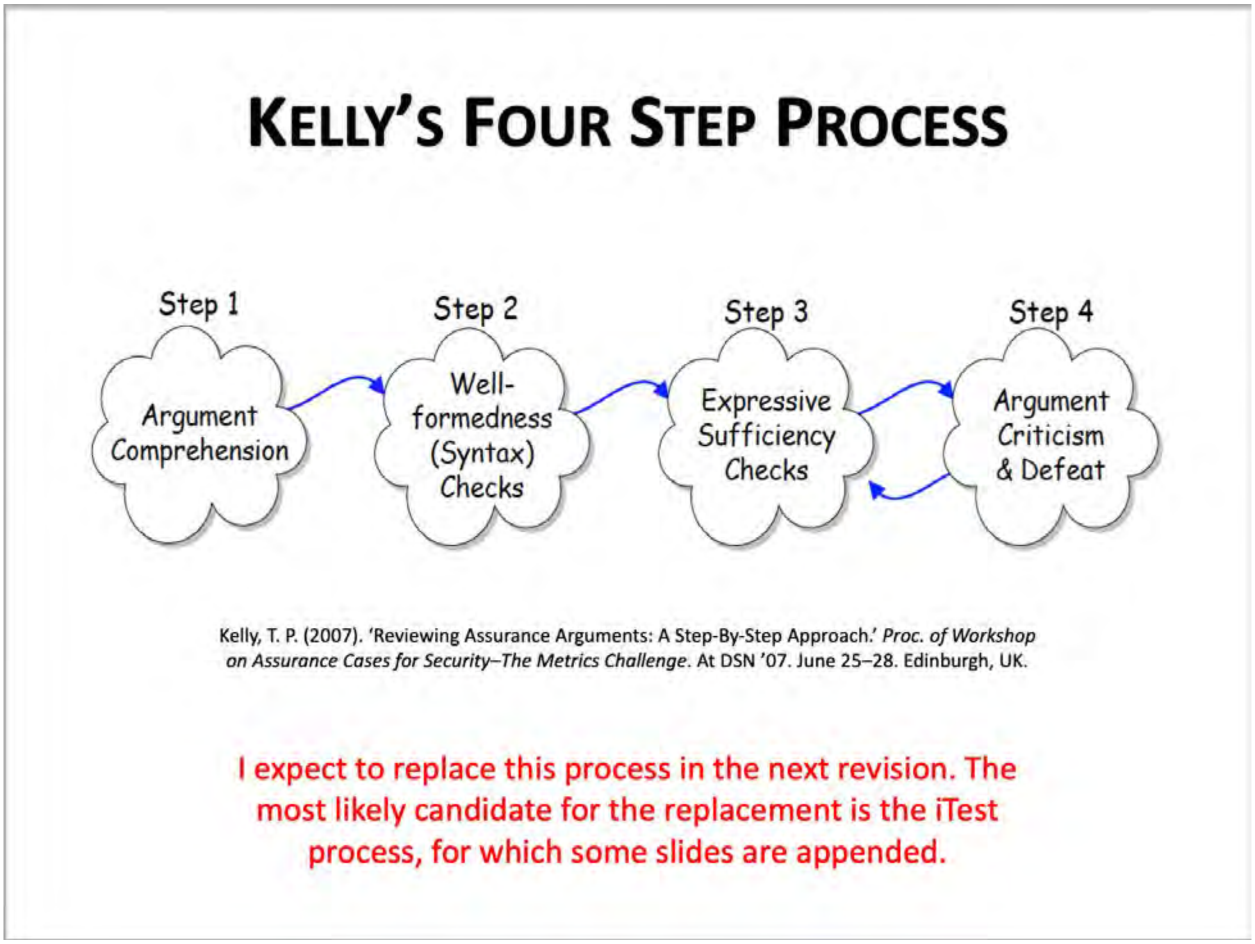
Finally, the positive property *Balanced* refers to the assurance case identifying not only the strengths of the system or service but also its known weaknesses.

Those are seven positive properties that a good assurance case should possess.

You see also 6 negative properties that a good assurance case should not possess, but which a bad one probably will: ambiguous, biased, defeatable, ill-formed, suppositious, and un-owned. I'm going to defer talking about what these mean until after we've gone through a process for systematic evaluation of the assurance case argument.

[Question to participants: Before we proceed, does anyone have a question?]

I'm going to present one particular way to undertake a systematic evaluation. There are many other ways, 'though all of them will necessarily include similar sorts of things as the process that I'll show you. [By mid 2021 this approach will be replaced.]



Kelly's Four Step Process

Step 1

Argument Comprehension

Step 2

Well-formedness (Syntax) Checks

Step 3

Expressive Sufficiency Checks

Step 4

Argument Criticism & Defeat

Kelly, T. P. (2007). 'Reviewing Assurance Arguments: A Step-By-Step Approach.' *Proc. of Workshop on Assurance Cases for Security–The Metrics Challenge*. At DSN '07. June 25–28. Edinburgh, UK.

I expect to replace this process in the next revision. The most likely candidate for the replacement is the iTest process, for which some slides are appended.

The four step process you see here was developed by Tim Kelly at the University of York, and published in a DSN-affiliated workshop proceedings in 2007.

[Kelly, T. P. (2007). "Reviewing Assurance Arguments: A Step-By-Step Approach." *Proc. of Workshop on Assurance Cases for Security---The Metrics Challenge*. At DSN '07, June 25-28. Edinburgh.]

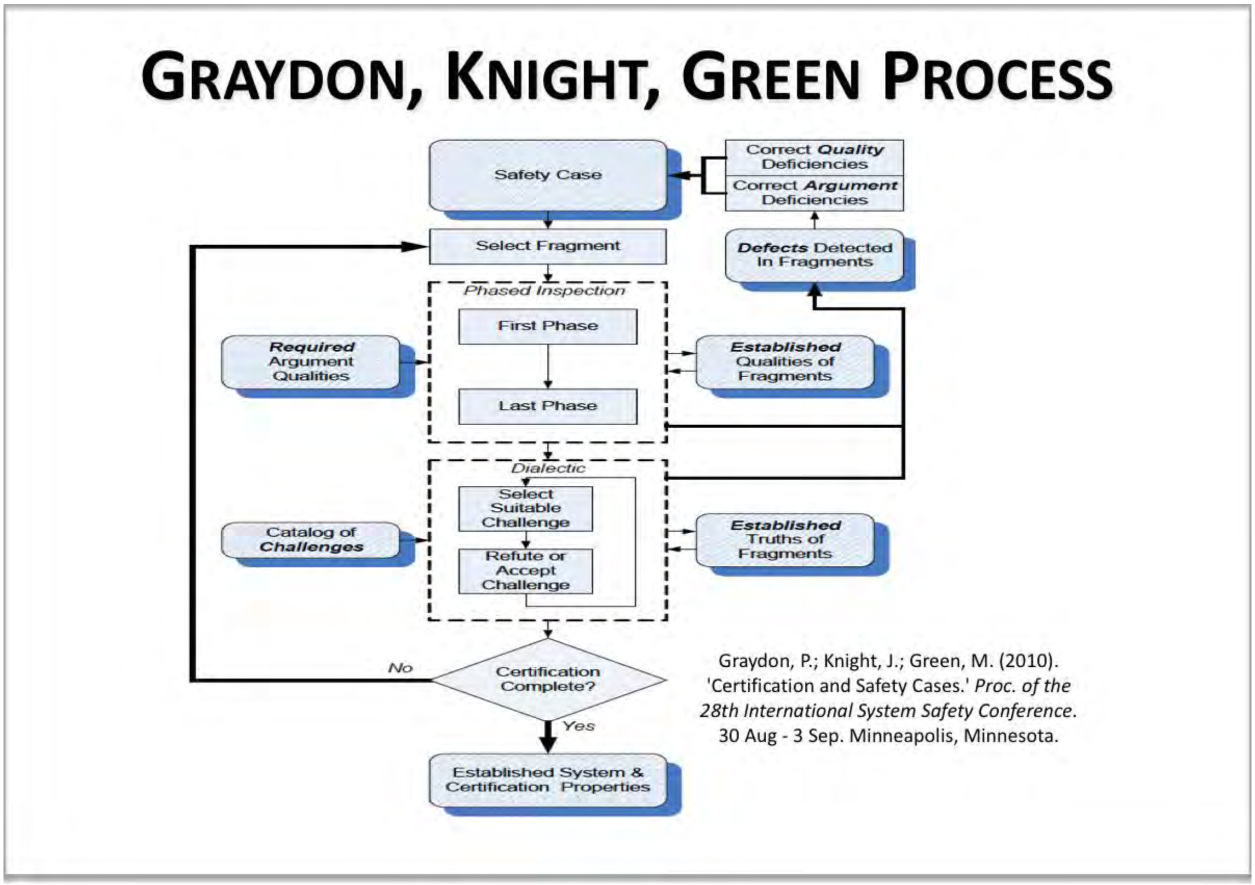
Argument evaluation starts with argument comprehension then proceeds to checking for well-formedness, followed by checking for expressive sufficiency, and concluding with argument criticism (and possible) defeat, which may lead to changes in the argument necessitating repeating step 3, followed by step 4.

We'll look at each of these steps in more detail shortly.

Before doing so, I want to also mention another argument evaluation process from which I've borrowed some parts.

Patrick Graydon, John Knight, and Mitchell Green, who were all at the University of Virginia at the time, published this process at the International System Safety Conference in 2010.

I'm not going to go into any detail, but just want to note that I'll be incorporating some of the ideas from the GKG approach into my elaboration of the Four Step Process.



Graydon, Knight, Green Process

Safety Case

Correct *Quality* Deficiencies

Correct *Argument* Deficiencies

Select Fragment

Defects Detected In Fragments

Phased Inspection

First Phase

Last Phase

Required Argument Qualities

Established Qualities of Fragments

Dialectic

Select Suitable Challenge

Refute or Accept Challenge

Catalog of *Challenges*

Established Truths of Fragments

No

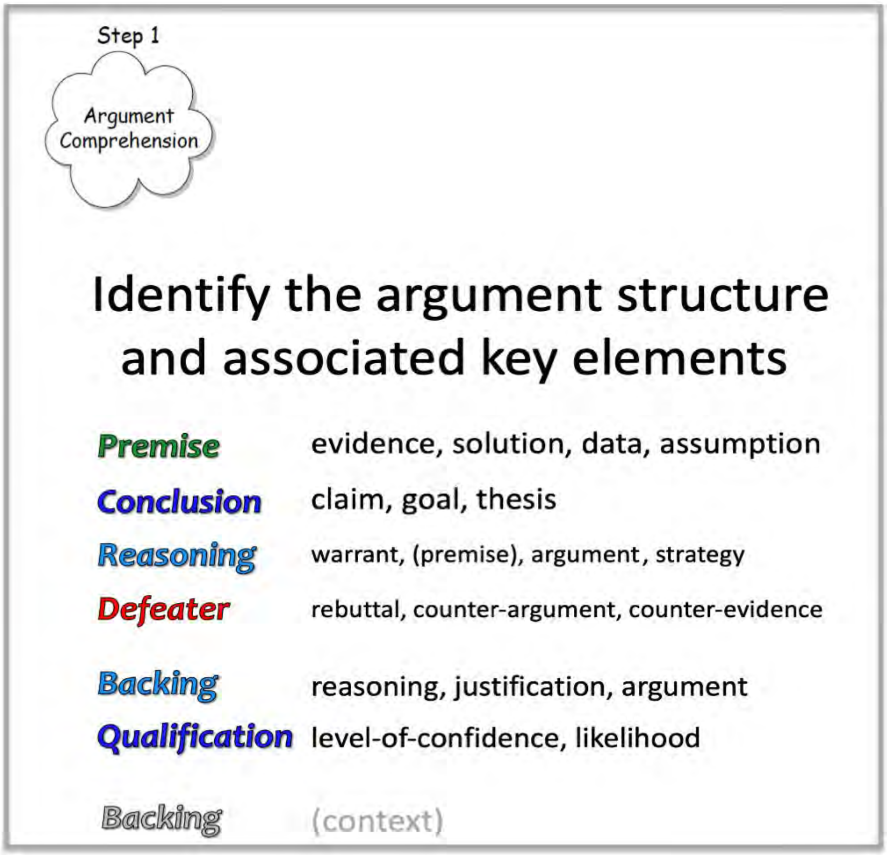
Certification Complete?

Yes

Established System & Certification Properties

Graydon, P.; Knight, J.; Green, M. (2010). 'Certification and Safety Cases.' *Proc. of the 28th International System Safety Conference*. 30 Aug - 3 Sep. Minneapolis, Minnesota.

So let’s look at each of the four steps in Kelly’s process in turn. Step 1 is understanding the argument.



Step 1

Argument Comprehension

Identify the argument structure and associated key elements

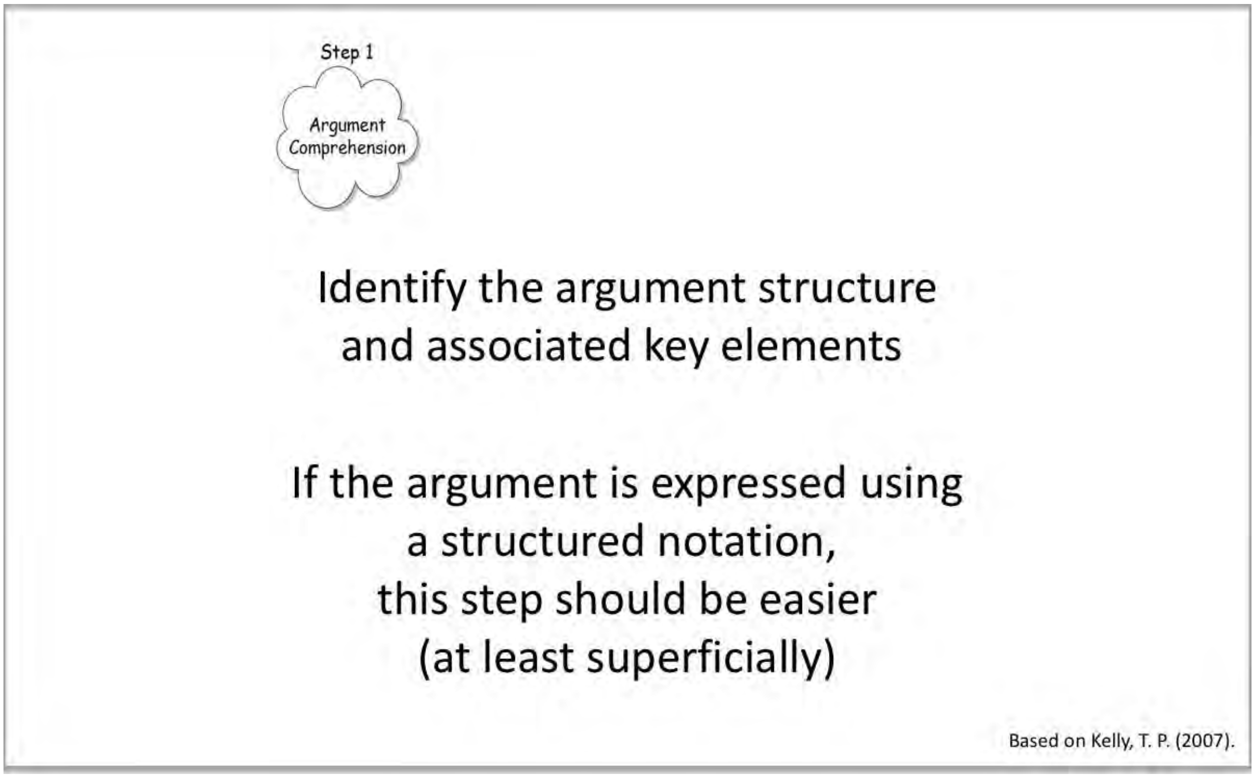
Premise	evidence, solution, data, assumption
Conclusion	claim, goal, thesis
Reasoning	warrant, (premise), argument, strategy
Defeater	rebuttal, counter-argument, counter-evidence
Backing	reasoning, justification, argument
Qualification	level-of-confidence, likelihood
Backing	(context)

Or to be more precise, identifying the argument structure and associated key elements.

As I hope you recall from Module 1, those key elements include *premise*, *conclusion*, *reasoning*, (all three of which are necessarily present) and the other elements (which may or may not be present) *qualification*, *defeater, backing,* and *binding*.

As we did in Module 1, other common names for these concepts are listed, too.

Kelly notes in his paper that if the argument is expressed using a structured notation, this step should be easier. I've added the qualification "at least superficially", because using a structured notation doesn't really guarantee that a comprehensible argument will be created. Certainly, if the argument is expressed in an entirely unstructured way using regular prose, re-representing it in some structured way, (which doesn't have to be graphical) can be a wise thing to do at this stage.



Step 1

Argument Comprehension

Identify the argument structure and associated key elements

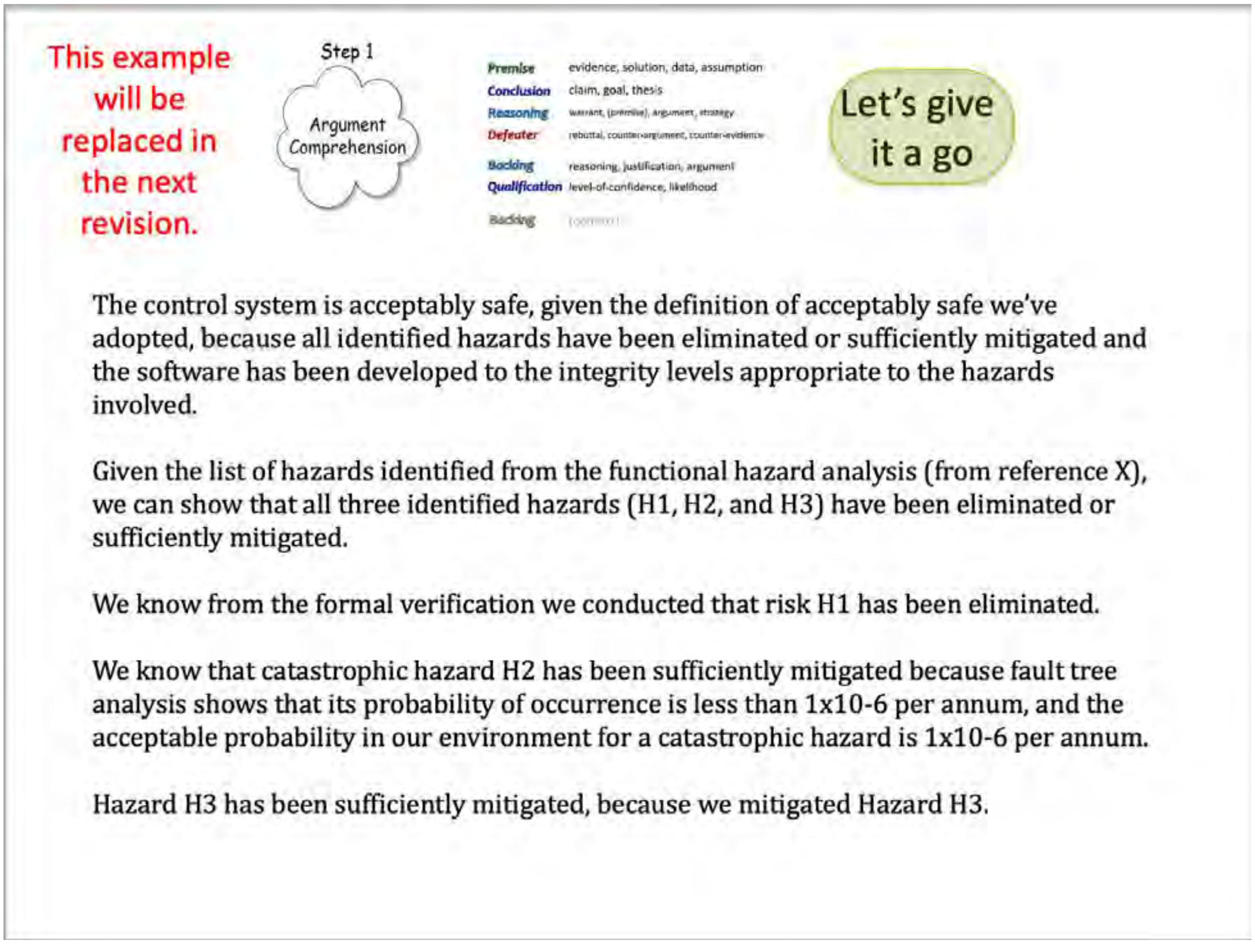
If the argument is expressed using a structured notation, this step should be easier (at least superficially)

Based on Kelly, T. P. (2007).

Now I'm going to show you a short assurance case, written in natural language. The example is based on the example I used in the 2008 notations paper mentioned in Module 1 [Holloway, C. M. 2008. "Safety Case Notations: Alternatives for the Non-Graphically Inclined?" IET 3nd International Conference on System Safety. 21-23 October 2008, Birmingham, UK. Available at `http://hdl.handle.net/2060/20080042416`.]

What you'll see is not identical to the case presented in the paper, but it is very similar.

Here it is. (In the planned revision, this example will be replaced.)



This example will be replaced in the next revision.

Step 1

Argument Comprehension

Premise	evidence, solution, data, assumption
Conclusion	claim, goal, thesis
Reasoning	warrant, (premise), argument, strategy
Defeater	rebuttal, counter-argument, counter-evidence
Backing	reasoning, justification, argument
Qualification	level-of-confidence, likelihood
Backing	(context)

Let's give it a go

The control system is acceptably safe, given the definition of acceptably safe we've adopted, because all identified hazards have been eliminated or sufficiently mitigated and the software has been developed to the integrity levels appropriate to the hazards involved.

Given the list of hazards identified from the functional hazard analysis (from reference X), we can show that all three identified hazards (H1, H2, and H3) have been eliminated or sufficiently mitigated.

We know from the formal verification we conducted that risk H1 has been eliminated.

We know that catastrophic hazard H2 has been sufficiently mitigated because fault tree analysis shows that its probability of occurrence is less than 1x10-6 per annum, and the acceptable probability in our environment for a catastrophic hazard is 1x10-6 per annum.

Hazard H3 has been sufficiently mitigated, because we mitigated Hazard H3.

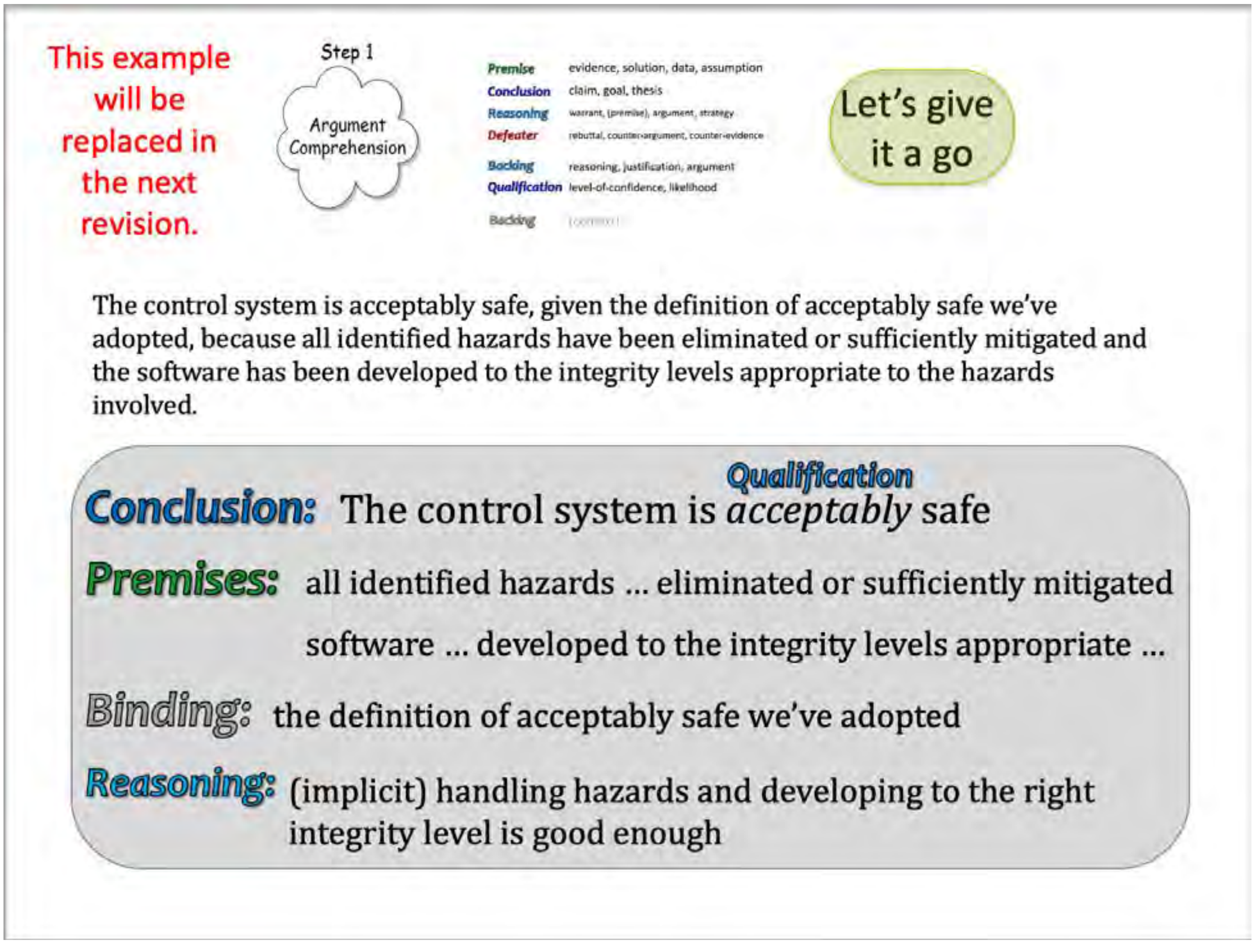
What I want you to do is to identify the top level conclusion and the premises and reasoning upon which it rests, along with any qualifications or bindings that are associated with them.

Don't try to do anything more than that. Remember that these top-level premises may well serve as conclusions for lower-level arguments. Do not worry about the lower-level arguments.

As two very big hints ... you don't have to read very much of the text, and it is entirely possible that an important element may be implicit.

Please do not turn the page until you have attempted the exercise.

Here’s my answer.



This example will be replaced in the next revision.

Step 1

Argument Comprehension

Premise	evidence, solution, data, assumption
Conclusion	claim, goal, thesis
Reasoning	warrant, (premise), argument, strategy
Defeater	rebuttal, counter-argument, counter-evidence
Backing	reasoning, justification, argument
Qualification	level-of-confidence, likelihood
Backing	[context]

Let’s give it a go

The control system is acceptably safe, given the definition of acceptably safe we’ve adopted, because all identified hazards have been eliminated or sufficiently mitigated and the software has been developed to the integrity levels appropriate to the hazards involved.

Qualification

Conclusion: The control system is *acceptably* safe

Premises: all identified hazards ... eliminated or sufficiently mitigated

software ... developed to the integrity levels appropriate ...

Binding: the definition of acceptably safe we’ve adopted

Reasoning: (implicit) handling hazards and developing to the right integrity level is good enough

The conclusion is “The control system is … safe” with the qualification of “acceptably”.

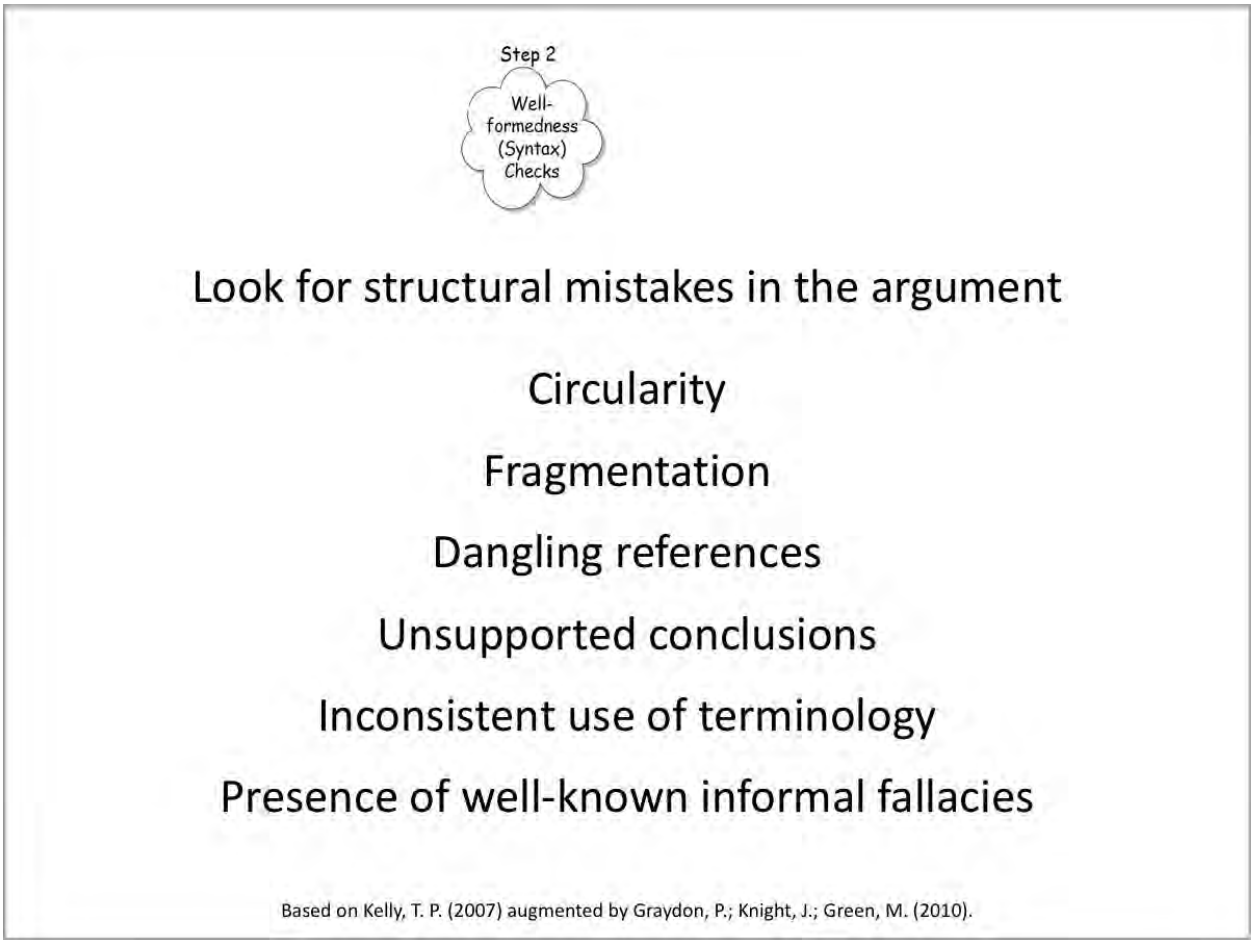
The two premises are “All identified hazards have been eliminated or sufficiently mitigated” and “The software has been developed to the integrity levels appropriate to the hazards involved.”

A definition of “acceptably safe” needs to identified in a binding, and the reasoning seems to be implicit, something along the lines of “handling hazards and developing to the right integrity level is good enough.”

That’s the argument comprehension step.

[Question to participants: Any questions?]

Step two is called “Well-formedness (Syntax) Checks.” It involves looking for structural mistakes in the argument.



Step 2

Well-formedness (Syntax) Checks

Look for structural mistakes in the argument

Circularity

Fragmentation

Dangling references

Unsupported conclusions

Inconsistent use of terminology

Presence of well-known informal fallacies

Based on Kelly, T. P. (2007) augmented by Graydon, P.; Knight, J.; Green, M. (2010).

Among the structural mistakes that might exist are six that you see listed here.

Circularity refers to an argument that has as a premise a statement that is equivalent to its conclusion. This could happen directly, ‘though it is more likely to happen indirectly, where the conclusion at (for example level `n`) in an argument reappears in some form as a premise in (for example) level `n+3`.

Fragmentation refers to arguments that are disconnected from the main argument.

A *dangling reference* is a reference in the argument to something that doesn’t exist (or at least isn’t present within the assurance case materials available to the evaluator).

Unsupported conclusions are conclusions for which no argument is given. In effect they are treated as premises (something about which everyone can agree), but their truth remains in doubt.

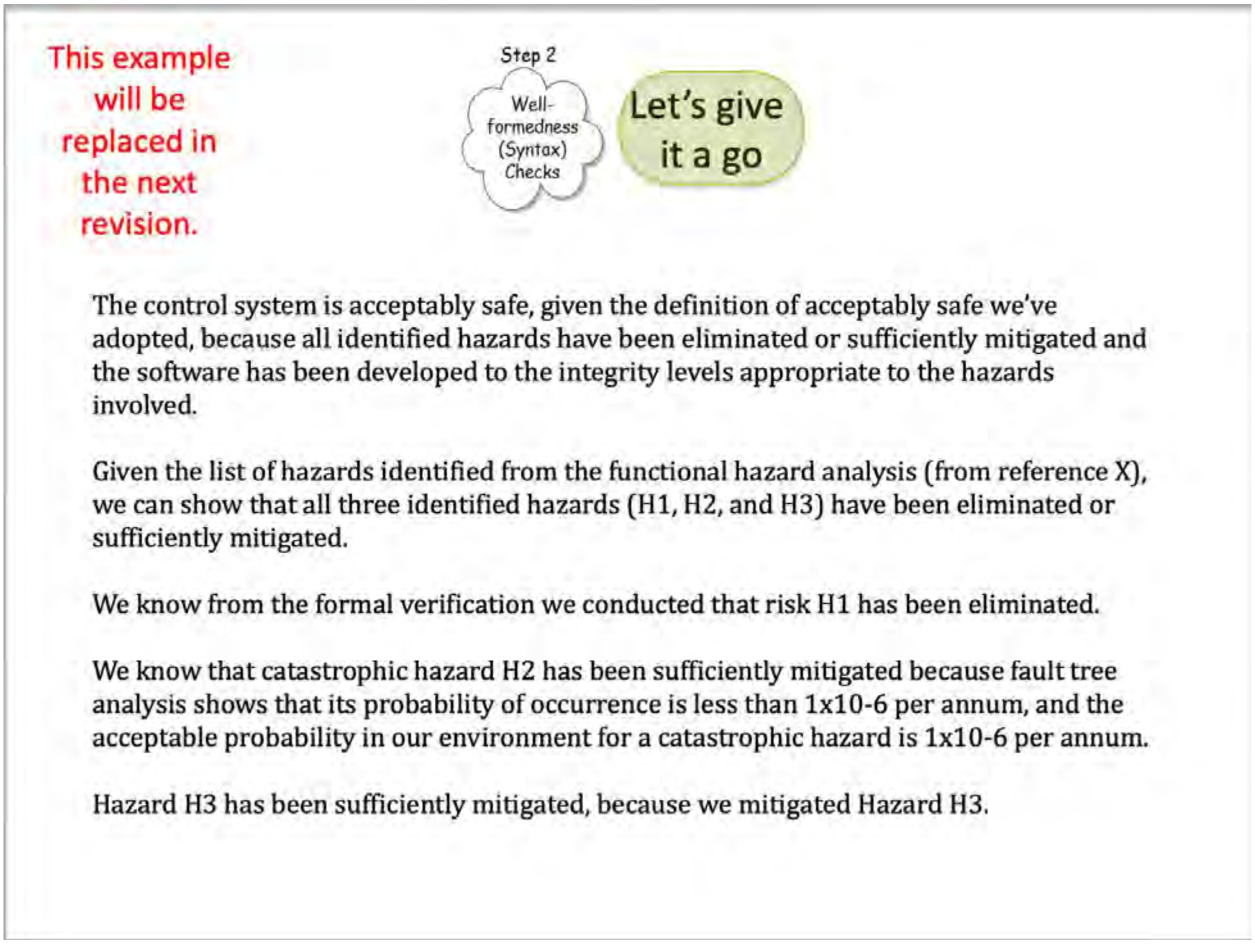
Inconsistent use of terminology is self-evident, I think.

Finally, the *presence of well-known informal fallacies* refers to using arguments that are known to be inadequate.

One example is known as the fallacious composition, which occurs when an argument claims that, because a property holds over the parts of a system or service, it therefore holds for the larger entity, without considering possible interactions between parts or external influences.

A prototypical example of fallacious composition within a safety case is an argument that claims that a whole system is safe solely because its subsystems A, B, and C are safe, while failing to consider the effect on safety of interactions among the subsystems.
Any questions before you get to try to identify some structural mistakes?

See if you can find some structural mistakes in the argument we just looked at for step 1. There are at least three.



This example will be replaced in the next revision.

Step 2

Well-formedness (Syntax) Checks

Let's give it a go

The control system is acceptably safe, given the definition of acceptably safe we've adopted, because all identified hazards have been eliminated or sufficiently mitigated and the software has been developed to the integrity levels appropriate to the hazards involved.

Given the list of hazards identified from the functional hazard analysis (from reference X), we can show that all three identified hazards (H1, H2, and H3) have been eliminated or sufficiently mitigated.

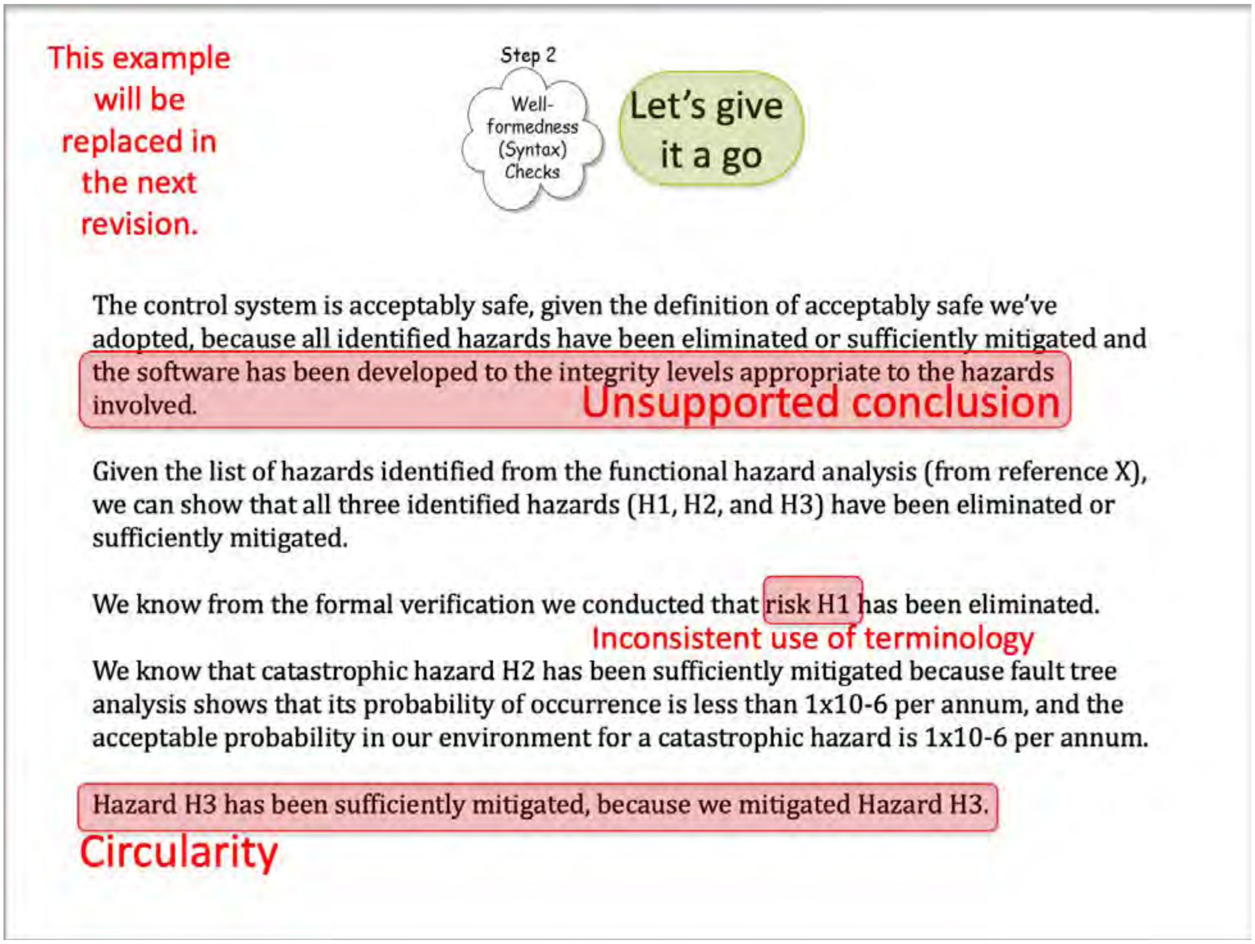
We know from the formal verification we conducted that risk H1 has been eliminated.

We know that catastrophic hazard H2 has been sufficiently mitigated because fault tree analysis shows that its probability of occurrence is less than 1x10-6 per annum, and the acceptable probability in our environment for a catastrophic hazard is 1x10-6 per annum.

Hazard H3 has been sufficiently mitigated, because we mitigated Hazard H3.

Please do not turn the page until you have attempted the exercise.

Here’s my answer.



This example will be replaced in the next revision.

Step 2

Well-formedness (Syntax) Checks

Let’s give it a go

The control system is acceptably safe, given the definition of acceptably safe we’ve adopted, because all identified hazards have been eliminated or sufficiently mitigated and the software has been developed to the integrity levels appropriate to the hazards involved.

Unsupported conclusion

Given the list of hazards identified from the functional hazard analysis (from reference X), we can show that all three identified hazards (H1, H2, and H3) have been eliminated or sufficiently mitigated.

We know from the formal verification we conducted that risk H1 has been eliminated.

Inconsistent use of terminology

We know that catastrophic hazard H2 has been sufficiently mitigated because fault tree analysis shows that its probability of occurrence is less than 1x10-6 per annum, and the acceptable probability in our environment for a catastrophic hazard is 1x10-6 per annum.

Hazard H3 has been sufficiently mitigated, because we mitigated Hazard H3.

Circularity

One structural mistake is that the top-level premise concerning software being developed to appropriate integrity levels is not supported by any argument, hence it is an unsupported (lower-level) conclusion.

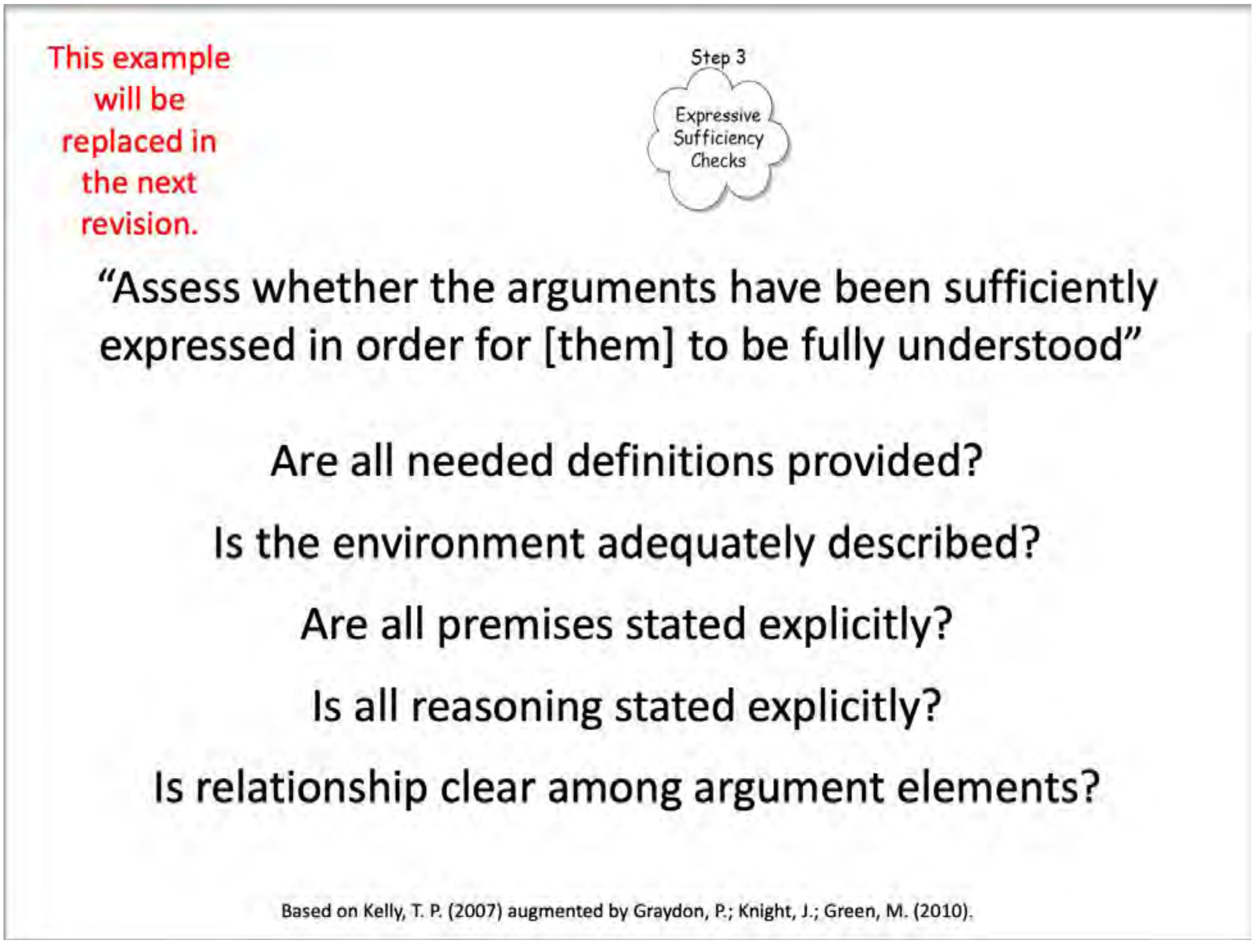
Another structural mistake is the use of the word ‘risk’ in relation to H1 in the third paragraph, which is inconsistent terminology as ‘hazard’ is used elsewhere.

Finally, the argument concerning hazard H3 is blatantly circular.

That’s step 2.

[Note to participants: Any questions?]

Step 3 involves assessing whether the arguments have been sufficiently expressed in order for them to be fully understood.



This example will be replaced in the next revision.

Step 3

Expressive Sufficiency Checks

"Assess whether the arguments have been sufficiently expressed in order for [them] to be fully understood"

Are all needed definitions provided?

Is the environment adequately described?

Are all premises stated explicitly?

Is all reasoning stated explicitly?

Is relationship clear among argument elements?

Based on Kelly, T. P. (2007) augmented by Graydon, P.; Knight, J.; Green, M. (2010).

Specifically, answering questions such as these listed on the slide.

Are all needed definitions provided? Word or phrases without definitions may be understood differently by different people. This problem is especially acute for some technical words, in which different domains have very different definitions. 'Verification' and 'validation' are perhaps the prototypical examples of such words. The agreed definitions of the two words within the computer science / systems engineering communities are almost exactly opposite from the agreed definitions within the controls theory community.

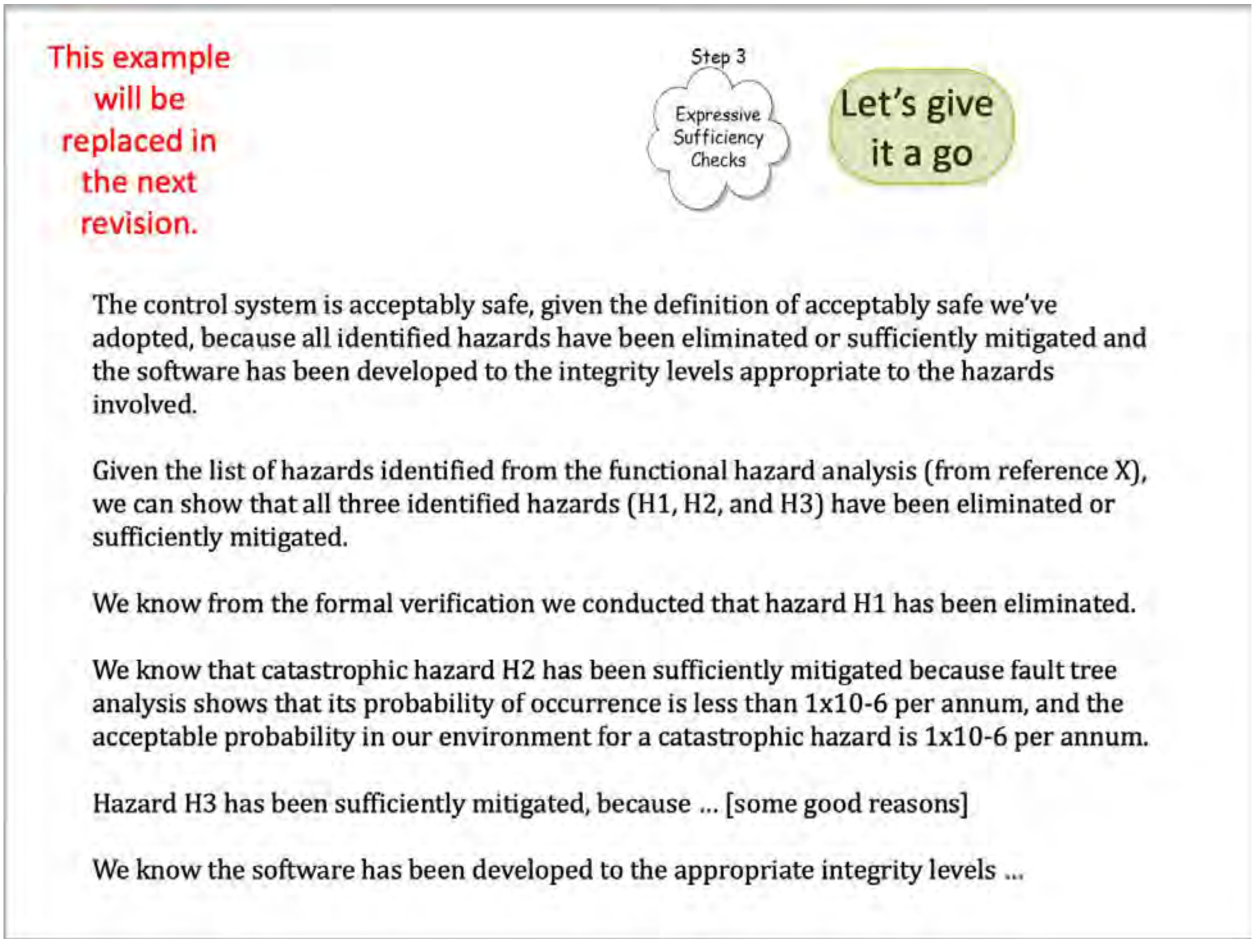
Is the environment adequately described? Failure to describe the environment in which a system or service is expected to operate can easily result in an assurance case that makes invalid assumptions about the operating environment.

Are all premises stated explicitly? As we noted in Module 1, implicit premises are a common occurrence in informal arguments.

Is all reasoning stated explicitly? As we also noted in Module 1, implicit reasoning is even more common than implicit premises. The implicitness of reasoning is especially acute in guidance documents developed without careful regard to assurance arguments. [See, for example, Holloway, C. Michael, & Graydon, P. J. *Explicate '78: Assurance Case Applicability to Digital Systems*. DOT/FAA/TC-17/67. January 2018. Available at https://www.faa.gov/aircraft/air_cert/design_approvals/air_software/media/TC-17-67.pdf].

Finally (well, not really finally as there are other questions one may ask, too, but finally for this list), are relationships clear among argument elements? Can you tell which arguments are sub-arguments of which other ones? Can you work out which conclusions serve as premises for other arguments? And so on.

As you are probably expecting, we'll now see if you can find some sufficiency issues in our example. I've modified it to fix the structural problems we identified in step 2.



This example will be replaced in the next revision.

Step 3

Expressive Sufficiency Checks

Let's give it a go

The control system is acceptably safe, given the definition of acceptably safe we've adopted, because all identified hazards have been eliminated or sufficiently mitigated and the software has been developed to the integrity levels appropriate to the hazards involved.

Given the list of hazards identified from the functional hazard analysis (from reference X), we can show that all three identified hazards (H1, H2, and H3) have been eliminated or sufficiently mitigated.

We know from the formal verification we conducted that hazard H1 has been eliminated.

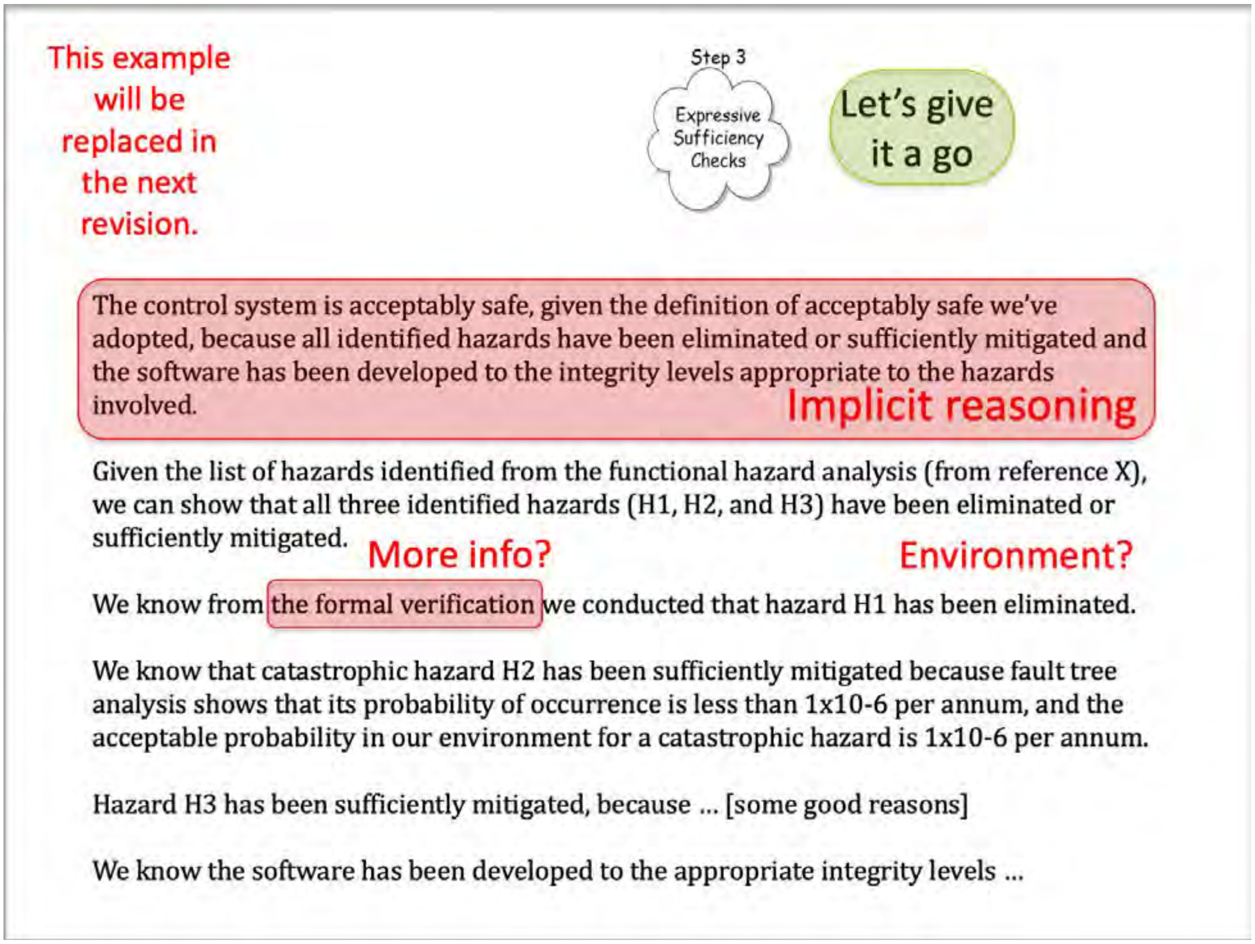
We know that catastrophic hazard H2 has been sufficiently mitigated because fault tree analysis shows that its probability of occurrence is less than 1x10-6 per annum, and the acceptable probability in our environment for a catastrophic hazard is 1x10-6 per annum.

Hazard H3 has been sufficiently mitigated, because ... [some good reasons]

We know the software has been developed to the appropriate integrity levels ...

Please do not turn the page until you have attempted the exercise.

Here’s my answer.



This example will be replaced in the next revision.

Step 3

Expressive Sufficiency Checks

Let’s give it a go

The control system is acceptably safe, given the definition of acceptably safe we’ve adopted, because all identified hazards have been eliminated or sufficiently mitigated and the software has been developed to the integrity levels appropriate to the hazards involved.

Implicit reasoning

Given the list of hazards identified from the functional hazard analysis (from reference X), we can show that all three identified hazards (H1, H2, and H3) have been eliminated or sufficiently mitigated.

More info?

Environment?

We know from the formal verification we conducted that hazard H1 has been eliminated.

We know that catastrophic hazard H2 has been sufficiently mitigated because fault tree analysis shows that its probability of occurrence is less than 1x10-6 per annum, and the acceptable probability in our environment for a catastrophic hazard is 1x10-6 per annum.

Hazard H3 has been sufficiently mitigated, because ... [some good reasons]

We know the software has been developed to the appropriate integrity levels ...

As we already noted, the top-level argument has implicit reasoning.

We might also note that nothing at all is said concerning the environment in which the control system is supposed to operate.

And finally, most likely we need more information about the specifics of the formal verification performed relative to H1 before we can know whether relying on the verification provides sufficient grounds for believing the hazard has been eliminated.

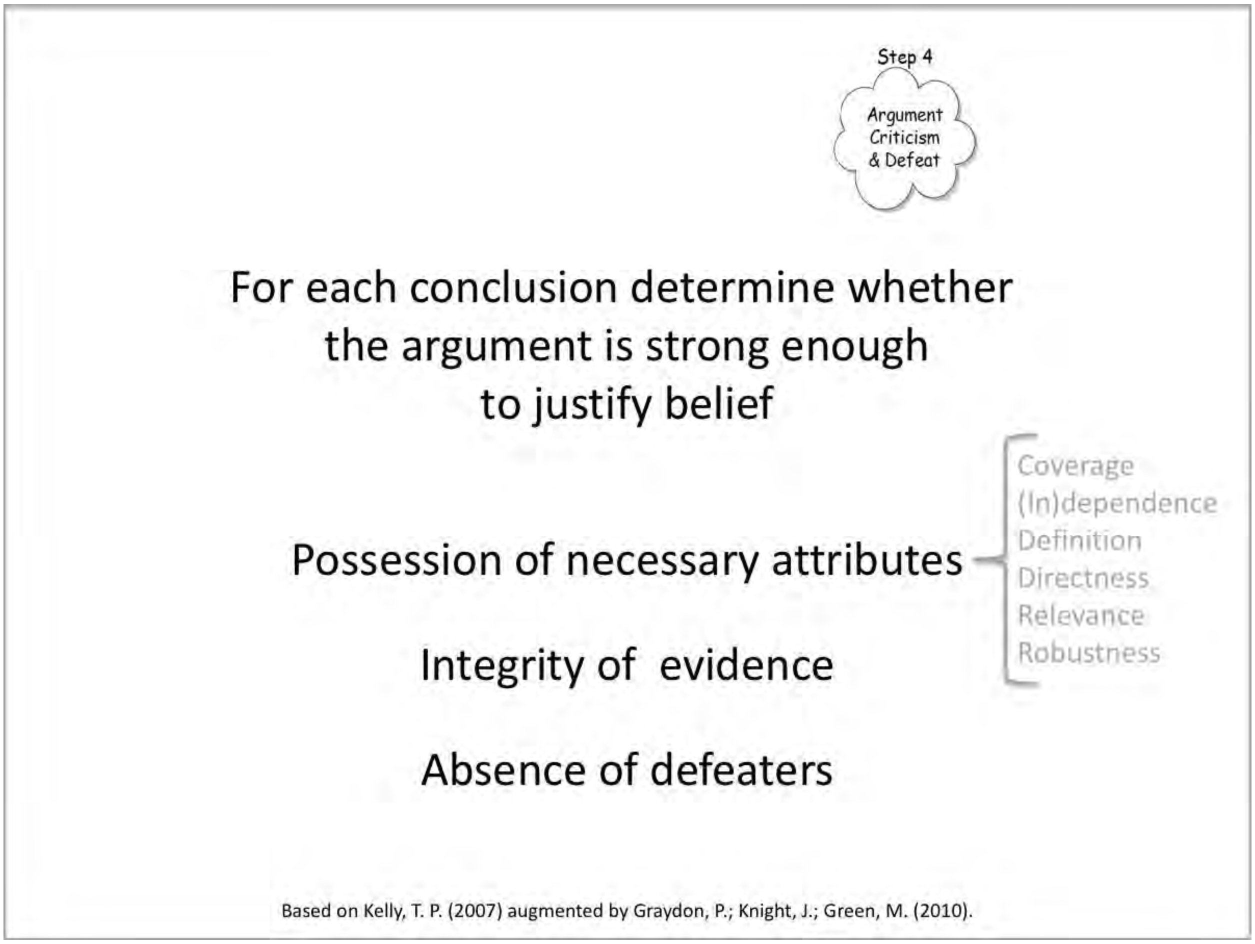
That’s the third step.

[Note to participants: What questions do you have?]

The final step is argument criticism, in which for each conclusion, we seek to determine whether the argument for it is strong enough to justify belief.

This is the most time-consuming and subjective step in the process.

In Kelly’s process, there are 3 aspects to this quest: (1) Possession of necessary attributes; (2) Integrity of evidence; (3) and Absence of defeaters.



Step 4

Argument Criticism & Defeat

For each conclusion determine whether the argument is strong enough to justify belief

Possession of necessary attributes

- Coverage
- (In)dependence
- Definition
- Directness
- Relevance
- Robustness

Integrity of evidence

Absence of defeaters

Based on Kelly, T. P. (2007) augmented by Graydon, P.; Knight, J.; Green, M. (2010).

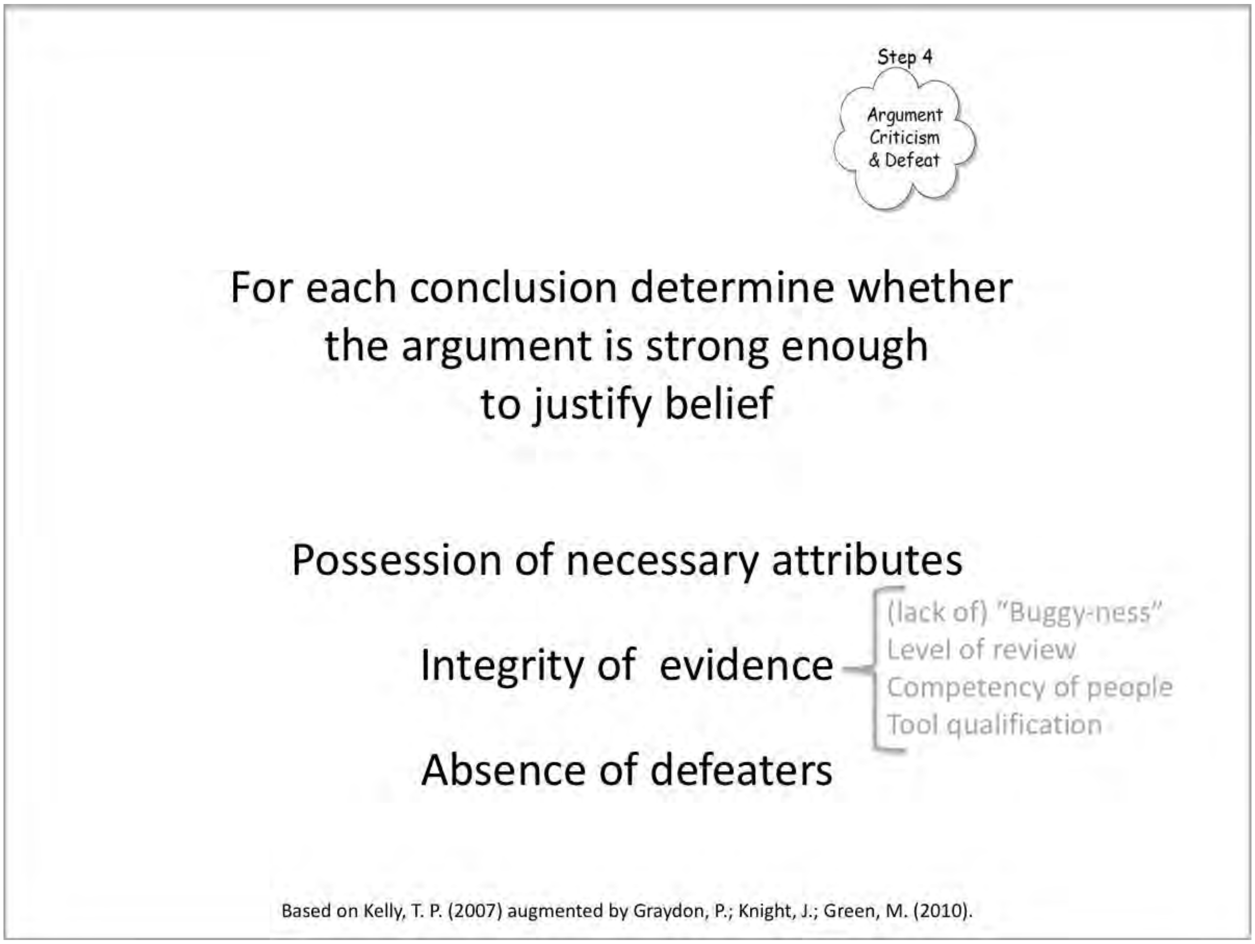
We're going to concentrate on the 3rd one of these, but I'll mention just a bit more about the first two.

'Attributes' here refers to attributes of the individual argument being considered, not of the overall case itself. In his paper, Tim lists the six attributes you see here (coverage, dependence/independence, definition, directness, relevance, and robustness), while noting that the list is not complete, and the attributes are not necessarily disjoint.

As one example, suppose an argument is claimed to support the conclusion, "All identified hazards have been addressed", but which is based on premises concerning only three hazards, when the hazard analysis shows seven hazards were identified. Such an argument would not have adequate *coverage*.

Concerning *integrity of evidence* Tim Kelly specifically mentions four considerations: (lack of) "buggy-ness", level of review, competency of people, and tool qualification.

The basic idea is that in evaluating an argument we need to have confidence that the grounded premises (as I've mentioned before, I consider this phrase a much better phrase than evidence) at the base of our argument really say what we think they say.



Step 4

Argument Criticism & Defeat

For each conclusion determine whether the argument is strong enough to justify belief

Possession of necessary attributes

Integrity of evidence

- (lack of) "Buggy-ness"
- Level of review
- Competency of people
- Tool qualification

Absence of defeaters

Based on Kelly, T. P. (2007) augmented by Graydon, P.; Knight, J.; Green, M. (2010).

[Question to participants: What questions do you have about possession of necessary attributes and integrity of evidence?]

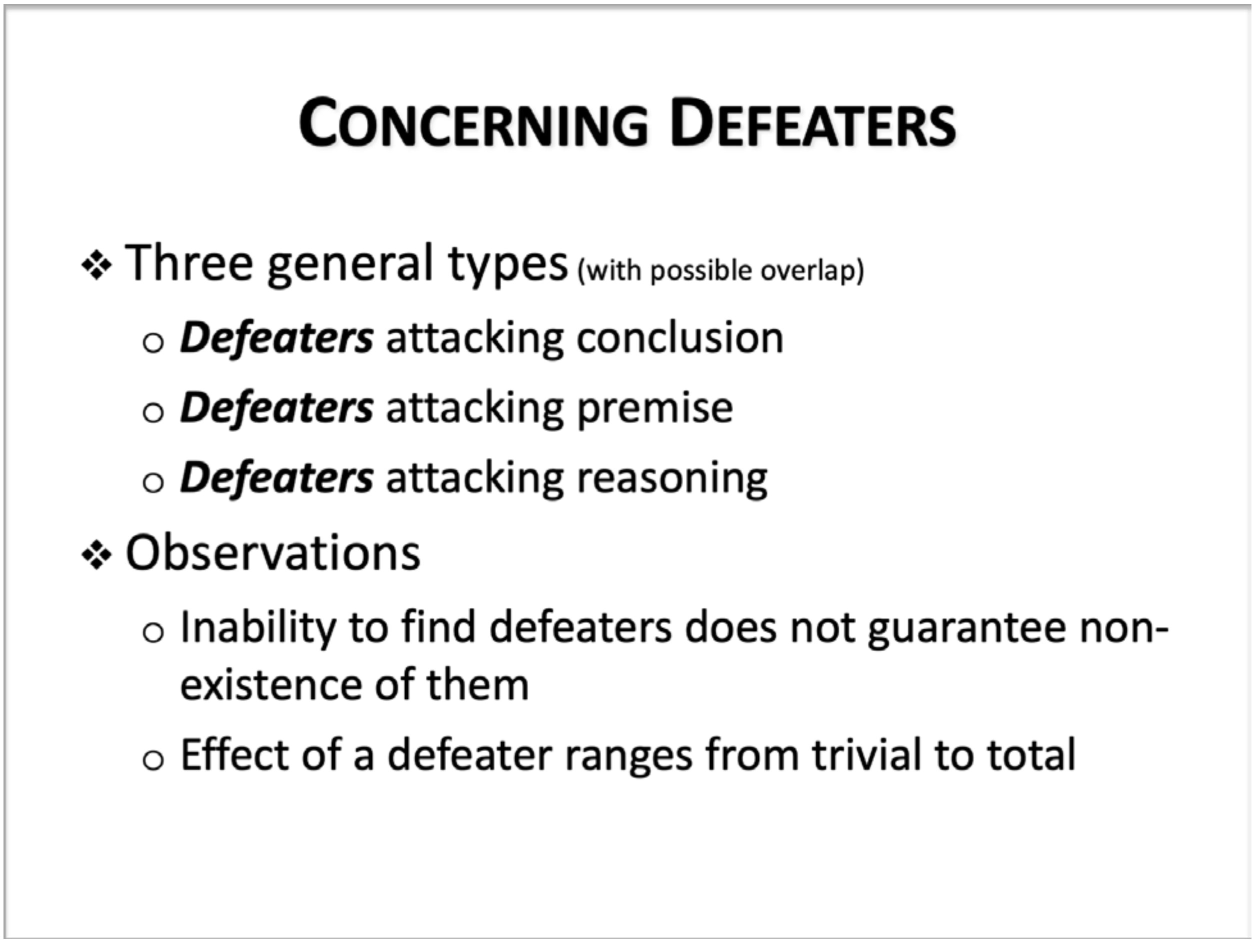
I want to talk now in a bit more detail about the last element, absence of defeaters.

Although you may not have heard the term *defeater* before in quite this context, your intuitive notion of what it means is likely fairly accurate. Rather than giving an abstract definition[2], I'll talk about three basic types and then lead us through some examples.

There are three general types: *defeaters* that attack a conclusion; *defeaters* attacking a premise; and *defeaters* attacking reasoning.

Some treatments of defeaters give different names to the different types (rebutting, undermining, and undercutting); and some only distinguish between two types, grouping premise and reasoning defeaters together; but we're not going to use those names as they can be a bit confusing.

[2] The reader interested in the philosophical foundations of the concept of defeaters is encouraged to visit `https://plato.stanford.edu/entries/reasoning-defeasible/`. I have avoided using the phrase 'defeasible reasoning' in these educational materials based on prior experiences in which the use of the phrase caused more confusion than enlightenment.



CONCERNING DEFEATERS

- Three general types (with possible overlap)
  - ***Defeaters*** attacking conclusion
  - ***Defeaters*** attacking premise
  - ***Defeaters*** attacking reasoning
- Observations
  - Inability to find defeaters does not guarantee non-existence of them
  - Effect of a defeater ranges from trivial to total

The three-fold categorization is not absolute, as there can be defeaters that attack more than one element of an argument.

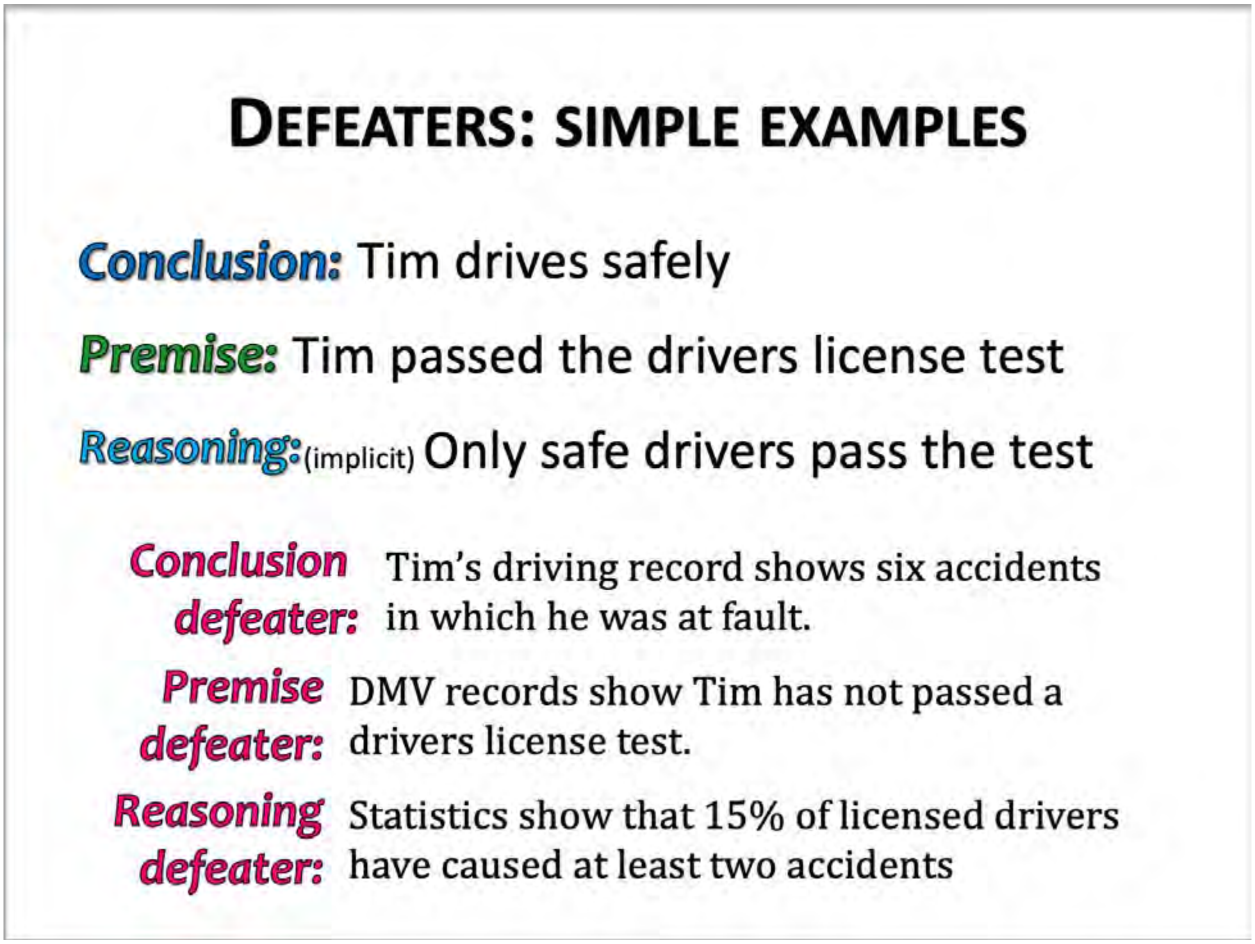
Before giving an example, I want to make two important observations concerning defeaters.

First, defeaters are partially analogous to software bugs in that the inability to find defeaters does not guarantee there aren't any.

Second the effect of a defeater on an argument ranges from trivial to total. Only defeaters that effectively attack the conclusion mean that the conclusion is necessarily not true.

Premise and reasoning defeaters show there's something wrong with the argument, but this may not necessarily mean that its conclusion is false. As an example, we will harken back to Module 1. Recall one of the simple examples we discussed was the following: Given (premise) "Annette was born in Lynchburg, Virginia" you should believe (conclusion) "Annette is a US citizen" because (reasoning) "People born in Virginia are US citizens." Supposed you discover that Annette was not born in Lynchburg, Virginia. You have defeated the premise. But the conclusion could still be true. As long as she was born in some other location in Virginia, the reasoning does not even need to change.

Also harkening back to Module 1 you may recall this simple argument.



DEFEATERS: SIMPLE EXAMPLES

Conclusion: Tim drives safely

Premise: Tim passed the drivers license test

Reasoning:(implicit) Only safe drivers pass the test

Conclusion defeater: Tim's driving record shows six accidents in which he was at fault.

Premise defeater: DMV records show Tim has not passed a drivers license test.

Reasoning defeater: Statistics show that 15% of licensed drivers have caused at least two accidents

For our purposes now, let's suppose that we have an agreed definition of what it means to 'drive safely' and that this definition involves, at least in part, the absence of 'at fault' accidents.

Let's start with the implicit reasoning, and consider what a defeater of this reasoning might look like. Well, suppose we have access to accident statistics that show 15% of licensed drivers have caused at least two accidents. Such statistics would certainly undercut our belief that *only* safe drivers pass the test.

This, in itself, doesn't mean that Tim doesn't drive safely, but it does mean that the argument provided should not give us confidence in his driving ability.

Consider the premise: "Tim passed the drivers license test". What's a defeater for this premise?

Well, the premise would be thoroughly undermined If DMV records show Tim has not passed the test.

Again, the argument now provides no confidence in the conclusion, but that alone doesn't mean the conclusion is false.

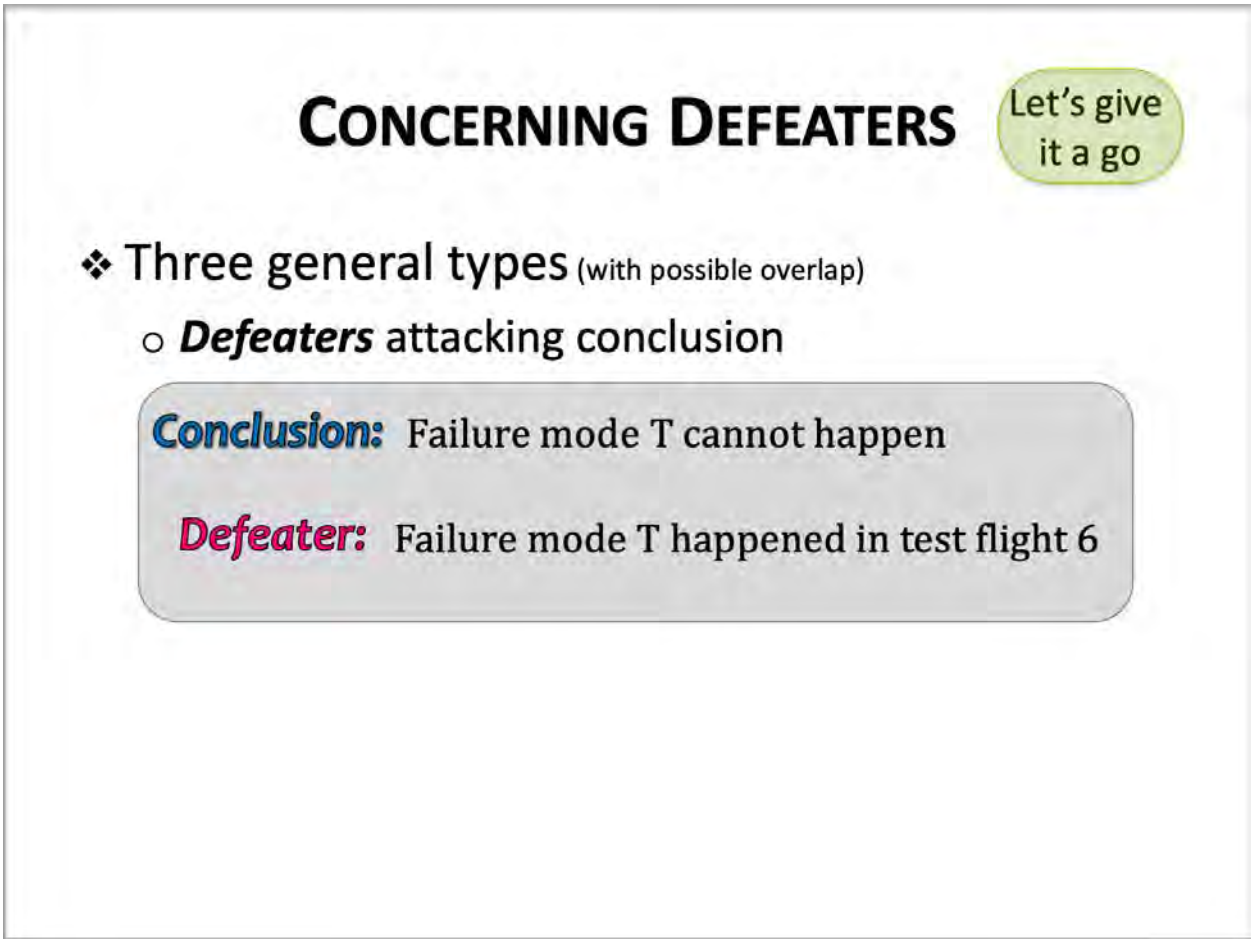
A defeater of the conclusion, on the other hand, does mean that the conclusion is false.

Here’s a possible example of such a defeater: “Tim’s driving record shows six accidents in which he was at fault.” Given such a record, I don’t know of anyone who would conclude that Tim drives safely.

Now that you’ve seen these simple examples, you’re ready to try some examples that are a bit more technically oriented. Let’s start with a conclusion defeater.

Suppose we have the conclusion “Failure Mode T cannot happen”.

What’s a defeater for it?



Concerning Defeaters

Let’s give it a go

- Three general types (with possible overlap)
  - ***Defeaters*** attacking conclusion

Conclusion: Failure mode T cannot happen

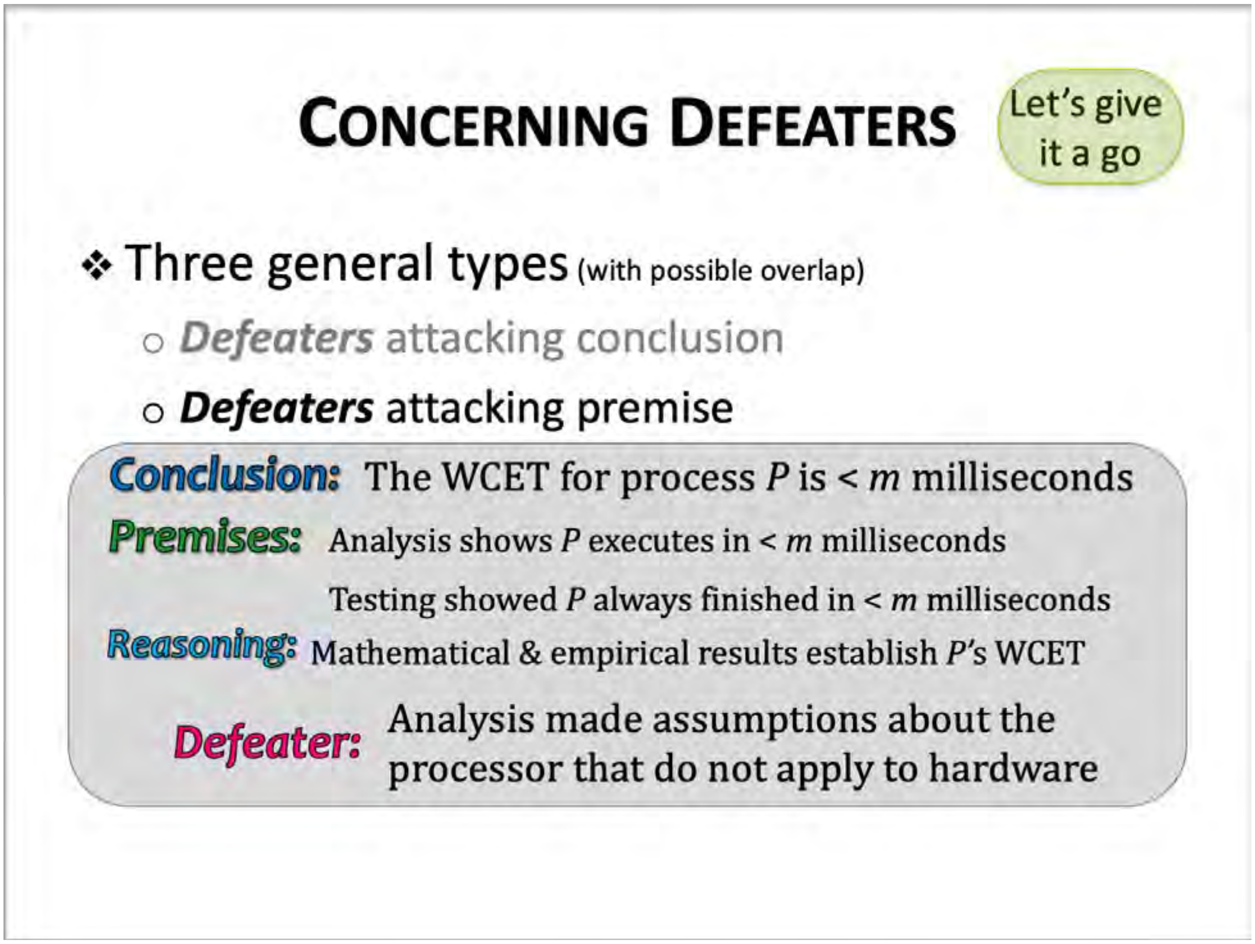
Defeater: Failure mode T happened in test flight 6

Here’s one example: “Failure mode T happened in test flight 6.” If it actually happened, the claim that it cannot happen cannot be true.

Let’s take a shot at finding a defeater of a premise.

Suppose we have conclusion “The WCET for process *P* is < *m* milliseconds”, with the two premises “The WCET for process *P* is < *m* milliseconds” and “Testing showed *P* always finished in < *m* milliseconds”, and the reasoning “Mathematical & empirical results establish *P*’s WCET” What’s an example of a defeater that attacks the truth of one or more of the premises?

Please do not turn the page until you have attempted the exercise.



Concerning Defeaters

Let's give it a go

- Three general types (with possible overlap)
  - ○ ***Defeaters*** attacking conclusion
  - ○ ***Defeaters*** attacking premise

Conclusion: The WCET for process *P* is < *m* milliseconds

Premises: Analysis shows *P* executes in < *m* milliseconds

Testing showed *P* always finished in < *m* milliseconds

Reasoning: Mathematical & empirical results establish *P*'s WCET

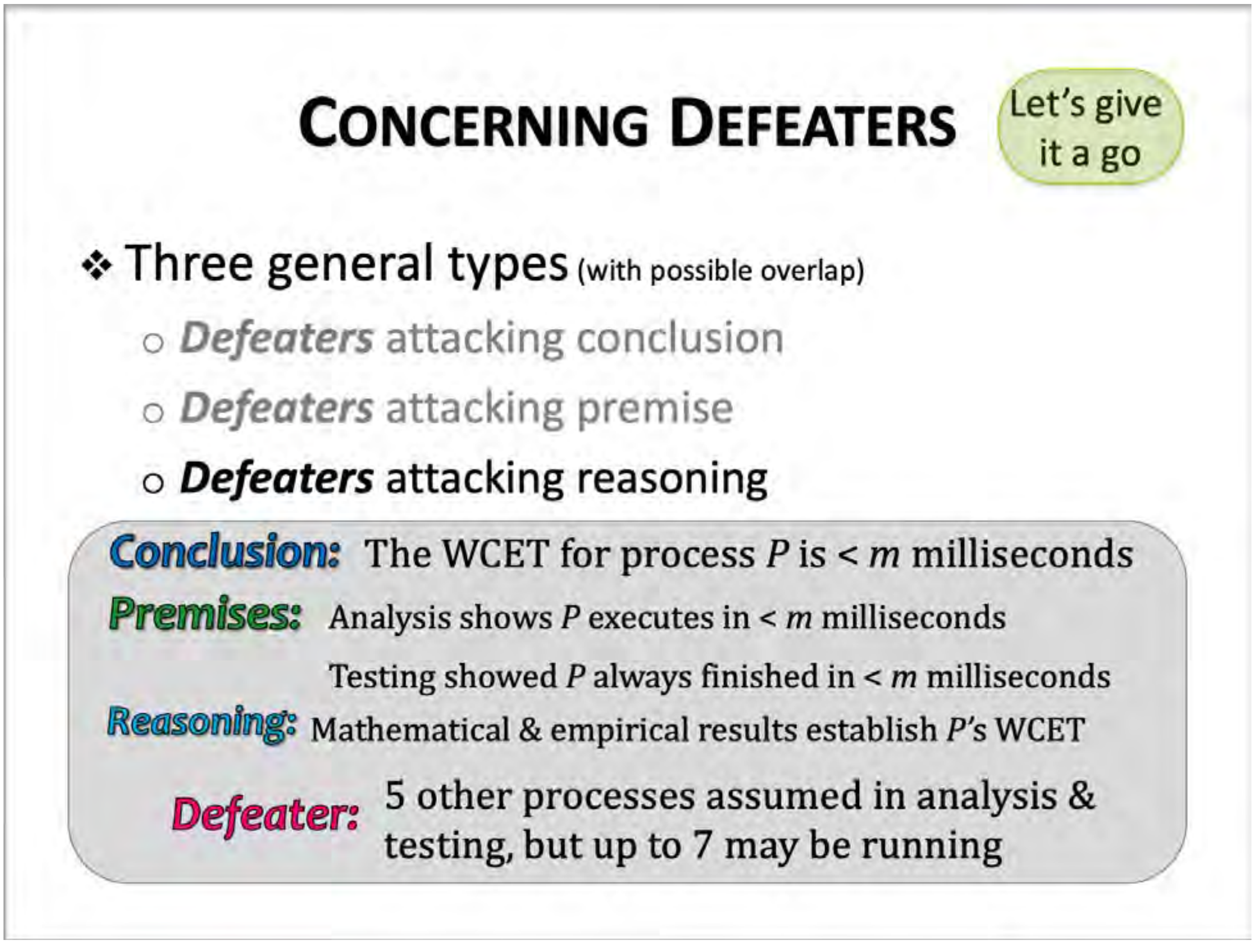
Defeater: Analysis made assumptions about the processor that do not apply to hardware

Here's one: "The Analysis made assumptions about the processor that do not apply to the actual hardware."

Finally, let's consider a defeater attacking reasoning using the same argument we just used, and assuming the premise defeater has been shown to not apply.

What's a possible defeater of the reasoning?

Please do not turn the page until you have attempted the exercise.



Concerning Defeaters

Let's give it a go

❖ Three general types (with possible overlap)

- ○ ***Defeaters*** attacking conclusion
- ○ ***Defeaters*** attacking premise
- ○ ***Defeaters*** attacking reasoning

Conclusion: The WCET for process *P* is < *m* milliseconds

Premises: Analysis shows *P* executes in < *m* milliseconds

Testing showed *P* always finished in < *m* milliseconds

Reasoning: Mathematical & empirical results establish *P*'s WCET

Defeater: 5 other processes assumed in analysis & testing, but up to 7 may be running

Here's one: "5 other processes assumed in analysis & testing, but up to 7 may be running"

If that assertion is true, the reasoning is weakened.

There's a lot more we could say here about defeaters, but this is a good time to pause for questions.

[Question to participants: Who has a question?]

Here are four questions for you to consider at your leisure:

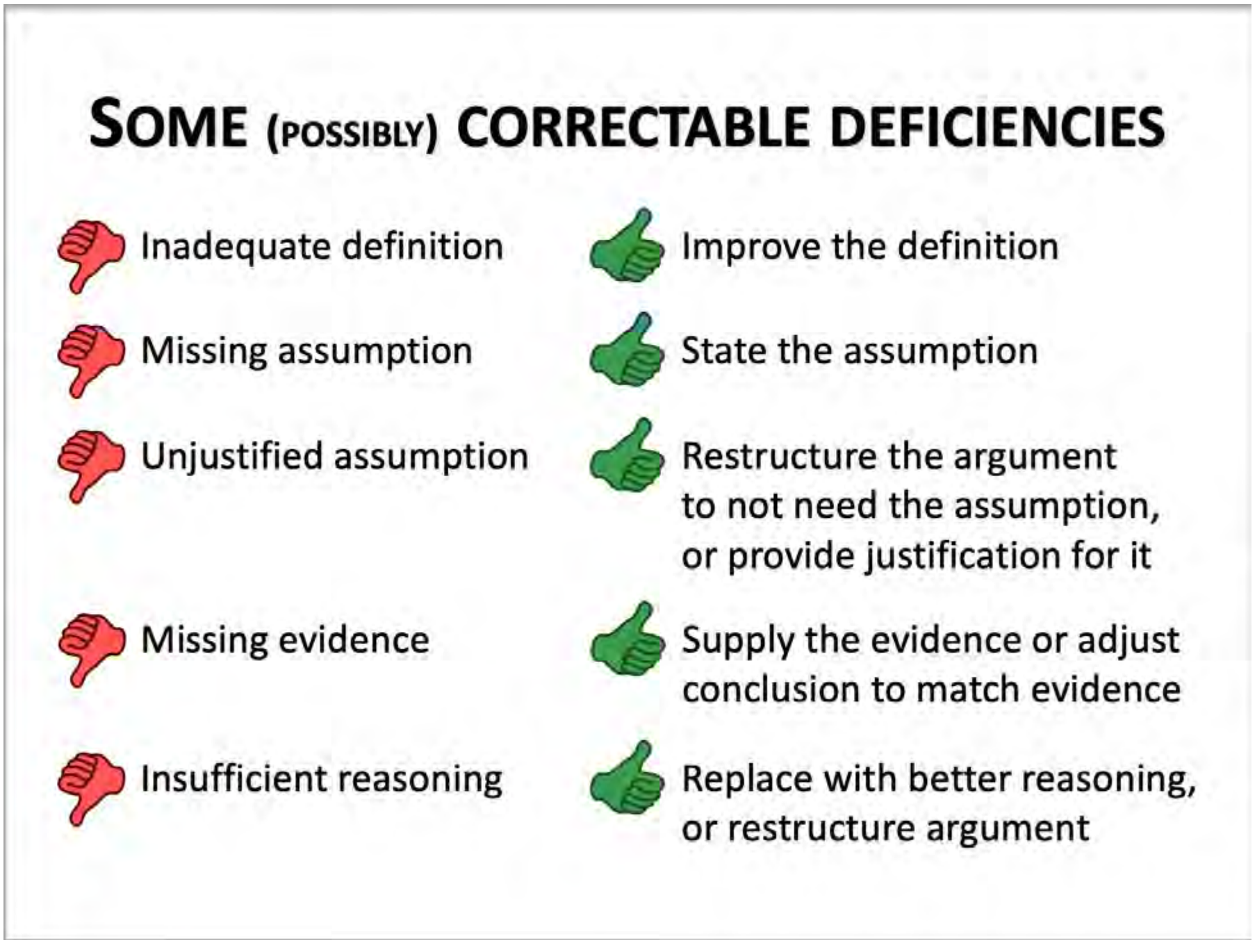
Does the number of possible defeaters of an argument say anything important about the cogency of the argument?

Does the number of resolved defeaters say anything important about the cogency of an argument? (A resolved defeater is a possible defeater that has been shown to not apply to the argument.)

Does the number of unresolved defeaters say anything important about the cogency of an argument?

How about the ratio of resolved to unresolved defeaters?

This slide lists some deficiencies that may be encountered in an argument, but for which there may be fairly simple corrections possible.



Some (possibly) correctable deficiencies

👎 Inadequate definition	👍 Improve the definition
👎 Missing assumption	👍 State the assumption
👎 Unjustified assumption	👍 Restructure the argument to not need the assumption, or provide justification for it
👎 Missing evidence	👍 Supply the evidence or adjust conclusion to match evidence
👎 Insufficient reasoning	👍 Replace with better reasoning, or restructure argument

For an inadequate definition, improving the definition *may* be an easy thing to do.

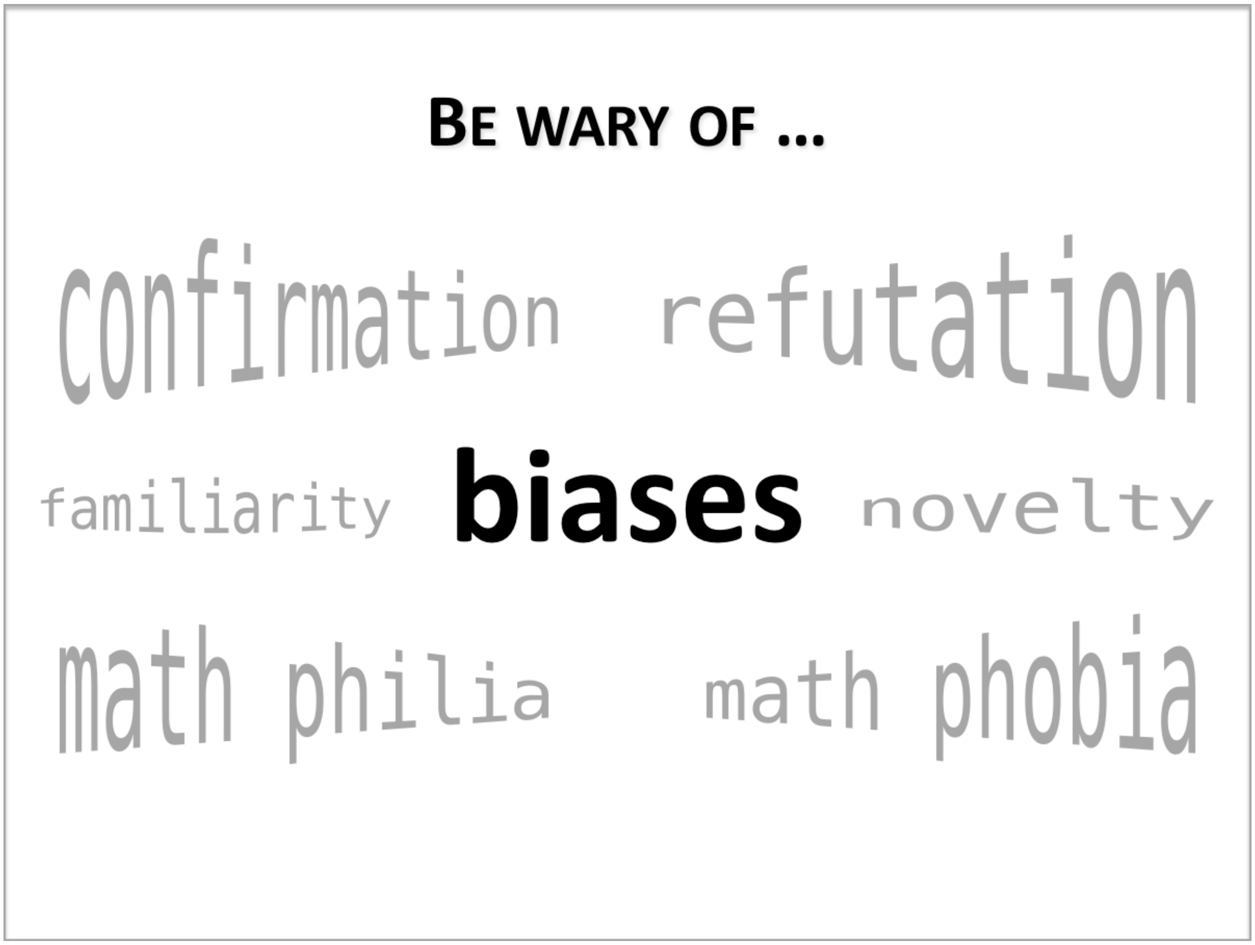
Perhaps a missing assumption can simply be stated.

It may be possible to eliminate an unjustified assumption, or to provide justification for the assumption.

For missing evidence, supplying the evidence may be possible, or perhaps the conclusion may be adjusted slightly to match the evidence. Say, for example the original conclusion was the system would be safe to operate with an ambient temperature between 0 and 100 degrees, but the available evidence only covers the range of 0 to 80 degrees. The temperature range in the conclusion could be adjusted accordingly.

And, as the final example, insufficient reasoning may be replaceable by a better one, or perhaps the argument can be restructured so this reasoning step is replaced altogether.

Let’s talk now for a bit about some things to be wary of when performing step four.



BE WARY OF …

confirmation refutation

familiarity **biases** novelty

math philia math phobia

One thing to look out for is biases, of which there are several types.

The most common type of bias mentioned concerning assurance cases is confirmation bias.

In general this phrase refers to the tendency to interpret new pieces of information in a way that confirms what you already think, rather than to interpret it critically.

But some people (such as myself) may be prone to a bias of a slightly different sort, namely a tendency towards refutation, or, to put it slightly differently, to interpret *everything* critically.

Another pair of biases to be wary of are the love of math and the fear of math.

For some people, seeing numbers (probabilities for example) in an assurance case will cause them to think happy thoughts and be inclined to believe that the case is a good one.

For some other people (me for example) seeing numbers (probabilities in particular) in an assurance case will make them nauseous, and nearly certain that the case is rubbish.

Finally, the third pair of biases that can cause problems is familiarity and novelty. Some folks tend to give more credence to things they know, whereas others tend to give more credence to things that are new.

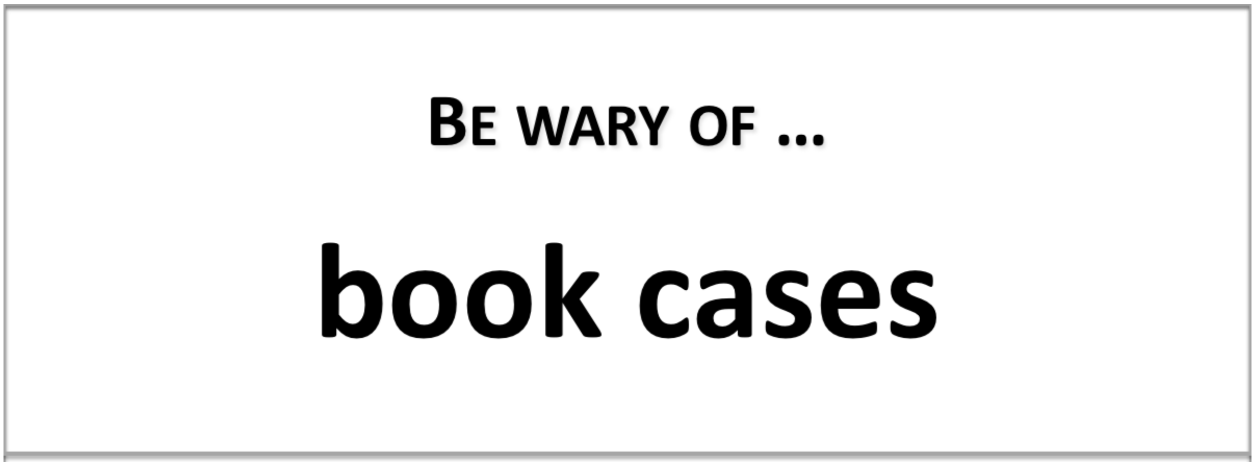
In general humans are much better at recognizing biases in others than we are at recognizing biases in ourselves. Having multiple people participate in the evaluation of an assurance case is one way to reduce the likelihood of the evaluation being skewed by biases.

Another set of things of which we need to be wary concern the cases themselves.

Be wary of …

centipedes

We need to be wary of any argument that has a whole lot of premises for any particular conclusion.

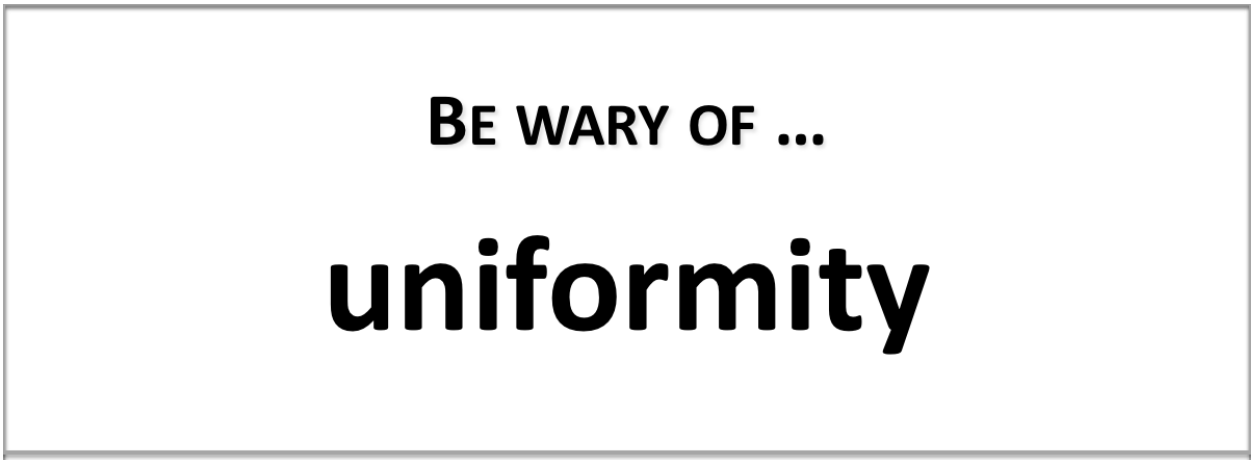


Be wary of …

book cases

Really big assurance cases should cause concern, even if the individual arguments are not centipedes, but simply the level of detail is very great.

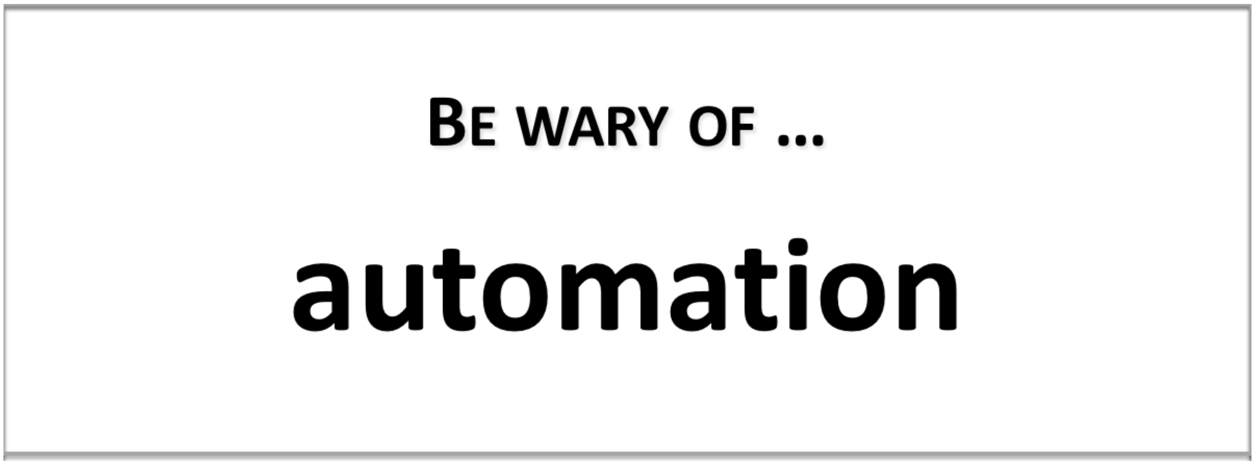
Both centipedes and bookcases are worrisome because understanding them may well exceed the intellectual capabilities of even the brightest evaluator.



Be wary of …

uniformity

Also worrisome are assurance cases in which everything seems to have been given the same level of attention, suggesting insufficient consideration of some of the issues we raised in Module 2 when we discussed the 5Ws.



Be wary of …

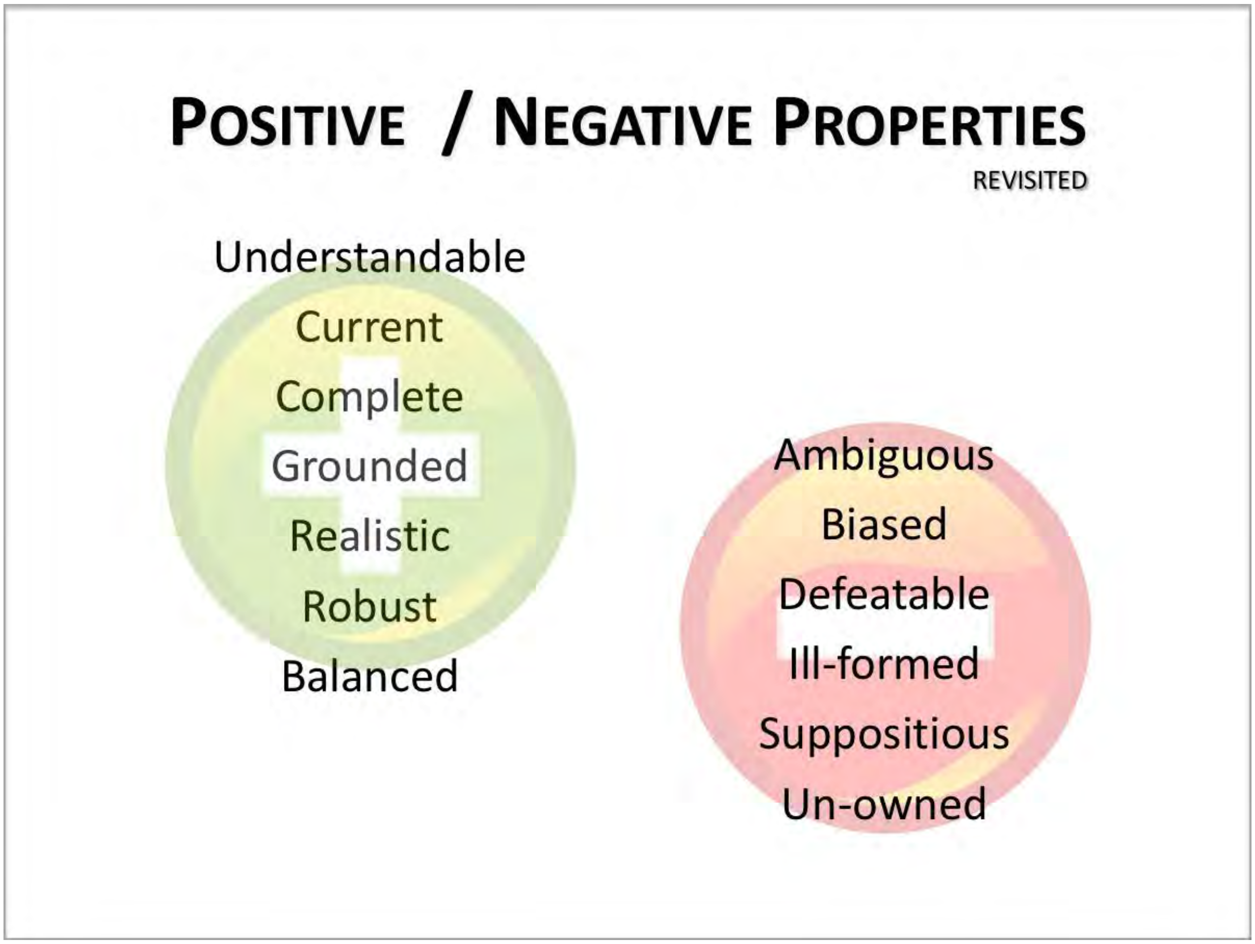
automation

Finally, you should run as fast as you can away from any assurance case that someone says was generated automatically. As we've said several times in every module so far, a primary value of assurance cases arises from how they can stimulate careful thinking. Automation is the antithesis of careful thinking[3].

[Question to participants: What questions do you have about the causes of wariness?]

Before ending, let's quickly revisit the list of positive & negative properties.

[3] Careful, thoughtful use of automation to generate documentation to support an assurance case (for example, providing links to various bits of evidence) may be appropriate. Automating the creation of arguments (see Module 4) or the detailed evaluation of them is fraught with danger. Perhaps the day will come when sufficient foundations will have been laid for effective creation or evaluation, but those days are not yet here.



Positive / Negative Properties

REVISITED

Understandable
Current
Complete
Grounded
Realistic
Robust
Balanced

Ambiguous
Biased
Defeatable
Ill-formed
Suppositious
Un-owned

I explained the meaning of the positive properties earlier, but deferred discussing the negative ones until now.

Ambiguous is self-evident.

We just finished talking about various ways an assurance case can be *biased*.

As you can probably guess, a *defeatable* assurance case is one in which unresolved defeaters exist for important parts of the argument.

An *ill-formed* case is one that doesn't make it out of step 2 in the four step process.

By *Suppositious* I mean an assurance case with arguments based heavily on assumptions, for which no further argument or premises are provided.

Finally, *un-owned* refers to an assurance case for which the "Who" question hasn't been well answered.

Evaluating an assurance case is not easy, but it is not impossible. Subjectivity *is* necessarily involved, but subjectivity is necessarily involved in evaluating *anything* other than, perhaps, some aspects of pure mathematics.

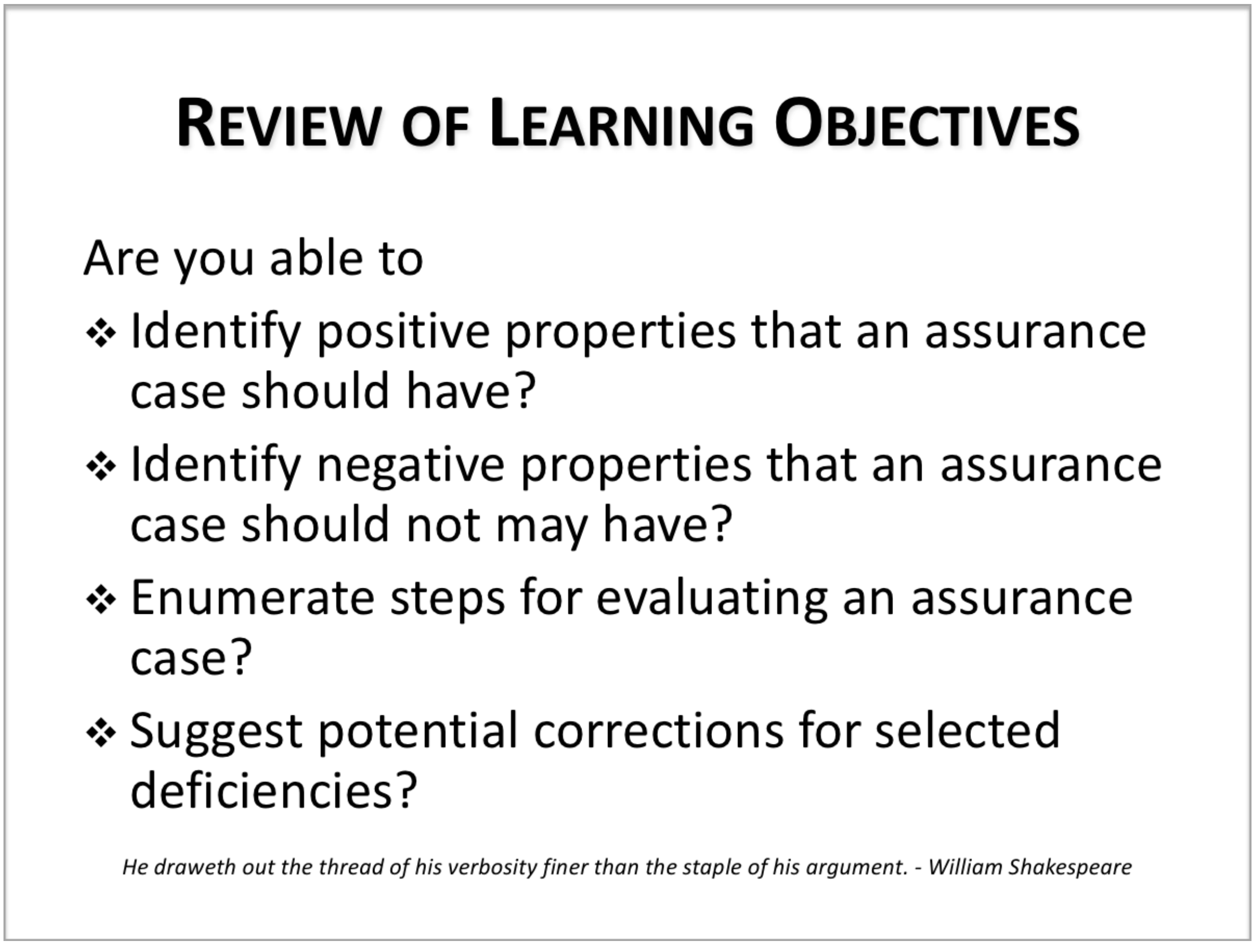
Just as assurance cases provide a framework for making *explicit* conclusions, premises, reasoning (and other argument elements), so too do they provide a framework for making the areas of subjectivity *explicit*, and thus subject to scrutiny.

Subjectivity that is subject to scrutiny is surely more desirable than subjectivity that's hidden.

[Question to participants: Any questions before we end by reviewing the learning objectives?]

At the beginning, I listed four things that I hoped you'd be able to do by the end of this module.

Here are those four things recast in the form of questions.



REVIEW OF LEARNING OBJECTIVES

Are you able to

- Identify positive properties that an assurance case should have?
- Identify negative properties that an assurance case should not may have?
- Enumerate steps for evaluating an assurance case?
- Suggest potential corrections for selected deficiencies?

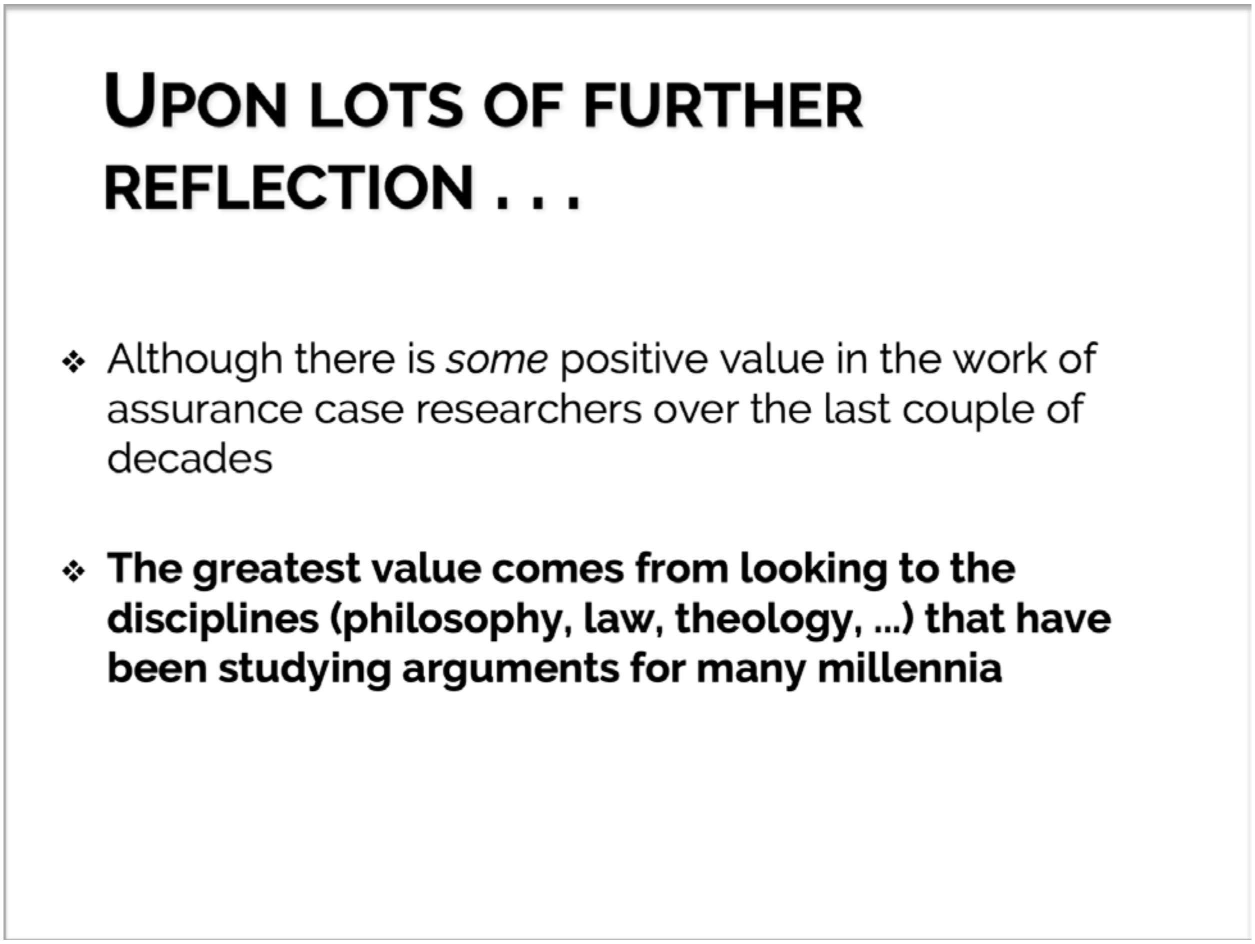
He draweth out the thread of his verbosity finer than the staple of his argument. - William Shakespeare

Think to yourself how you'd answer these questions.

After you've thought about the questions for a little bit, please ask me any questions that you still have for me.

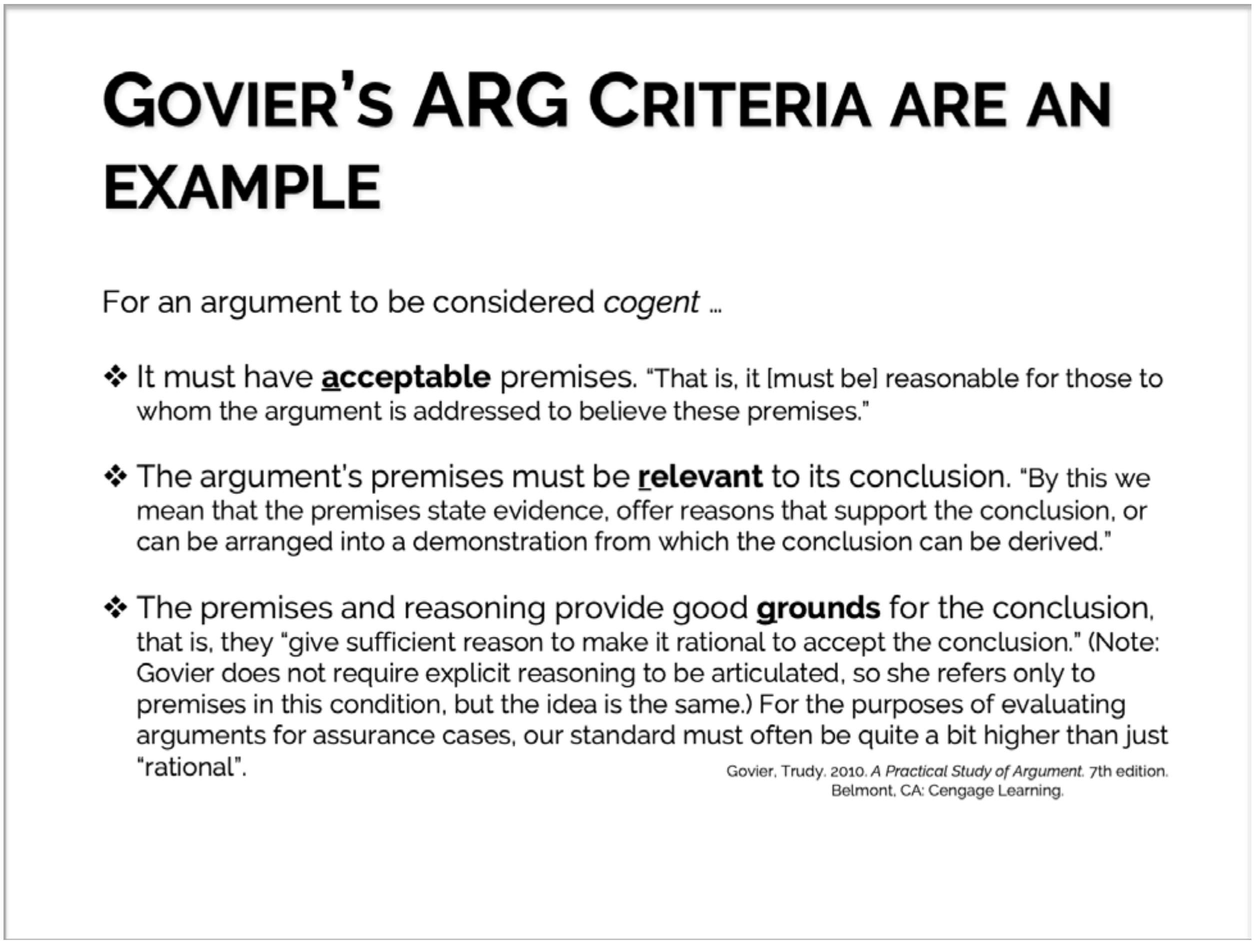
If you have questions or comments about this material contact the author at `c.michael.holloway@nasa.gov.`

The following material is included without written commentary as a preview of what is likely to be coming when I revise this module.



UPON LOTS OF FURTHER REFLECTION . . .

- Although there is *some* positive value in the work of assurance case researchers over the last couple of decades
- **The greatest value comes from looking to the disciplines (philosophy, law, theology, …) that have been studying arguments for many millennia**

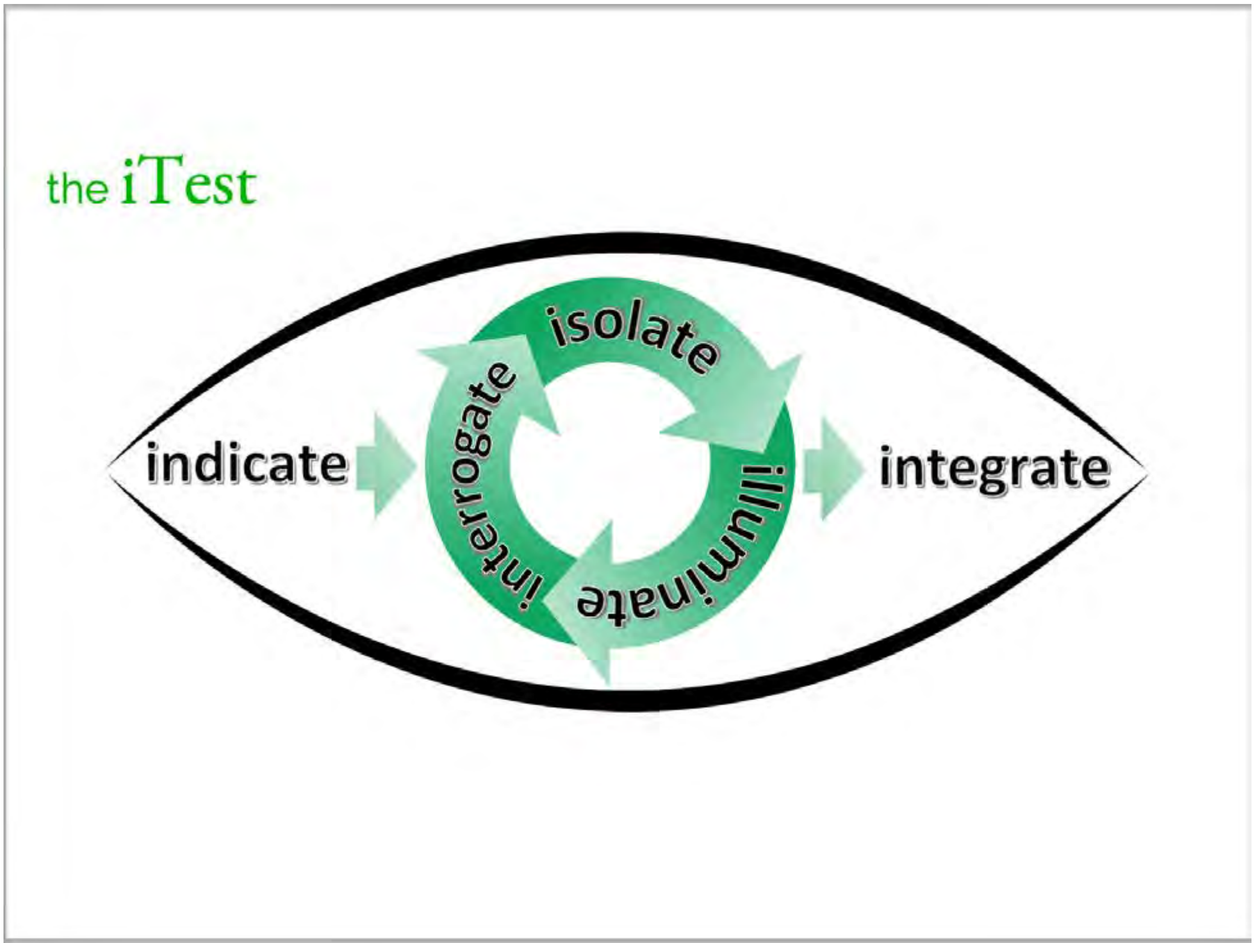


GOVIER'S ARG CRITERIA ARE AN EXAMPLE

For an argument to be considered *cogent* …

- ❖ It must have **acceptable** premises. "That is, it [must be] reasonable for those to whom the argument is addressed to believe these premises."
- ❖ The argument's premises must be **relevant** to its conclusion. "By this we mean that the premises state evidence, offer reasons that support the conclusion, or can be arranged into a demonstration from which the conclusion can be derived."
- ❖ The premises and reasoning provide good **grounds** for the conclusion, that is, they "give sufficient reason to make it rational to accept the conclusion." (Note: Govier does not require explicit reasoning to be articulated, so she refers only to premises in this condition, but the idea is the same.) For the purposes of evaluating arguments for assurance cases, our standard must often be quite a bit higher than just "rational".

Govier, Trudy. 2010. *A Practical Study of Argument*. 7th edition. Belmont, CA: Cengage Learning.



the iTest

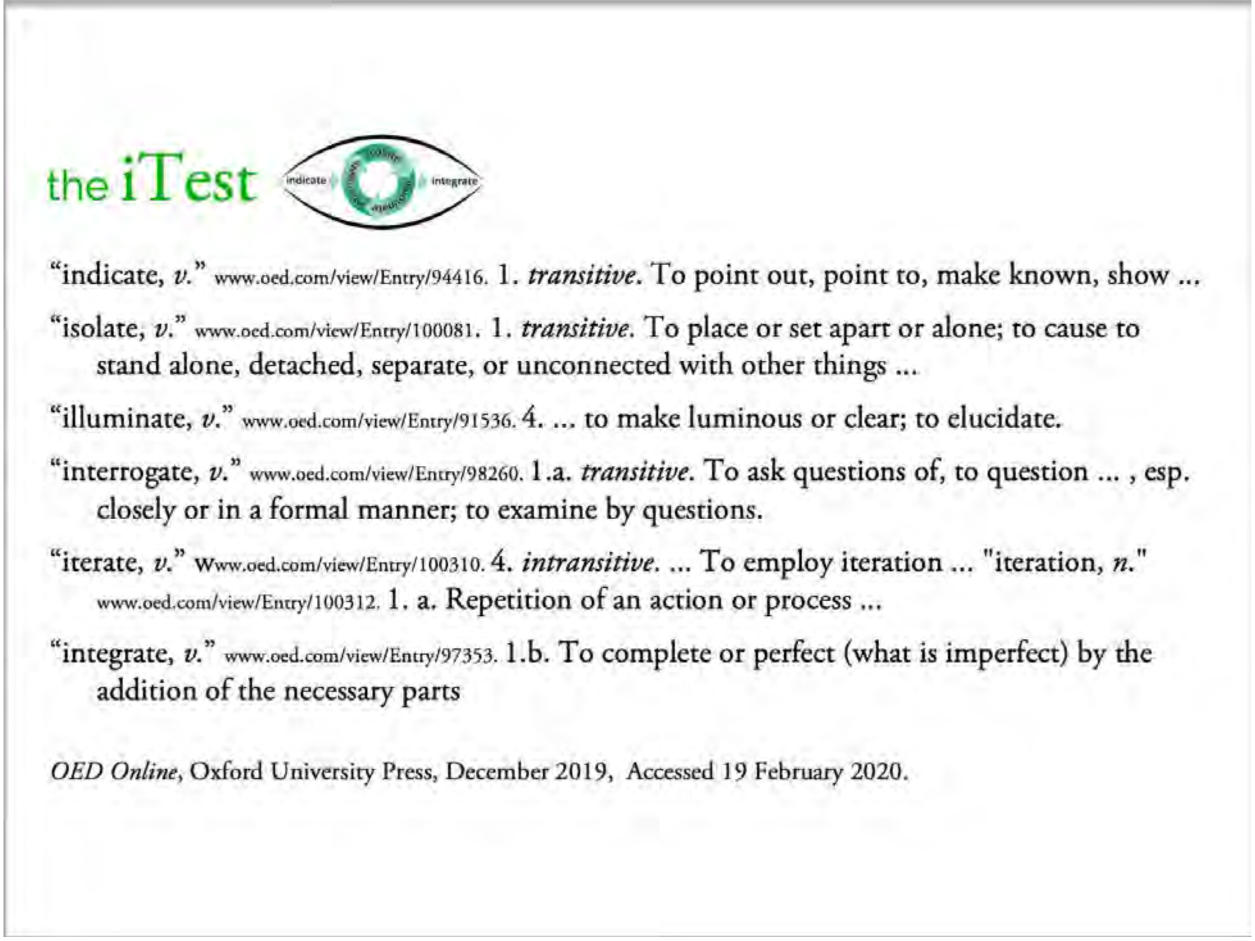
indicate

isolate

illuminate

interrogate

integrate



the iTest

indicate
integrate

"indicate, *v.*" www.oed.com/view/Entry/94416. 1. ***transitive***. To point out, point to, make known, show ...

"isolate, *v.*" www.oed.com/view/Entry/100081. 1. ***transitive***. To place or set apart or alone; to cause to stand alone, detached, separate, or unconnected with other things ...

"illuminate, *v.*" www.oed.com/view/Entry/91536. 4. ... to make luminous or clear; to elucidate.

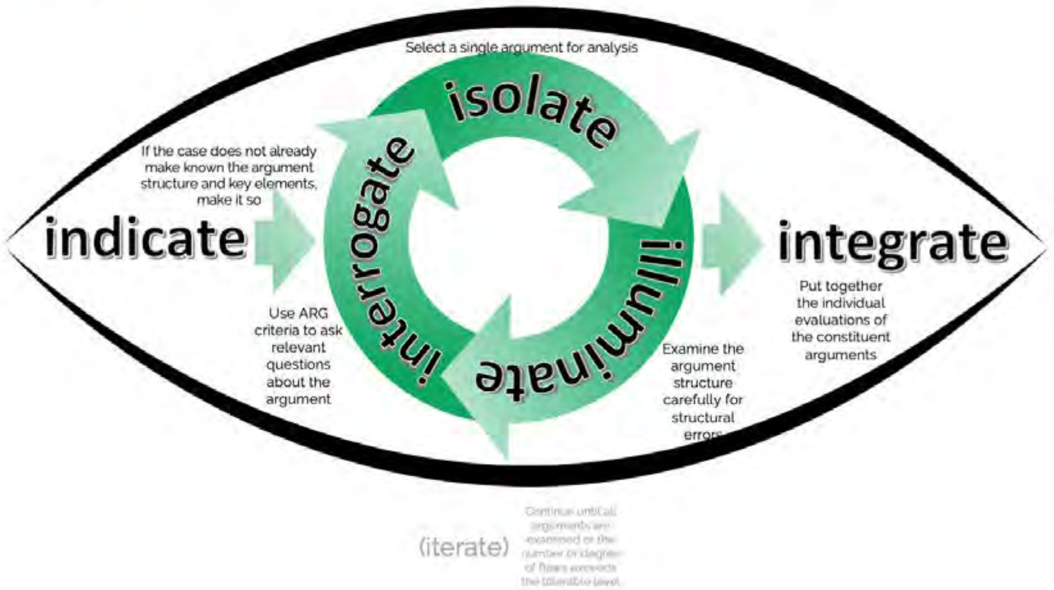
"interrogate, *v.*" www.oed.com/view/Entry/98260. 1.a. ***transitive***. To ask questions of, to question ... , esp. closely or in a formal manner; to examine by questions.

"iterate, *v.*" Www.oed.com/view/Entry/100310. 4. ***intransitive***. ... To employ iteration ... "iteration, *n.*" www.oed.com/view/Entry/100312. 1. a. Repetition of an action or process ...

"integrate, *v.*" www.oed.com/view/Entry/97353. 1.b. To complete or perfect (what is imperfect) by the addition of the necessary parts

OED Online, Oxford University Press, December 2019, Accessed 19 February 2020.

the iTest


Select a single argument for analysis
isolate
If the case does not already make known the argument structure and key elements, make it so
indicate
interrogate
illuminate
integrate
Put together the individual evaluations of the constituent arguments
Use ARG criteria to ask relevant questions about the argument
Examine the argument structure carefully for structural errors
(iterate)
Continue until all arguments are examined or the number or degree of flaws exceeds the tolerable level.

the iTest

indicate if the case does not already make known the argument structure and key elements, make it so

isolate select a single argument for analysis

illuminate examine the argument carefully for structural errors

interrogate use the ARG criteria to ask relevant questions about the argument

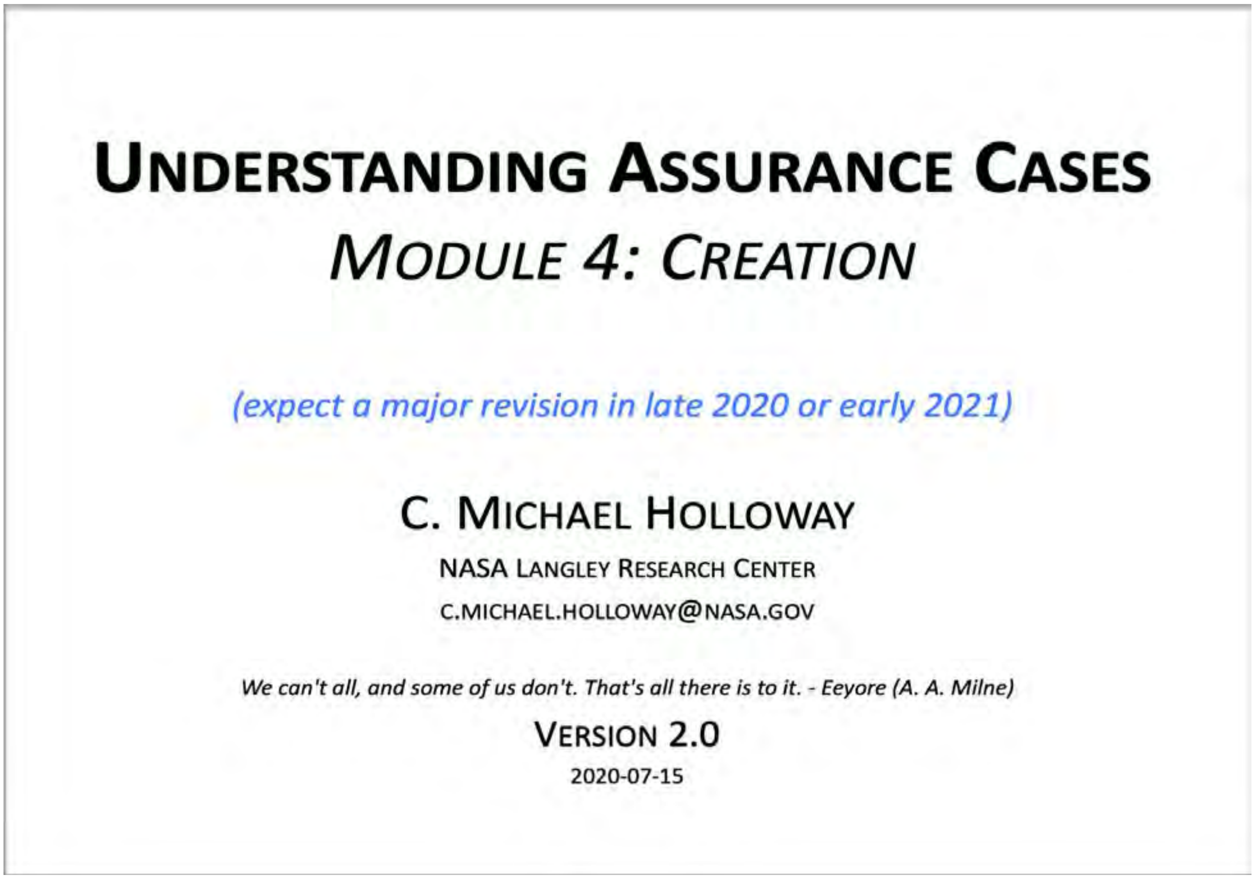
integrate put together the individual evaluations of the constituent arguments

Understanding Assurance Cases: An Educational Presentation in Five Parts

Module 4: Creation

C. Michael Holloway
c.michael.holloway@nasa.gov

Senior Research Computer Engineer
Safety-Critical Avionics Systems Branch
NASA Langley Research Center, Hampton, Virginia, U.S.A.



UNDERSTANDING ASSURANCE CASES

MODULE 4: CREATION

(expect a major revision in late 2020 or early 2021)

C. MICHAEL HOLLOWAY

NASA LANGLEY RESEARCH CENTER

C.MICHAEL.HOLLOWAY@NASA.GOV

We can't all, and some of us don't. That's all there is to it. - Eeyore (A. A. Milne)

VERSION 2.0

2020-07-15

This material was created in 2015-16, as part of the Explicate '78 project. The project was supported in substantial part by the Assurance Case Applicability to Digital Systems task under the reimbursable interagency agreement with the Federal Aviation Administration for Design, Verification, and Validation of Advanced Digital Airborne Systems Technology (IAI-1073 Annex 2 for NASA; DTFACT-10-X0008, Modification 0004 for the FAA). The original presentations were delivered to a selected group of FAA civil servants and NASA Langley personnel. The audio was recorded and partial transcripts (containing only the words spoken by the presenter, Mr. Holloway) produced. The intent from the beginning was to collect the material into a form that could be made available publicly. The text adheres closely to the original transcript, except where changes have been made to the original presentation since it was first given, as part of work for for NASA IA-303333/FAA IA NO 692M15-19-T-00029 Annex 1/TO 1. The full collection consists of six documents (including this one), which are available electronically through `https://shemesh.larc.nasa.gov/arg/uac.html`.

(I expect to make a major revision to this Module in late 2020 or early 2021)

Greetings.

Welcome to the fourth and penultimate module in an educational series about Understanding Assurance Cases. In this module, we will examine the ***Creation*** of assurance cases.

If you have not already completed Modules 1 - 3 (Foundation, Application, and Evaluation respectively), please stop reading this document, and complete, at least, Foundation and Application before continuing[1].

I'm quite sure that A. A. Milne *did not* have creating assurance cases in mind when he had Eeyore say "We can't all, and some of us don't. That's all there is to it." [Milne, A. A. 1928. *Winnie the Pooh*. London: Methuen & Co, Ltd.] But it's apt none-the-less. Creating cogent assurance cases is not something that everyone can do. Perhaps only a few of you will ever try to create a real case, but knowing a bit about what goes into such an endeavor may be useful for you nonetheless.

As with all the modules, feel free to interrupt me at *any* point if you have a burning question. I reserve the right to defer the answer to later on that's appropriate, but otherwise I'll do my best to answer it. As with the other modules, there will be times when I'll ask you questions, too. Like now.

[Question to participants: Does anyone have any questions or comments that you want to make before we proceed further?]

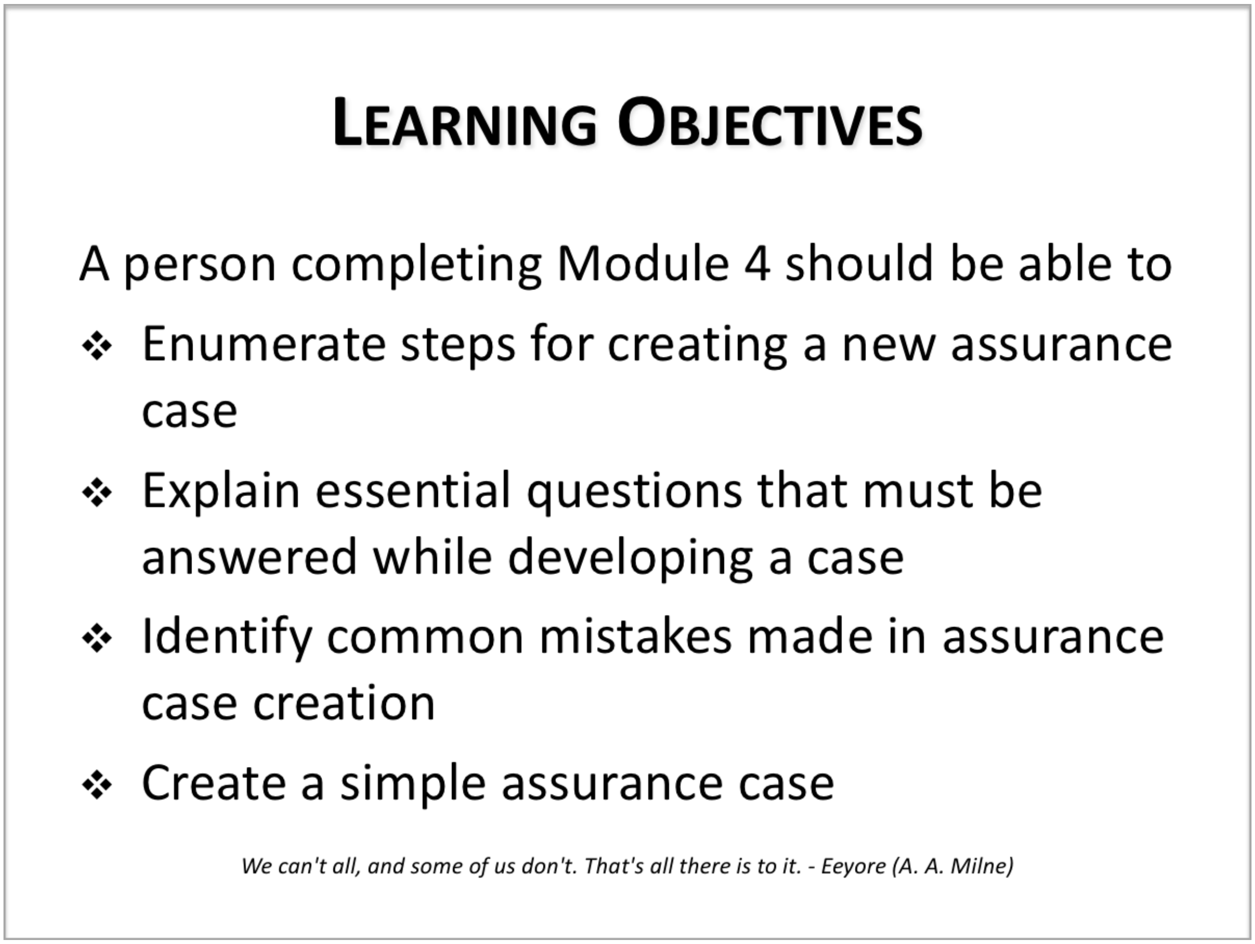
Let's list our learning objectives. By the time we're finished today, I hope that you'll be able to do at least these four things:

- Enumerate steps for creating a new assurance case.
- Explain essential questions that must be answered while developing a case.
- Identify common mistakes made in assurance case creation.
- Create a simple assurance case.

As I'm sure you realize, when we're done with this module, you're not going to be an expert in creating assurance cases (unless you're one already), but you should have a little better acquaintance with what's involved in creating them.

We're only going to be able to scratch the surface, But I will provide you with a homework exercise that, if you choose to do it, will help you scratch a bit deeper.

[1] Just in case someone does not follow the suggestion, and thus misses the preliminary information first expounded in Module 1 and repeated verbatim in Module 2, here is that information in simplified form: Within the assurance case community, intramural debates abound about a variety of topics we will discuss. Except in rare instances the existence of these debates is intentionally ignored or mentioned only briefly in this material. (See Module 1 or 2 for an explanation of why). Also, all images you see were either created by me (Michael Holloway) or are in the public domain via CC0 1.0 Universal.



Learning Objectives

A person completing Module 4 should be able to

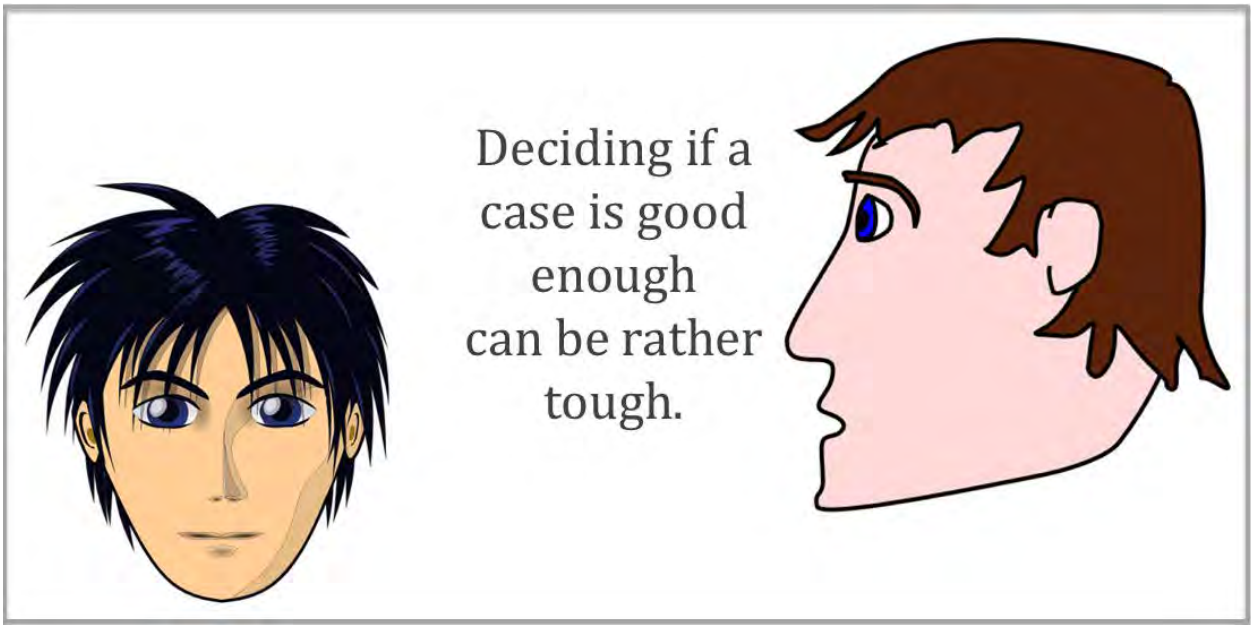
- Enumerate steps for creating a new assurance case
- Explain essential questions that must be answered while developing a case
- Identify common mistakes made in assurance case creation
- Create a simple assurance case

We can't all, and some of us don't. That's all there is to it. - Eeyore (A. A. Milne)

[Question to participants: Any questions about these learning objectives?]

As you probably expect, we begin with the continuing saga of Jon, Mike, and (the unseen) Tim.

When last we left our friends Mike had just told Jon, "Deciding if a case is good enough can be rather tough."



Jon thinks for a few seconds, then asks “How tough is it to create a case in the first place?”

“Hmmmm,” says Mike. “Good question. I guess it sorta depends.”

“It sorta depends on what?” inquires Jon.

“Lots of things,” says Mike, unhelpfully. But after a brief pause he continues, “… what the case is trying to show … what kind of evidence you have … who you’re trying to convince”

Jon interrupts his dad at this point: “I’m trying to convince you Dad, remember?”

“That you are my son …”

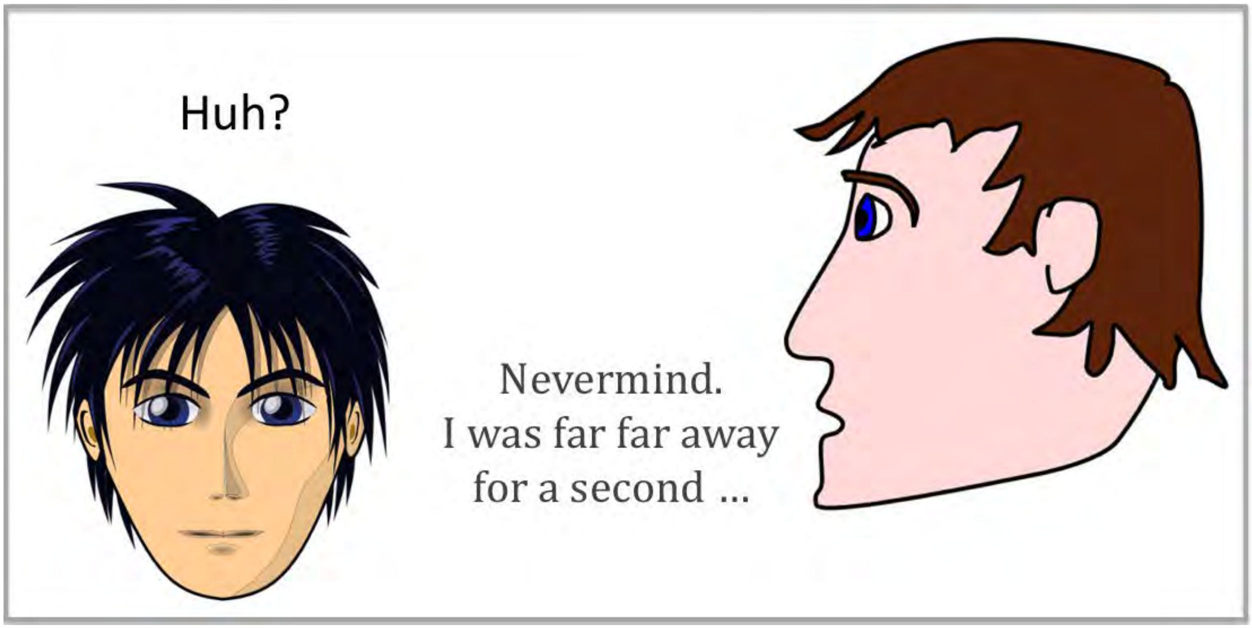
Then after a pause, with a slight grin on his face, Mike continues, “’Tis probably best to just give up now.”

Jon, not seeing the grin on his dad’s face, exclaims with a slightly annoyed tone, “I don’t wanna give up! Tim’s my only hope for getting to the game!”

Mike, with a bigger grin on his face, replies, “No, there is another.”

“Huh?” asks Jon, failing to recognize the reference.

“Never mind. I was far, far away for a second,” says Mike, continuing his excursion into the Star Wars universe[2].



[2] One of the major supporters of the work that led to these five educational modules was Mike DeWalt. At the time he was the FAA’s Chief Scientist and Technical Advisor (CSTA) for Aircraft Computer Software and a huge fan of the Star Wars saga. This reference was for him. Mike is no longer a CSTA, having retired in 2016. He is, presumably still a Star Wars fan.

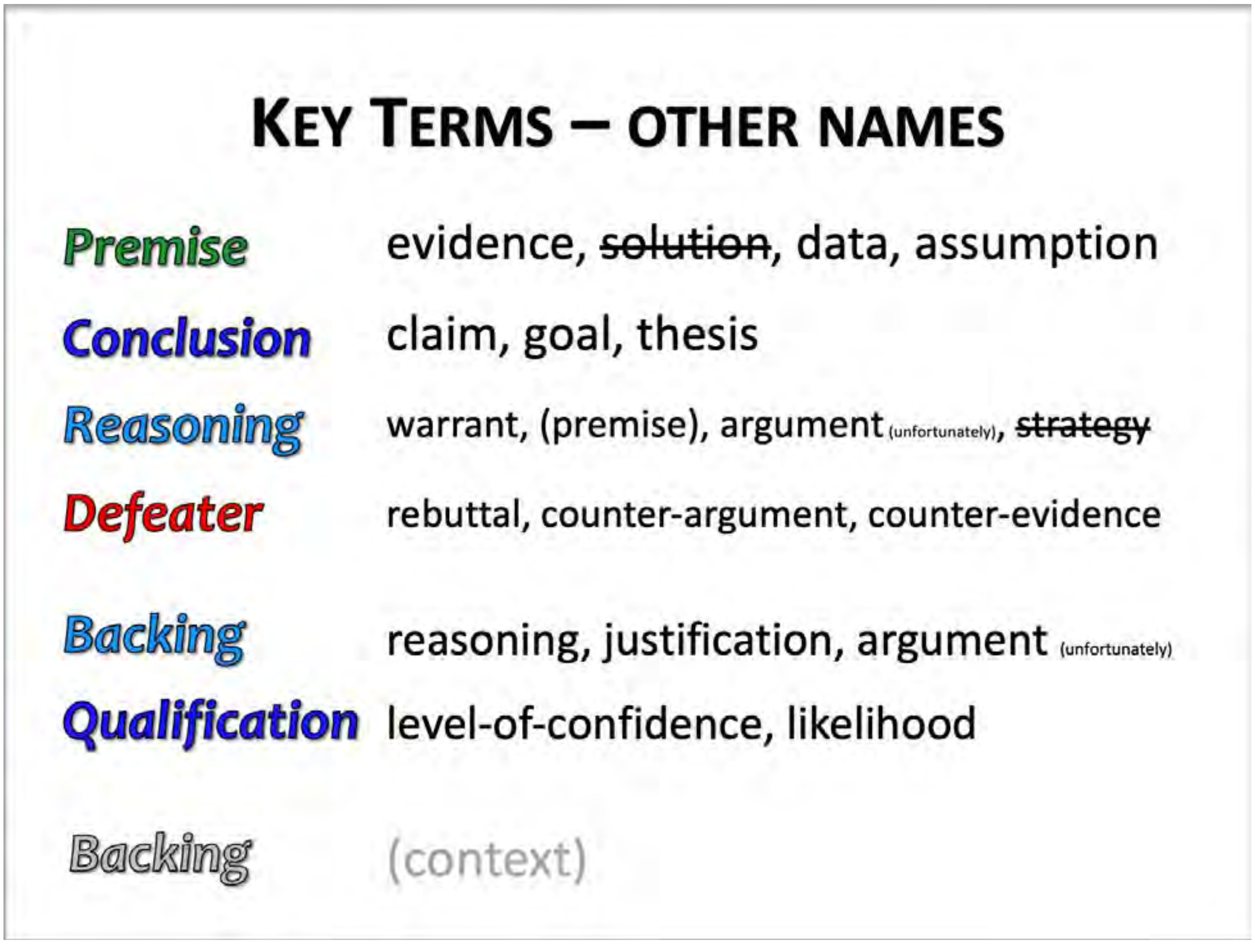
After seeing no hint of recognition on Jon's face, he replies, "I'll show you how to create a case to convince me."

"Thanks Dad!" replies Jon happily.

Despite what some of you may think, I'm not Mike, as my mother reminded anyone who tried to call me that when I was growing up, but I am going to have a go at explaining a bit about creating assurance cases.

Because it may have been a while since some of you completed the last module, I think it's probably a good idea to briefly review argument terms.

You see here a slide that we first saw in Module 1. (Changes will be made here soon.)



KEY TERMS – OTHER NAMES

Premise	evidence, ~~solution~~, data, assumption
Conclusion	claim, goal, thesis
Reasoning	warrant, (premise), argument (unfortunately), ~~strategy~~
Defeater	rebuttal, counter-argument, counter-evidence
Backing	reasoning, justification, argument (unfortunately)
Qualification	level-of-confidence, likelihood
Backing	(context)

On the left side are the terms that we're using in this course: *premise, conclusion, reasoning, defeater, backing (*incorporated in reasoning*)*, *qualification*, and *binding*.

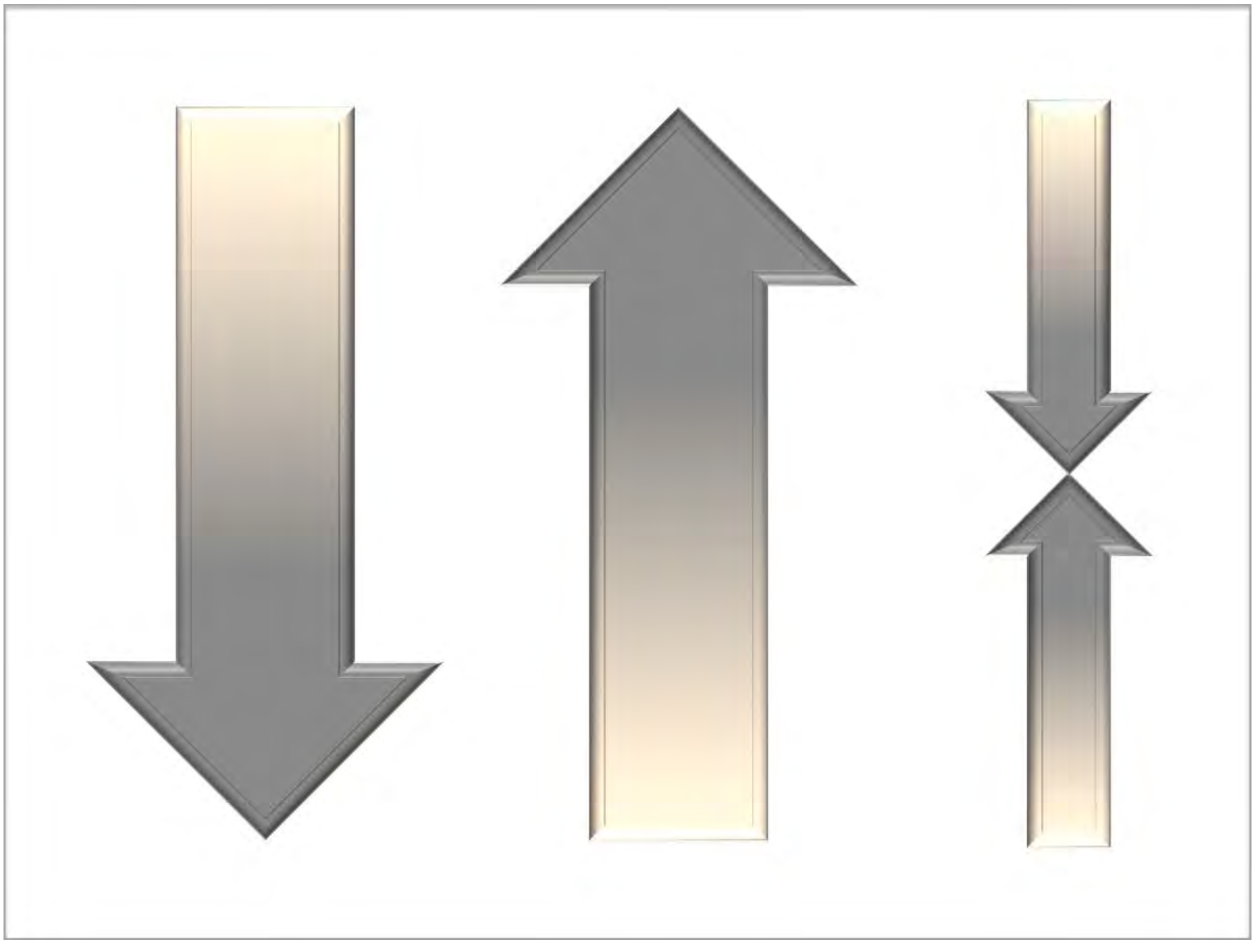
The right side lists some popular alternative terms.

As I've mentioned before, within the assurance case community, the most common terms tend to be *evidence* (instead of *premise*), *claim* or *goal* (instead of *conclusion*) and *argument* (instead of *reasoning*).

I've explained before why I prefer our terms to those, and won't got back over my arguments, unless someone asks me to do so[3].

[Question to participants: Any questions about terms?]

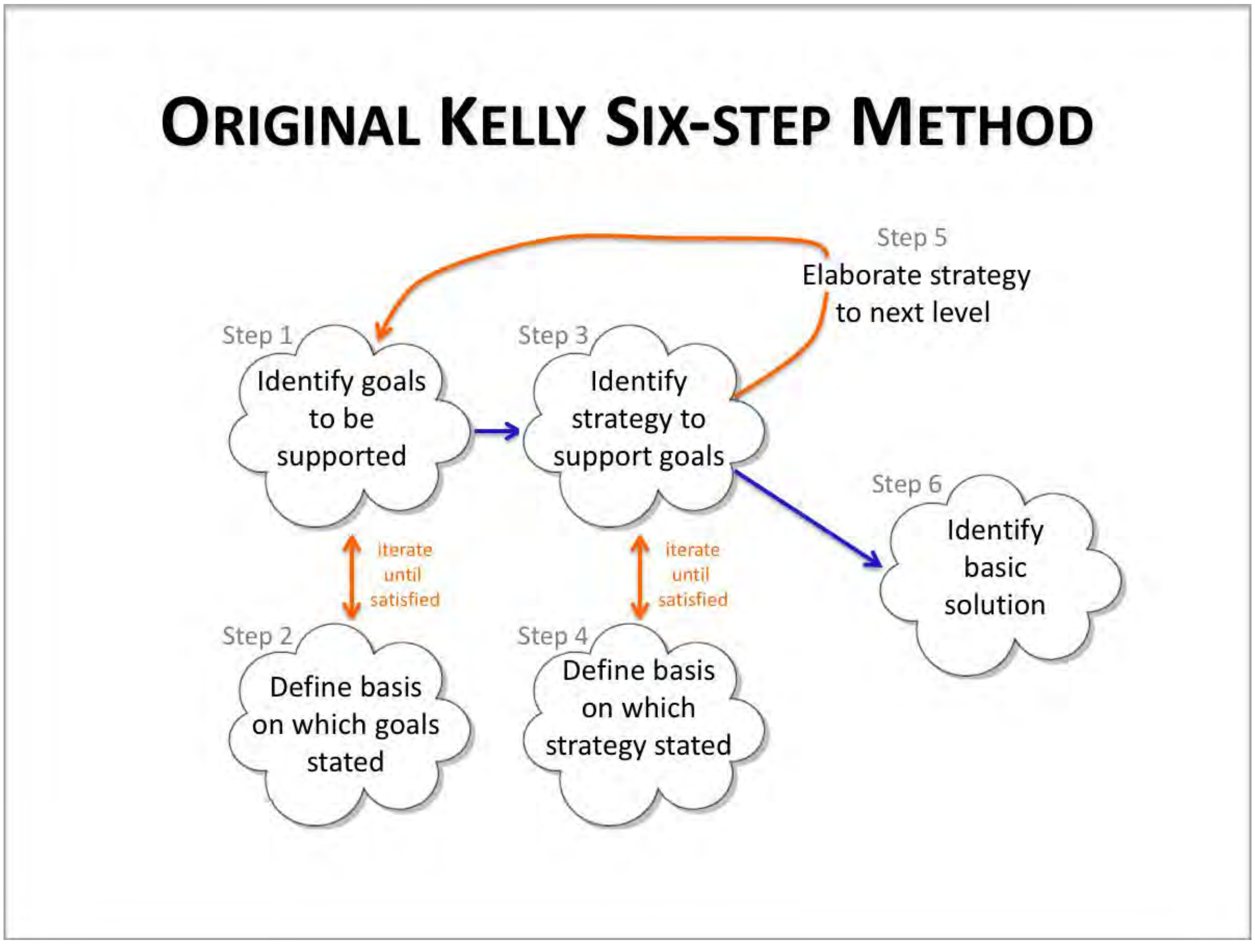
On to talking specifically about assurance case creation.



As you might imagine, in creating an assurance case, one might choose to proceed from the top down, or from the bottom up, or (as is most common) use a combination of the two. For pedagogical purposes, looking at idealized versions of a top down approach and a bottom up approach seems the most helpful. We will start with a top down approach.

In his doctoral thesis in 1998, Tim Kelly from the University of York proposed a six step method for creating safety cases using the Goal Structuring Notation. This slide, derived from a figure in the GSN Community Standard, illustrates that method, using the GSN terminology.

[3] Folks who are reading the material instead of seeing it being presented may look to pages 21-22 in Module 1.



ORIGINAL KELLY SIX-STEP METHOD

Step 1
Identify goals to be supported

Step 2
Define basis on which goals stated

iterate until satisfied

Step 3
Identify strategy to support goals

Step 4
Define basis on which strategy stated

iterate until satisfied

Step 5
Elaborate strategy to next level

Step 6
Identify basic solution

In the years since 1998, other top down approaches have been proposed. But most of them are really nothing more than variations on the six step method, and no evidence has been produced to suggest any of the variations are definitely better, so, we'll follow this approach, 'though rewording it to correspond to the terminology that I prefer.

Step 1: Identify conclusions to be supported.

Step 2: Define basis on which conclusions stated.

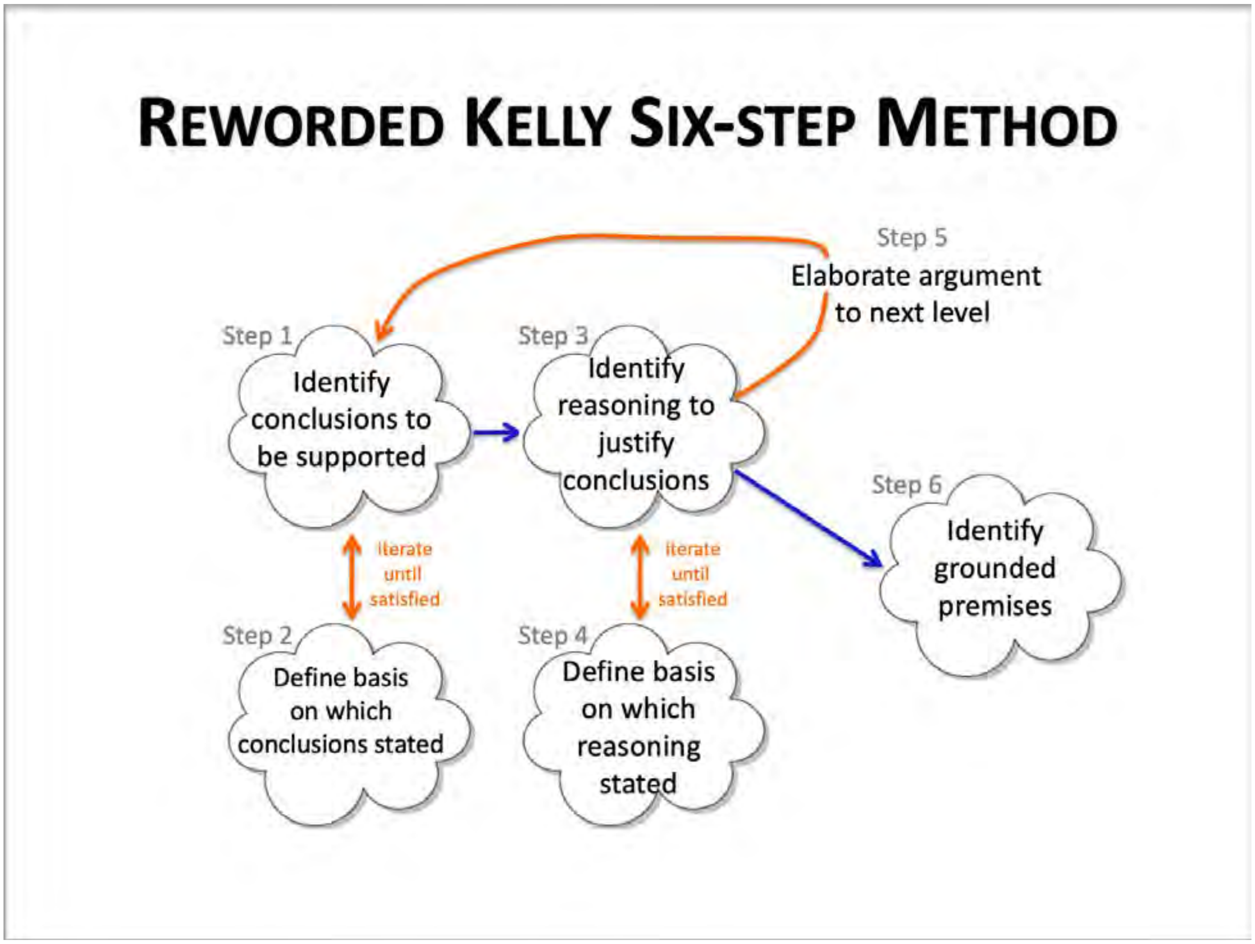
Step 3: Identify reasoning to justify conclusions.

Step 4: Define basis on which reasoning stated.

Step 5: Elaborate argument to next level.

Step 6: Identify grounded premises.

Here is the figure modified with the different (aka better) terminology.



REWORDED KELLY SIX-STEP METHOD

Step 1
Identify conclusions to be supported

Step 2
Define basis on which conclusions stated

iterate until satisfied

Step 3
Identify reasoning to justify conclusions

Step 4
Define basis on which reasoning stated

iterate until satisfied

Step 5
Elaborate argument to next level

Step 6
Identify grounded premises

Let's see what each of these steps means, and how they relate to one another by way of an example. Because Jon seems like such a decent kid, let's use his situation as the basis for the example.

Recall Jon wants Tim to take him to a game. Jon's dad, Mike, doesn't know Tim, and wants assurance that Tim is a safe driver. He's asked Jon & Tim to build an assurance case.

What do you think an appropriate top-level *conclusion* (or goal or claim if you must) is for such a case?

Please do not turn the page until you have an answer to the question.

I suggest the following: “Tim is a safe enough driver to take Jon to the game.”

That’s step one: identifying the *conclusion* to be supported.

Perhaps some of you may see some problems (or at least ambiguities) with this statement as the *conclusion*. Handling such problems is the purpose of the next step.

For step 2, we need to define the basis on which the *conclusion* is stated. Or, in other, perhaps slightly clearer words, we need to decide if there’s additional information we need to know in order for our statement of the *conclusion* to make sense.

Any ideas?

Some questions to ask yourself as you formulate your own ideas:

- Are there any words or phrases for which definitions are needed?
- Are any unstated assumptions seemingly present?
- After adding definitions and assumptions, are any changes to the original statement necessary to ensure it is unambiguous?
- And what about Naomi? [4]

A hint: the answer to each of the first three questions is, “Yes.”

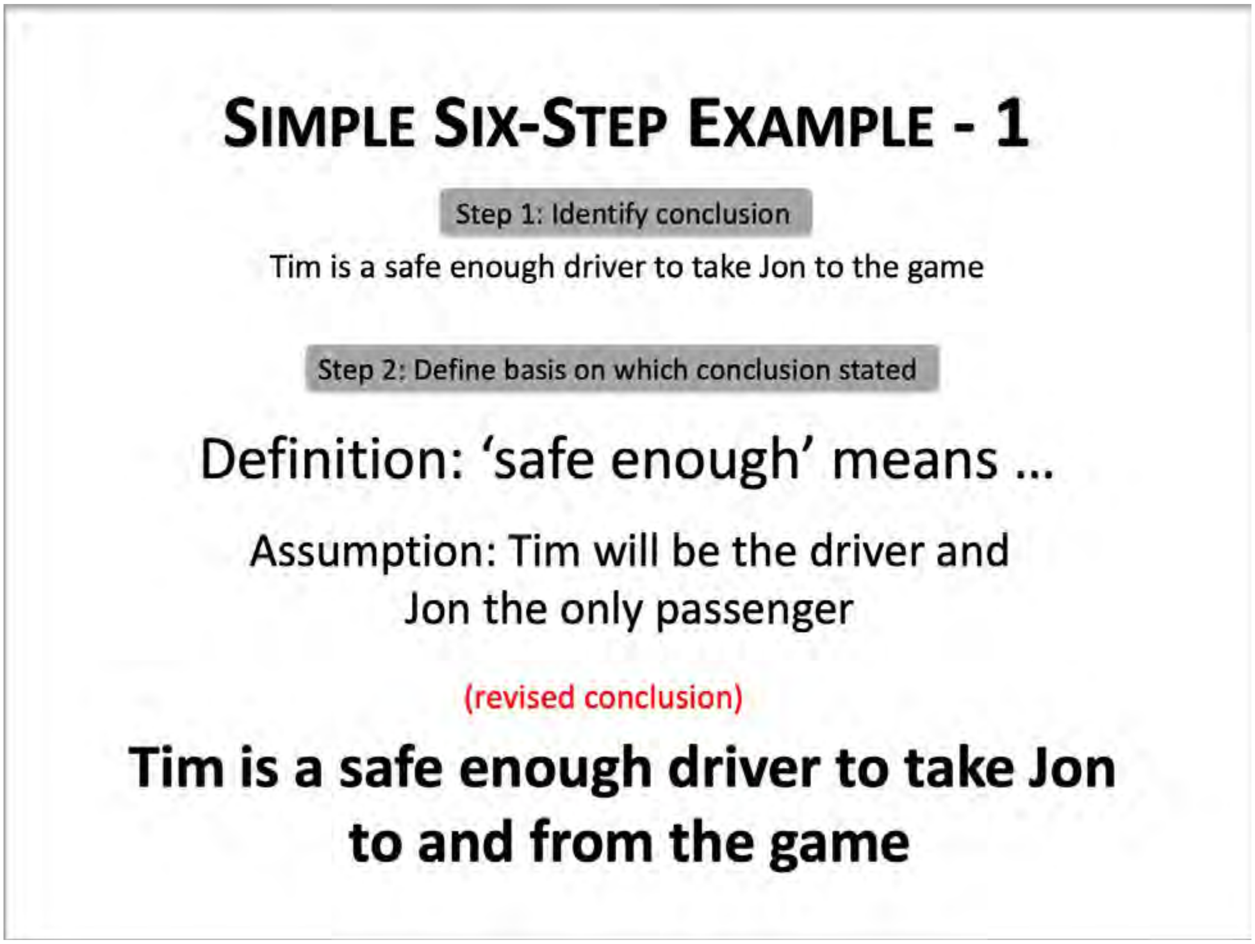
Another hint: While thinking about what changes to the original statement may be necessary to ensure it is unambiguous, complete the following quotation from President Kennedy’s announcement of the goal of going to the moon:

“I believe that this nation should commit itself to achieving the goal, before this decade is out, of landing a man on the moon ….”

Please do not turn the page until you have your own ideas.

[4] Folks who are reading the material instead of seeing it being presented, and who are confused by this question should refer to page 6 of Module 1.

Here are my answers.



SIMPLE SIX-STEP EXAMPLE - 1

Step 1: Identify conclusion

Tim is a safe enough driver to take Jon to the game

Step 2: Define basis on which conclusion stated

Definition: 'safe enough' means …

Assumption: Tim will be the driver and
Jon the only passenger

(revised conclusion)

Tim is a safe enough driver to take Jon to and from the game

One thing we certainly need is to know is the meaning of the phrase 'safe enough'.

[Question to participants: What do you think might be an appropriate definition?]

There are a variety of options, but perhaps "posing no greater risk to Jon than Mike would …" would be a good one.

It seems to me that we might need to make at least one assumption, something along the lines of "Tim will be the driver and Jon the only passenger."

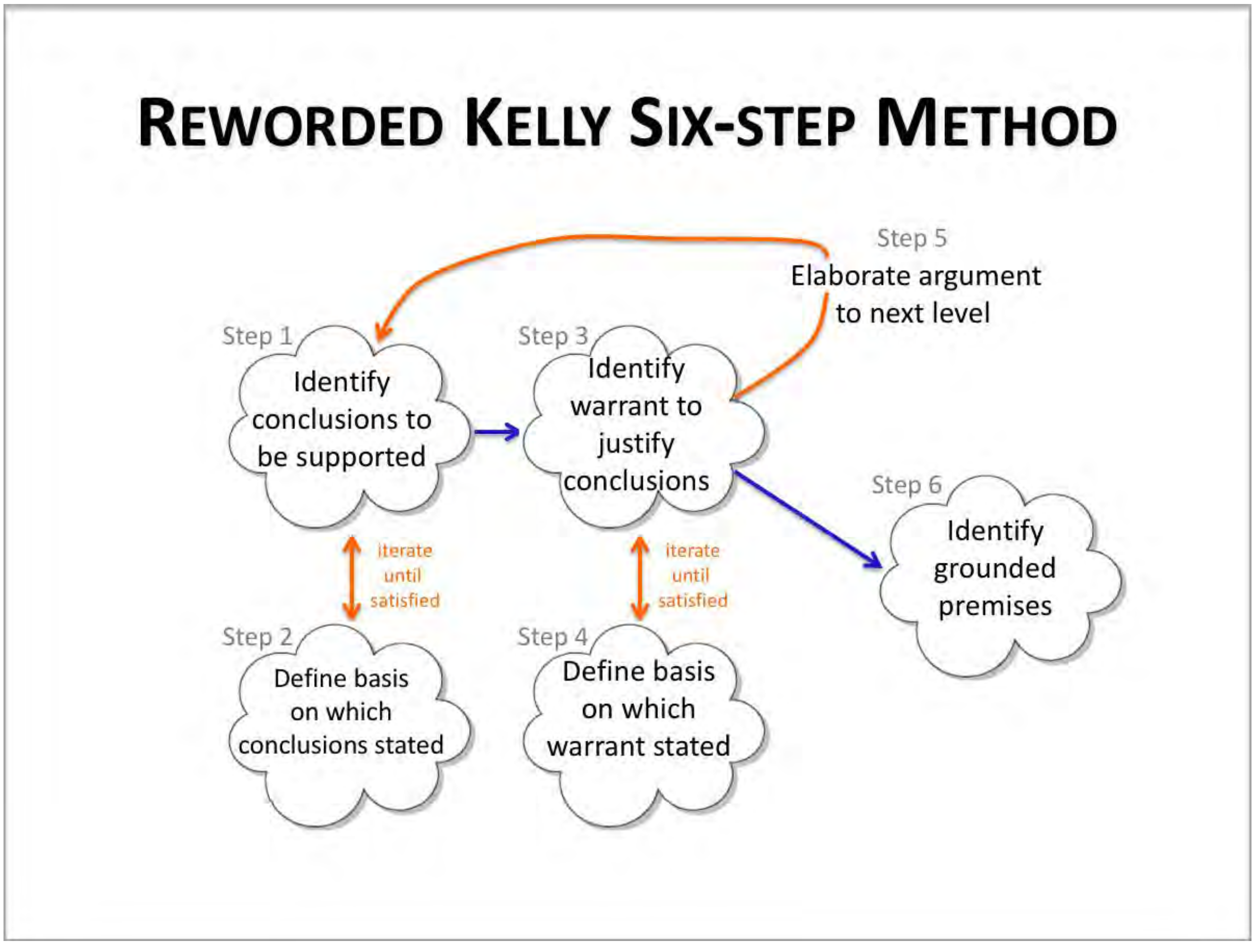
In writing the original conclusion, I was thinking of 'to the game' as being equivalent to 'to and from the game'; meaning it isn't okay for Tim to just get Jon safely to the game, but he also needs to get Jon back home afterwards. To avoid possible ambiguity, perhaps the conclusion ought to be as "Tim is a safe enough driver to take Jon to and from the game."[5]

[5] The rest of JFK's statement was, "… and returning him safely to the Earth." Landing wasn't enough; returning safely to Earth was equally as important.

We have now completed steps 1 and 2 for our simple example.

[Question to participants: What questions or comments do you have at this point?]

Returning to the graphical illustration of the method, you see that steps 3 & 4 are similar to steps 1 & 2 but applied to the reasoning instead of to the conclusion. Step 5 involves elaborating the argument to identify premises for the top level conclusion, which will likely be conclusions that need to be supported themselves.



REWORDED KELLY SIX-STEP METHOD

Step 1
Identify conclusions to be supported

Step 2
Define basis on which conclusions stated

iterate until satisfied

Step 3
Identify warrant to justify conclusions

Step 4
Define basis on which warrant stated

iterate until satisfied

Step 5
Elaborate argument to next level

Step 6
Identify grounded premises

So, what we want to do next is think about the sort of reasoning that we'd want to use to establish the conclusion that "Tim is a safe enough driver to take Jon to and from the game."

[Question to participants: Does anyone want to suggest possible reasoning?]

If you're having trouble thinking of the reasoning, try instead to think about the sorts of premises that you think you'd want to see for the conclusion (skipping mentally to Step 5). Then think about the reason those premises would give you confidence in the conclusion.

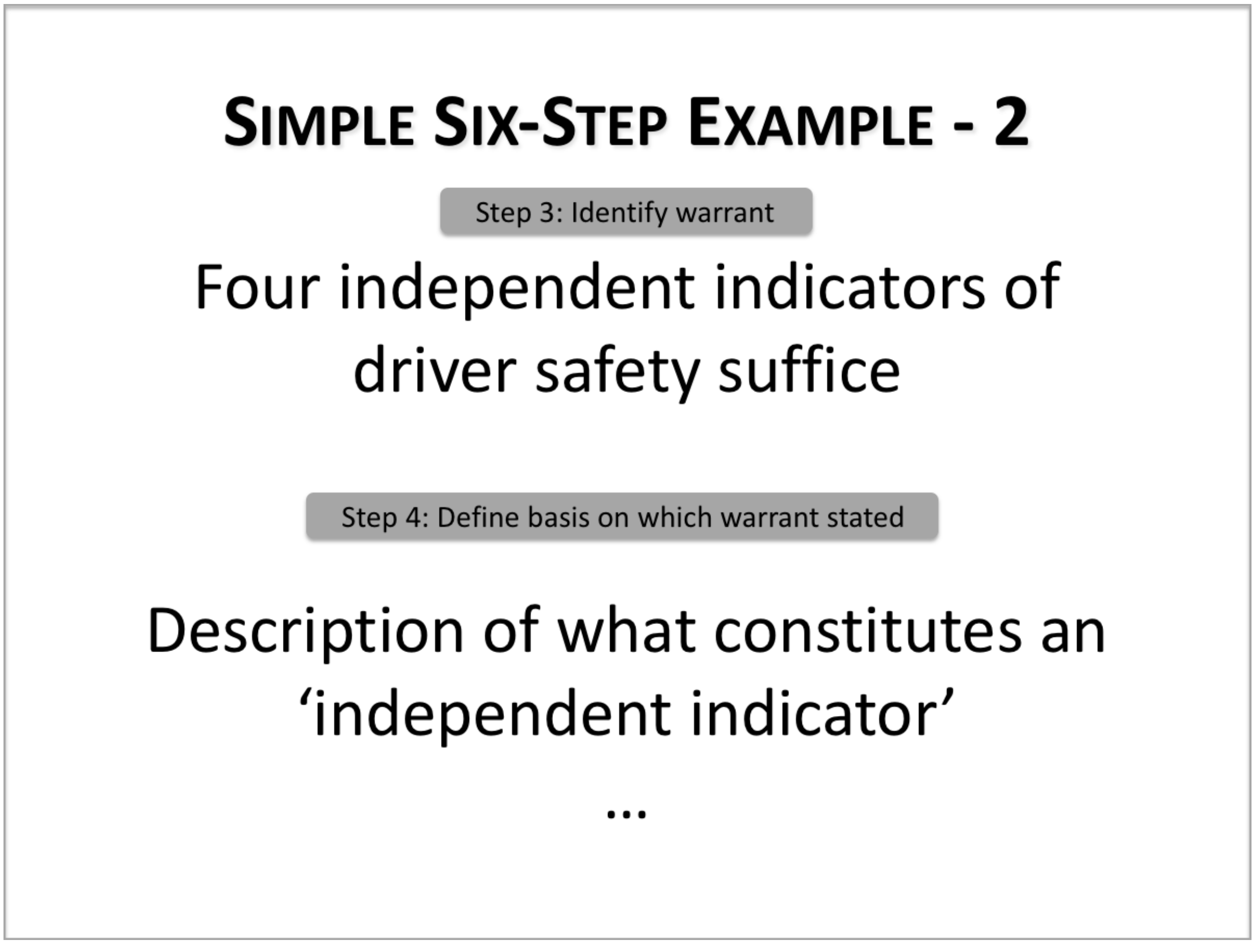
The 6-step method isn't intended to be a straightjacket that restricts your thinking into a strictly sequential order. It is really just a guideline to help prompt your thinking. Often considering Steps 3, 4, & 5 together may be the most useful approach to creating a case.

There plenty of different possibilities for plausible and sufficient reasoning. For the purposes of continuing the example, I will suggest something mundane.

Reasoning: “Four independent indicators of driver safety suffice.”

[Question to participants: What do we need to know for this reasoning to make sense?]

Well, at the very least we’d need to have a common understanding of what constitutes an ‘independent indicator’. For the purposes of the example, let us assume that we have completed Steps 3 and 4.



SIMPLE SIX-STEP EXAMPLE - 2

Step 3: Identify warrant

Four independent indicators of driver safety suffice

Step 4: Define basis on which warrant stated

Description of what constitutes an ‘independent indicator’

...

Let’s proceed to elaborating the argument (Step 5). We will do so by considering what might constitute the collection of acceptable independent indicators of driver safety.

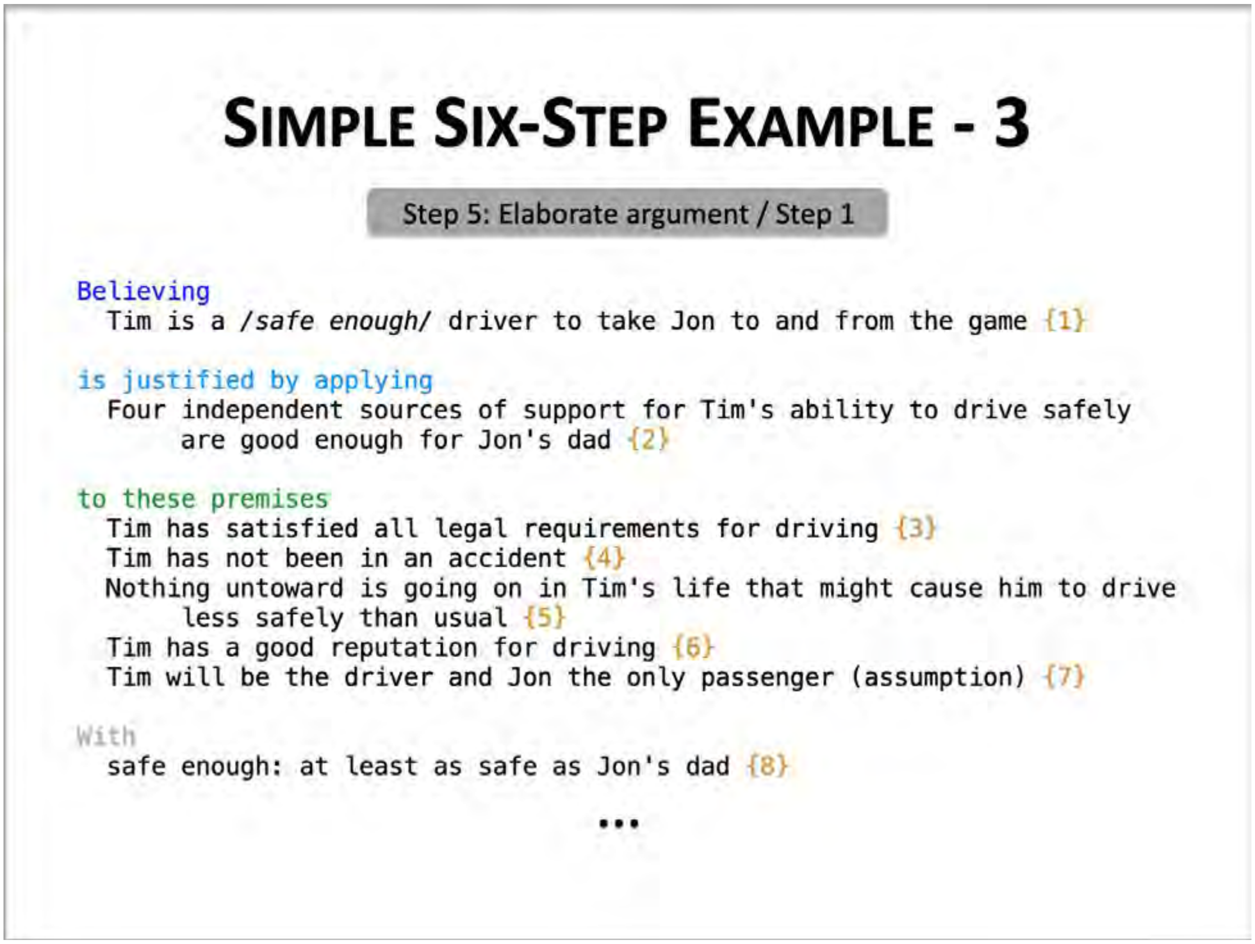
[Question to participants: Are you able to name some indicators?]

Please do not turn the page until you have thought of at least one.

Here are the four that I decided to write down:

1. Tim has satisfied all legal requirements for driving.
2. Tim has not been in an accident.
3. Tim has a reputation for driving safely.
4. Nothing is going on in Tim's life that might cause him to drive less safely than usual.

Of course, many more plausible possibilities exist, but this slide expresses what we've just discussed in FAN.



SIMPLE SIX-STEP EXAMPLE - 3

Step 5: Elaborate argument / Step 1

```
Believing
  Tim is a /safe enough/ driver to take Jon to and from the game {1}

is justified by applying
  Four independent sources of support for Tim's ability to drive safely
       are good enough for Jon's dad {2}

to these premises
  Tim has satisfied all legal requirements for driving {3}
  Tim has not been in an accident {4}
  Nothing untoward is going on in Tim's life that might cause him to drive
       less safely than usual {5}
  Tim has a good reputation for driving {6}
  Tim will be the driver and Jon the only passenger (assumption) {7}

With
  safe enough: at least as safe as Jon's dad {8}

                              •••
```

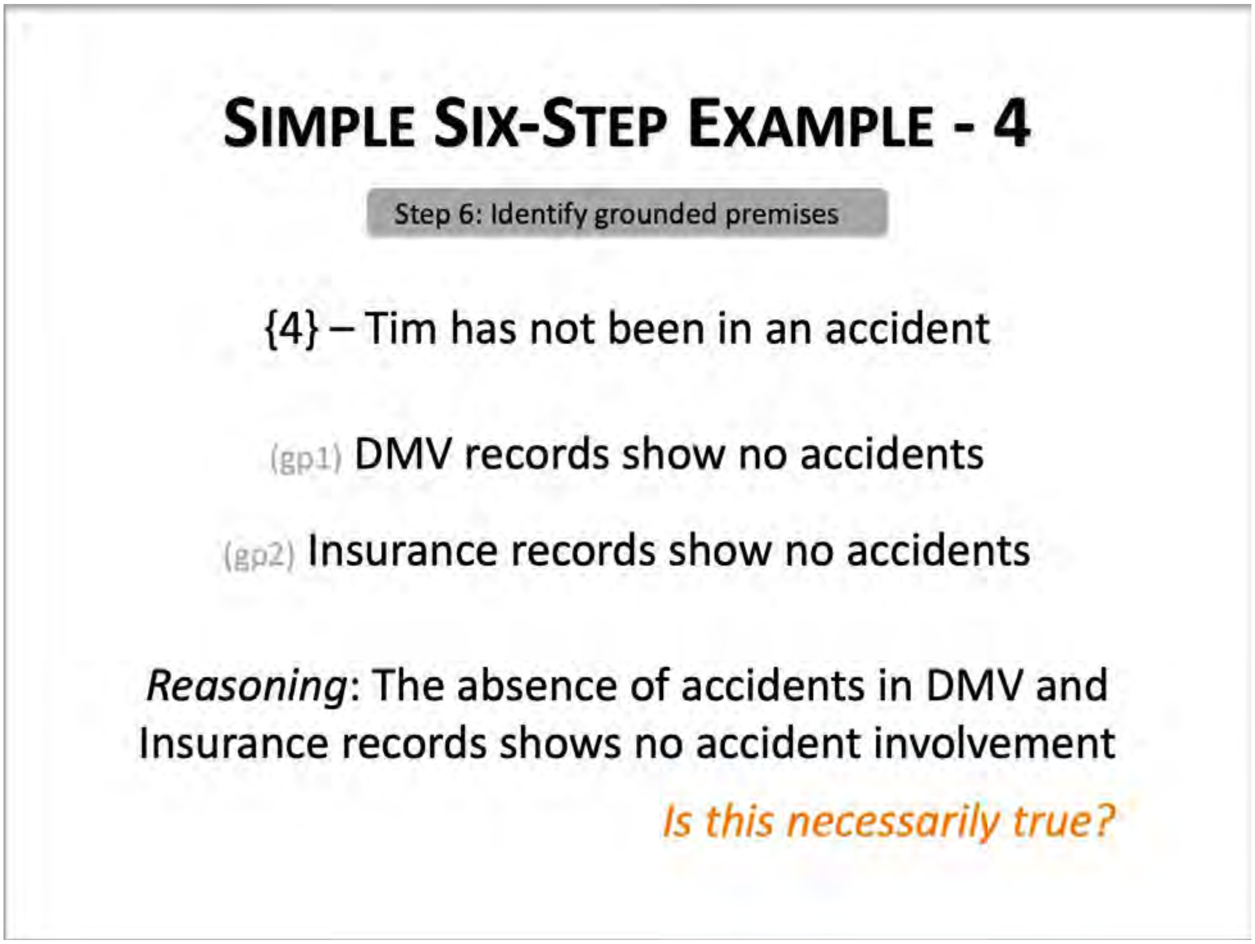
[Question to participants: Does that make sense? What questions do you have?]

Let's now think about grounded premises (or evidence if you prefer) for only one of these: "Tim has not been in an accident."

What might be facts or data that establish that Tim has not been in an accident?

Please do not turn the page until you have thought of at least one.

Here are two possible grounded premises: “DMV records show no accidents,” and “Insurance records show no accidents”



Simple Six-Step Example - 4

Step 6: Identify grounded premises

{4} – Tim has not been in an accident

(gp1) DMV records show no accidents

(gp2) Insurance records show no accidents

Reasoning: The absence of accidents in DMV and Insurance records shows no accident involvement

Is this necessarily true?

A reason why these two premises would be sufficient might be, “The absence of accidents in DMV and Insurance records shows no accident involvement.”

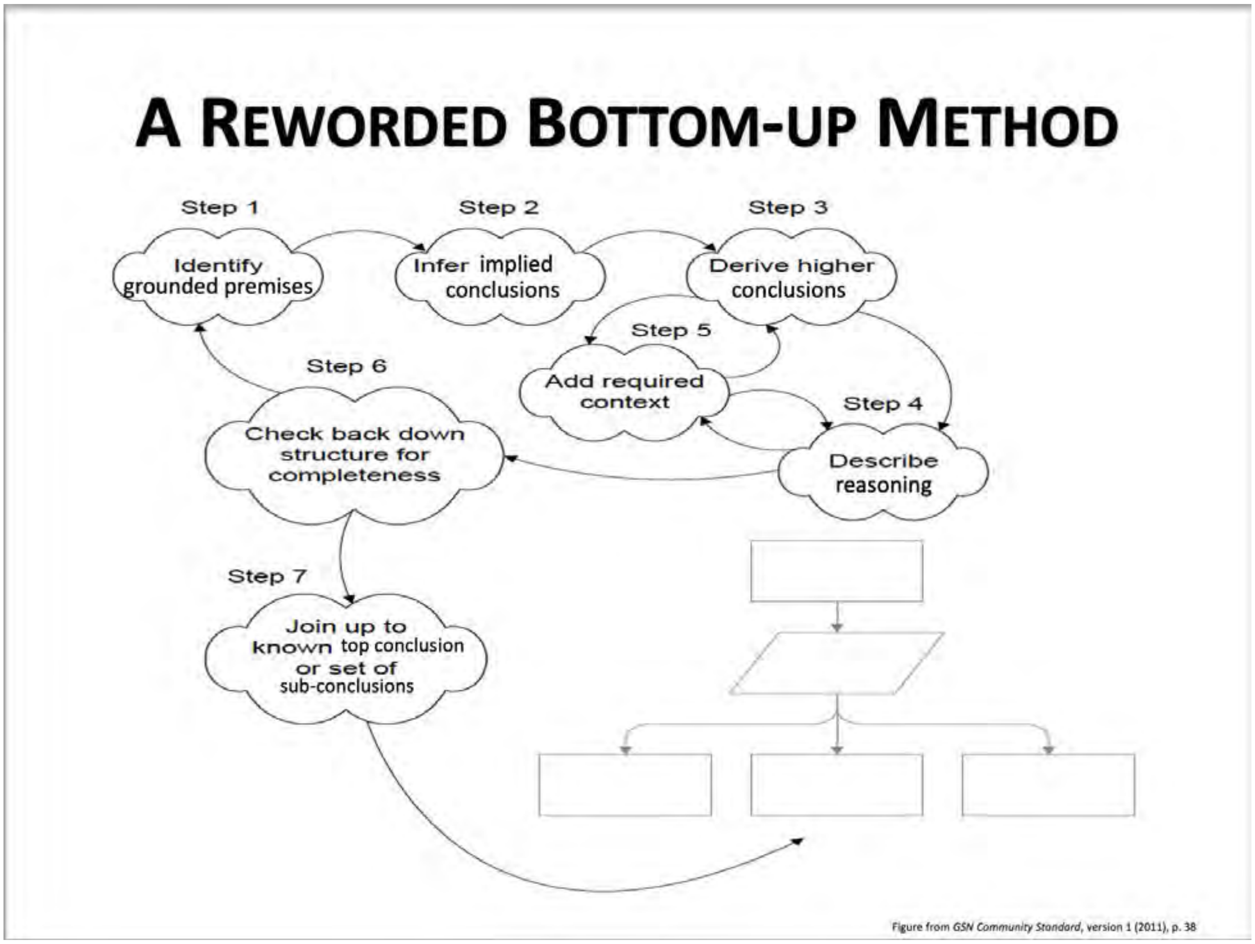
But is this *necessarily* true? Will it always be the case that the reasoning holds? That is, whenever DMV and Insurance records for a person contain no accidents, is it always true that the person has lived an accident-free driving life?

No … because the person, Tim in our example, could’ve had an unreported accident, or perhaps even several.

Some doubt will therefore exist as to whether we’ve fully established Tim’s accident-freedom. Hence, a reason we chose multiple independent indicators in the first place: no one of them alone provides sufficient confidence, but perhaps the combination of all four does justify the confidence. To complete the case, we’d continue in a similar fashion with each of the 3 other independent indicators, deciding what’s necessary to establish confidence that they are true. If we are unable to create an argument (or arguments) to provide sufficient confidence, then we will have to admit our efforts have failed to justify allowing Tim to take Jon to and from the game[6].

[6] I know several Tims. For one of those fellows, no convincing assurance case could ever be created for allowing one’s child in a car with that Tim behind the wheel.

Let’s look now at a primarily bottom-up method for creating assurance cases. It, too, was originally developed for GSN-style cases, but more recently than the method we just examined. I’ll skip showing you the version using GSN terminology[7], and move directly to one using our (better) terminology.



A REWORDED BOTTOM-UP METHOD

Step 1
Identify grounded premises
Step 2
Infer implied conclusions
Step 3
Derive higher conclusions
Step 5
Add required context
Step 4
Describe reasoning
Step 6
Check back down structure for completeness
Step 7
Join up to known top conclusion or set of sub-conclusions

Figure from *GSN Community Standard*, version 1 (2011), p. 38

You start with grounded premises, think about what they allow you to conclude, and why, and the needed context, and continue upwards. I’m not going to go through a full example, but let’s think about this approach a little bit.

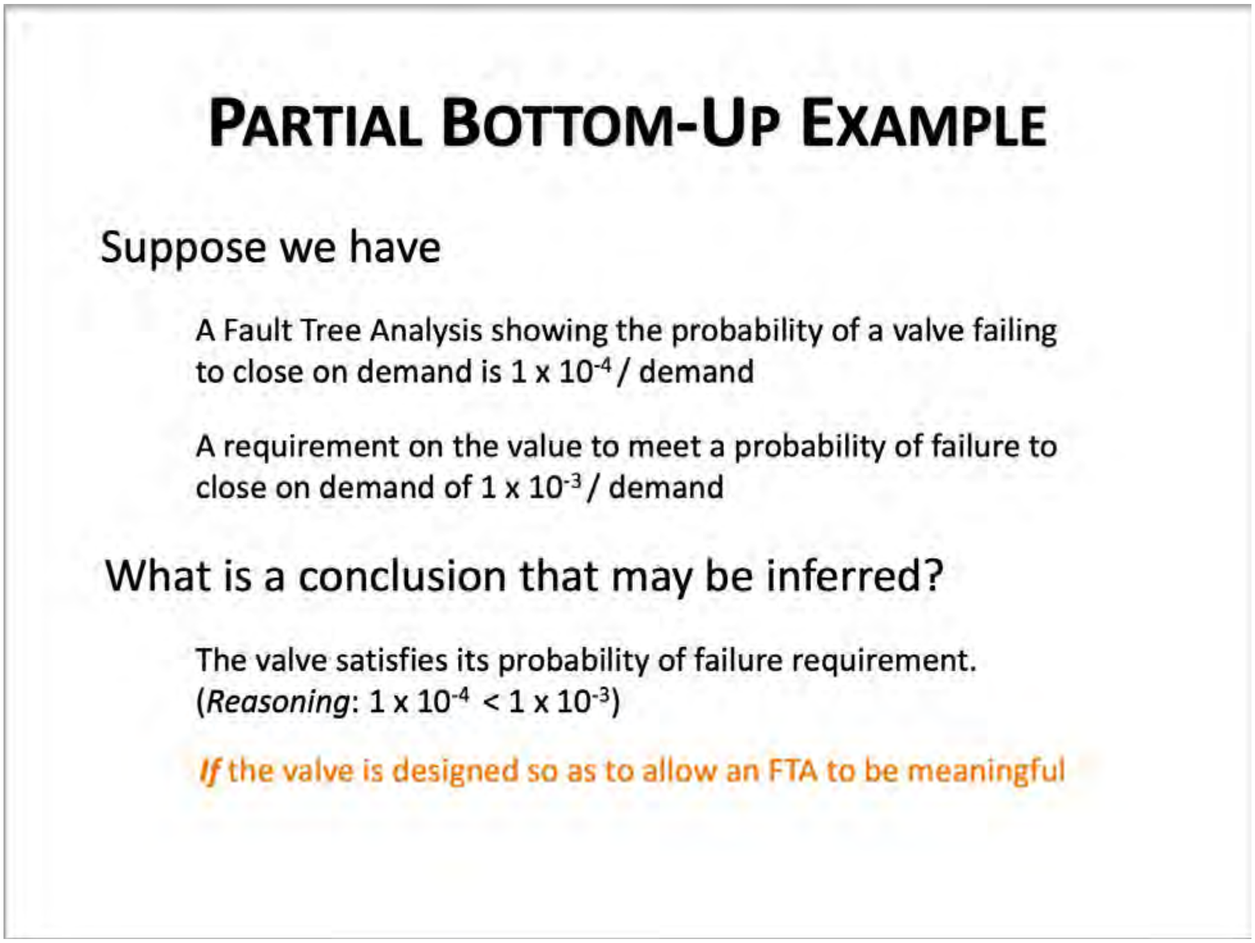
Suppose we have these two facts:

- A Fault Tree Analysis showing the probability of a valve failing to close on demand is 1 x 10-4 / demand
- A requirement on the value to meet a probability of failure to close on demand of 1 x 10-3 / demand.

What’s a conclusion that we can infer?

Please do not turn the page until you have an answer.

[7] The figure you see here is based on a figure that first appeared in *GSN Community Standard*, version 1 (2011). p. 38. Since that time the GSN standard has been updated, but the figure illustrating the bottom-up style is unchanged. [Assurance Case Working Group. 2018. *Goal Structuring Notation Community Standard Version 2*. SCSC-141B. `https://scsc.uk/scsc-141B`]



PARTIAL BOTTOM-UP EXAMPLE

Suppose we have

A Fault Tree Analysis showing the probability of a valve failing to close on demand is 1 x 10^{-4} / demand

A requirement on the value to meet a probability of failure to close on demand of 1 x 10^{-3} / demand

What is a conclusion that may be inferred?

The valve satisfies its probability of failure requirement. (*Reasoning*: 1 x 10^{-4} < 1 x 10^{-3})

If the valve is designed so as to allow an FTA to be meaningful

The valve satisfies its probability of failure requirement with the very simple reasoning: “1 x 10-4 < 1 x 10-3”.

But is this conclusion always justified in any circumstance, or are there conditions or context we need to consider?

At least one thing we need to consider is that the premises and reasoning justify confidence in the conclusion only “If the valve is designed so as to allow an FTA to be meaningful.”

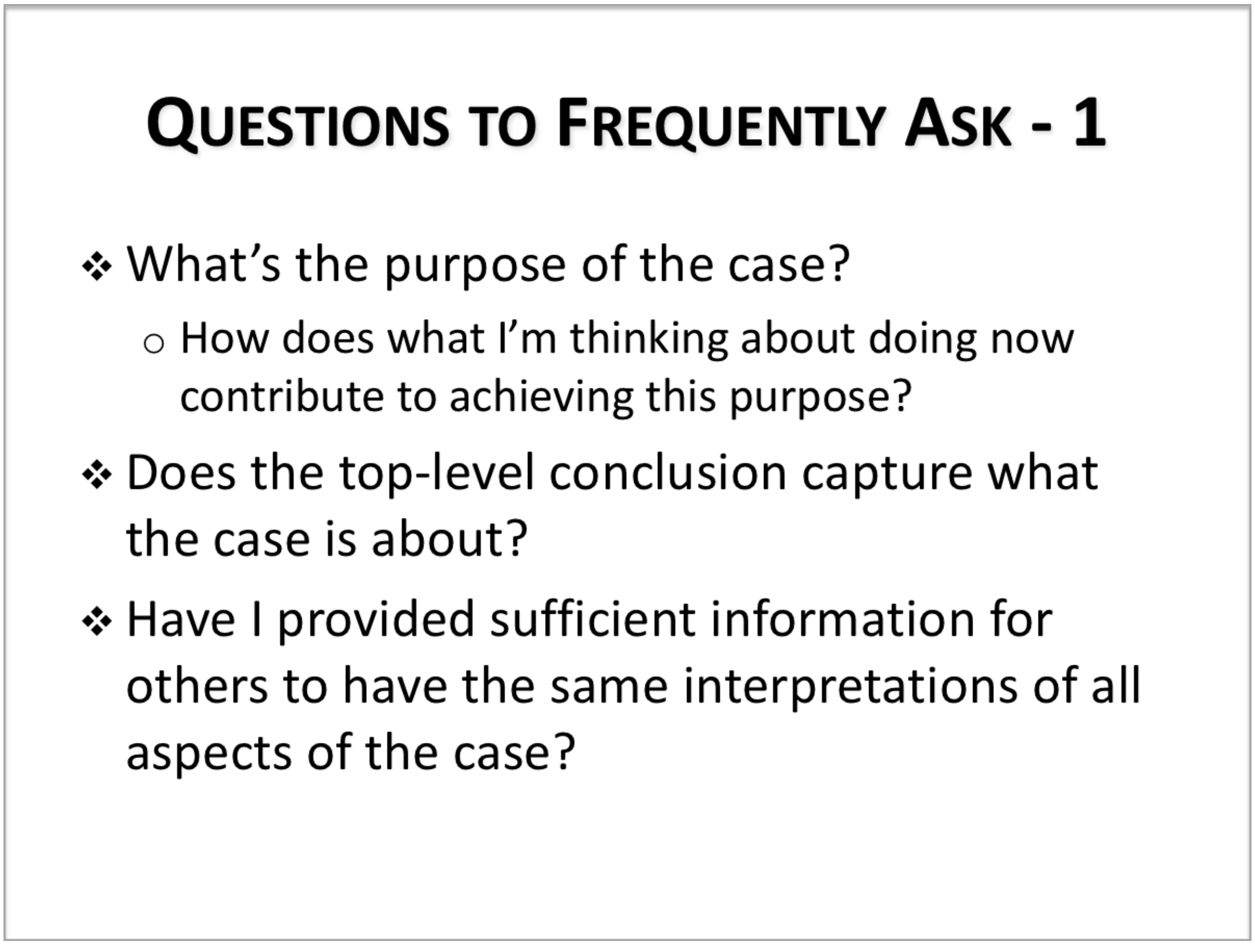
If, however, the valve’s design includes aspects that make FTA untrustworthy (it contains software for example) then we can’t legitimately make the conclusion we suggested.

[Question to participants: Surely you have question and comments at this point. What are they? Note: in the original presentation, this Q&A part lasted for about 15 minutes. People who are reading this material are encouraged to send questions and comments to the author at c.michael.holloway@nasa.gov]

For those of you who are interested in seeing a much bigger example, consider taking a look at the Explicate ’78 work [full report: Holloway, C.M., Graydon, P.J. 2018. *Explicate '78: Assurance Case Applicability to Digital Systems*. DOT/FAA/TC-17/67. https://go.usa.gov/xPEJr. shorter version: Holloway, C.M. 2015. “Explicate '78: Uncovering the Implicit Assurance Case in DO-178C”. *Engineering Systems for Safety. Proceedings of the 23rd Safety-critical Systems Symposium*. M.

Parsons & T. Anderson (eds).] Although slightly different terminology was used for some terms in that report, you should by this time have no difficulty in translating to our better terminology.

Let's move on now to talking about some of the questions that a creator of an assurance case should be often asking her or his self. So, instead of FAQs, we'll be talking about QFAs.



Questions to Frequently Ask - 1

- What's the purpose of the case?
    - How does what I'm thinking about doing now contribute to achieving this purpose?
- Does the top-level conclusion capture what the case is about?
- Have I provided sufficient information for others to have the same interpretations of all aspects of the case?

The first question you need to be frequently asking if you're creating an assurance case is, "What's the purpose of the case?" Also, ask yourself the associated question: "How does what I'm thinking about doing *now* contribute to achieving this purpose?" Your next steps may be different depending on the case's purpose.

Another important question is "Does the top-level conclusion capture what the case is about?" Suppose, for example, the top-level conclusion is solely about safety, but the case is supposed to provide justified confidence not only in safety, but also in achieving intended function; you need to modify the top-level conclusion.

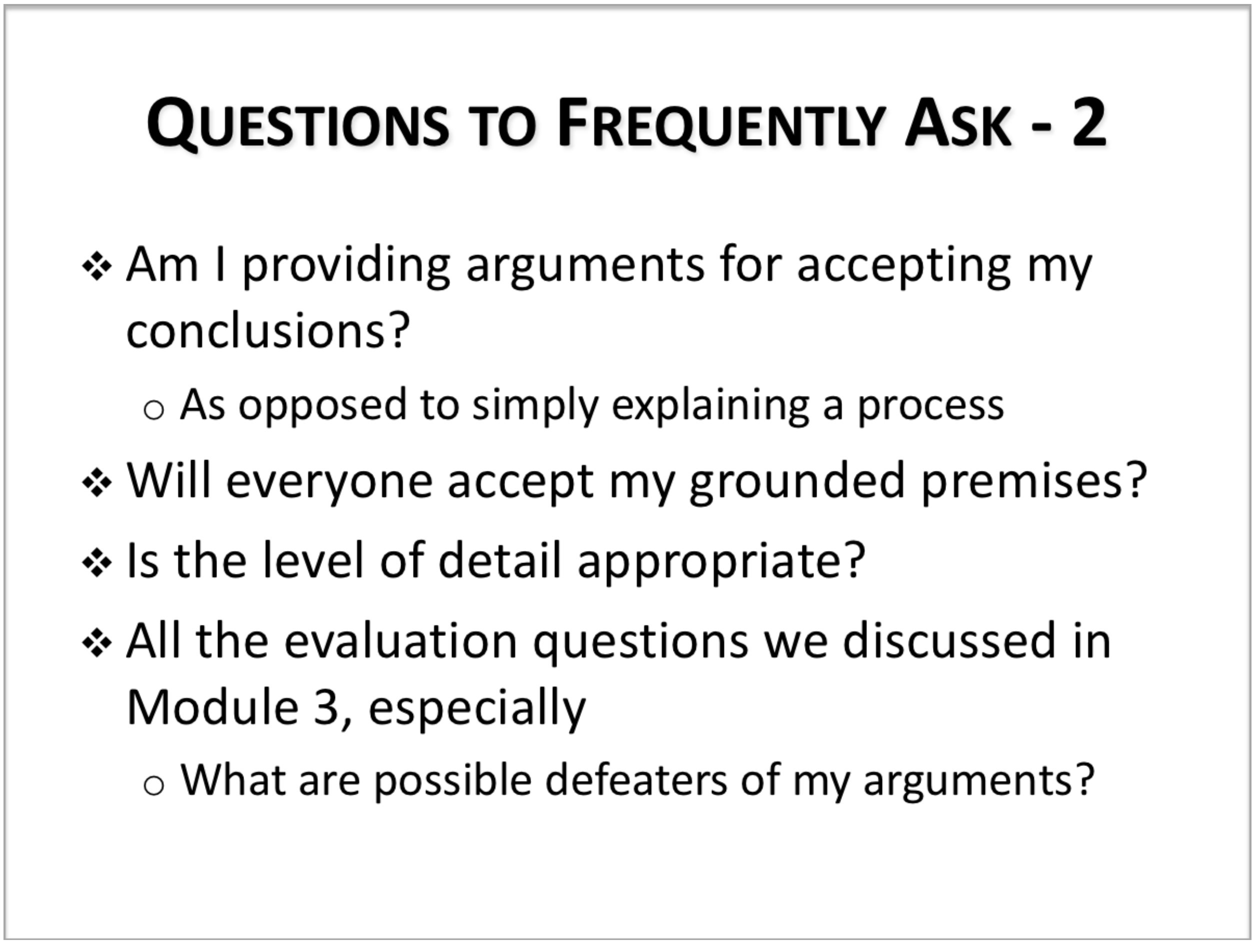
An especially critical question to ask often is the last one shown on this slide: "Have I provided sufficient information for others to have the same interpretations of all aspects of the case?"

Recall my example from a few minutes ago: my use of "to the game" instead of "to and from the game" opened up an opportunity for differing interpretations by different people. Eliminating all such possibilities is not necessarily feasible (because some people insist on imagining impossible interpretations) but striving to eliminate *feasible* alternate interpretations is always the right thing to do.

A brief aside: If you're skeptical about my claim that some people imagine impossible interpretations, then I think a simple example will cause you to give up the skepticism.

DO-178C Chapter 1, section 4, item d notes the "document describes activities for achieving" the objectives, but says explicitly: "The applicant may plan and, subject to the approval of the certification authority, adopt alternate activities to those described in this document." Despite the explicit words, there are some people who insist DO-178C requires that all the activities listed in it must be followed. The words do not allow such an interpretation, but some people imagine they do[8].

[Question to participants: Any questions about these QFA's before we move on to some more?]



Questions to Frequently Ask - 2

- ❖ Am I providing arguments for accepting my conclusions?
  - o As opposed to simply explaining a process
- ❖ Will everyone accept my grounded premises?
- ❖ Is the level of detail appropriate?
- ❖ All the evaluation questions we discussed in Module 3, especially
  - o What are possible defeaters of my arguments?

[8] During the writing of the document, some of us anticipated the possibility that some people would be negatively imaginative when reading the sentence. I suggested quite strongly we should delete the words "subject to the approval of the certification authority." Because the qualification was (and still is) already implicitly applied to *every sentence* in the guidance, writing it out explicitly here was unnecessary. It was also dangerous, because would likely encourage those who wanted to be encouraged to think alternate activities were deprecated. Only handful of others supported my position. Thus, the words remained.

Another important question to keep in mind is this one: “Am I providing arguments for accepting my conclusions as opposed to simply explaining a process?”

It is not uncommon to see an assurance case written by a neophyte looking much more like a simple description of *what was* done than an argument about *why* doing those things is sufficient to establish the truth of the top-level conclusion to an acceptable level of confidence. Typically in such cases, reasoning is missing, or written too poorly to explain the reasoning[9].

Another important question is “Will everyone accept my grounded premises?” (Or if you prefer the term ‘evidence’: “Will everyone accept my evidence?”)

Recall our quantified fault tree analysis example from earlier. If the analysis was applied to a subsystem or component for which obtaining real probability of failure numbers is possible, then citing the FTA results as a grounded premise is appropriate. Everyone should accept it.

But for other subsystems or components, for which probability numbers are fictitious (for example, a subsystem or component containing software), the FTA results should not be accepted. At least not without an additional argument justifying their acceptance for the particular subsystem or component in the case under consideration.

Do not forget: The assurance case argument structure must end with accepted grounded premises. If it does not, more argument is needed.

We’ve talked at several times during the course about this next question: “Is the level of detail appropriate?”[10]

We talked at length in Module 3 about other evaluation questions; all these constitute QFAs, particularly, but not only, the specific question, “What are possible defeaters of my arguments?” I won’t go back over our fairly extensive discussion of defeaters, but will stop at this point for questions or comments about this section on questions to frequently ask.

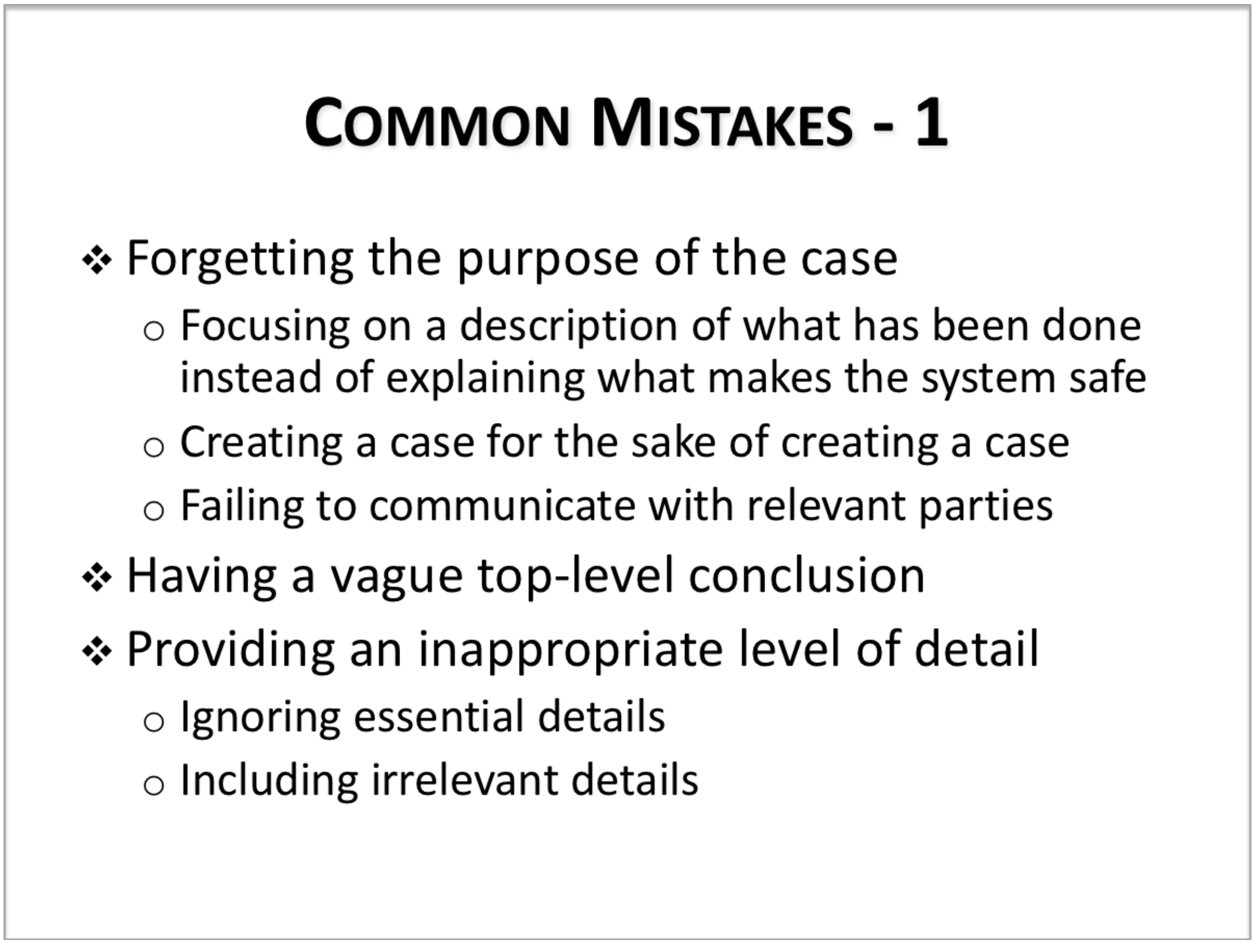
[Question to participants: What questions do you have?]

Let’s move now to talking about some common mistakes that happen when assurance cases are created. This discussion will mostly be a review of things we’ve talked about previously, both in earlier modules, and earlier in this module, so I’ll go through these quickly, unless you have some questions.

[9] I tend to think that the GSN use of the term ‘strategy’ (and its associated typical instantiations) can inadvertently contribute to missing reasoning going undetected. My pro-GSN friends dispute this contention. Neither side has developed a compelling argument to convince the other side of the error of their ways.

[10] The question of appropriate detail is one of those questions about which opinions differ strongly within the safety/assurance case community. At one far end of the spectrum are folks who claim a good assurance case must address in deep detail every aspect of the system or service. At the other far end are people who claim that no assurance case should ever be more than 1-5 pages long. My own opinion lies closer to the small case side than the huge case side.

As you may suspect, many of the common mistakes are rooted in failing to ask the questions I enumerated just now.



COMMON MISTAKES - 1

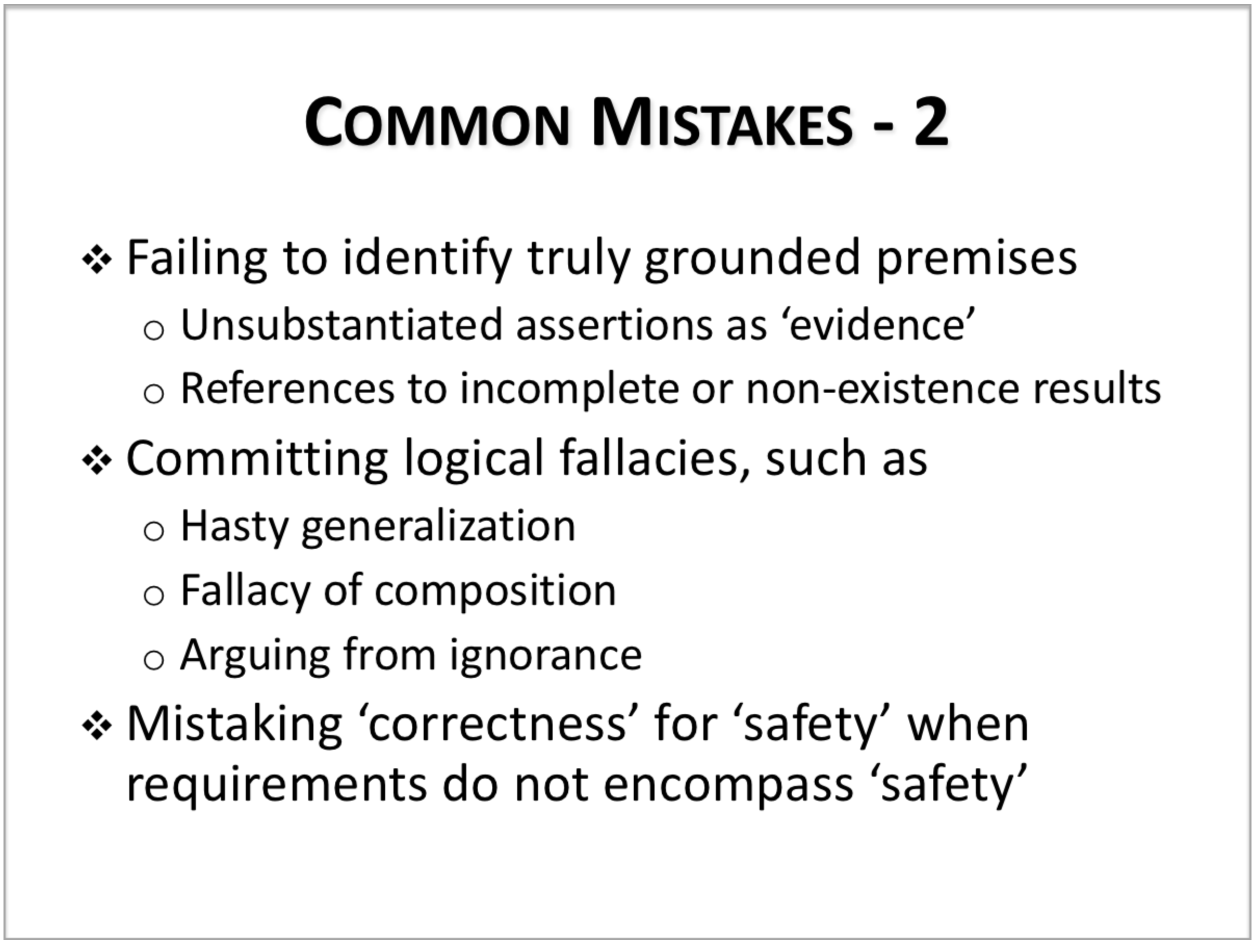
- ❖ Forgetting the purpose of the case
  - o Focusing on a description of what has been done instead of explaining what makes the system safe
  - o Creating a case for the sake of creating a case
  - o Failing to communicate with relevant parties
- ❖ Having a vague top-level conclusion
- ❖ Providing an inappropriate level of detail
  - o Ignoring essential details
  - o Including irrelevant details

Failing to ask about the purpose of the case easily results in making the mistake of forgetting the purpose of the case. This mistake may manifest itself in a number of ways, including the three you see listed here: focusing on a description of what has been done instead of explaining what makes the system safe; creating a case for the sake of creating a case; and failing to communicate with relevant parties.

Failing to question the top-level conclusion can result in having a vague (or otherwise deficient) top-level conclusion.

Not asking questions about detail frequently leads to providing an inappropriate level of detail, which can manifest in either direction: ignoring essential details, or including irrelevant details.

[Question to participants: Anyone have questions about these common mistakes before we move on to some more?]



COMMON MISTAKES - 2

- ❖ Failing to identify truly grounded premises
  - o Unsubstantiated assertions as ‘evidence’
  - o References to incomplete or non-existence results
- ❖ Committing logical fallacies, such as
  - o Hasty generalization
  - o Fallacy of composition
  - o Arguing from ignorance
- ❖ Mistaking ‘correctness’ for ‘safety’ when requirements do not encompass ‘safety’

Another common mistake, well, really a category of mistakes, is (as you may have guessed) failing to identify truly grounded premises. This failure may manifest in several ways. Giving unsubstantiated assertions as ‘evidence’, which is what we just discussed a few minutes ago. There may also be references to incomplete or non-existence results. Perhaps the author of the assurance case expected certain tests to be conducted, and thus included the results of those tests as grounded premises in the argument, but in reality those tests were never conducted.

Another category of mistakes is committing logical fallacies in the argument. I’ve listed three such fallacies on the slide.

Hasty generalization refers (as its name suggests) to making a generalization from insufficient premises. One of the most common instantiations of it is generalizing from too few observations.

As an example, suppose you start looking at odd integers. You observe that 1 is a square number, 3 is a prime, 5 is a prime, 7 is a prime, 9 is a square, and 11 is prime. You conclude, “All odd numbers are either squares or primes.” If just looked at one more odd number, 13, you’d think your generalization still holds; but the next odd number, 15, refutes the generalization.

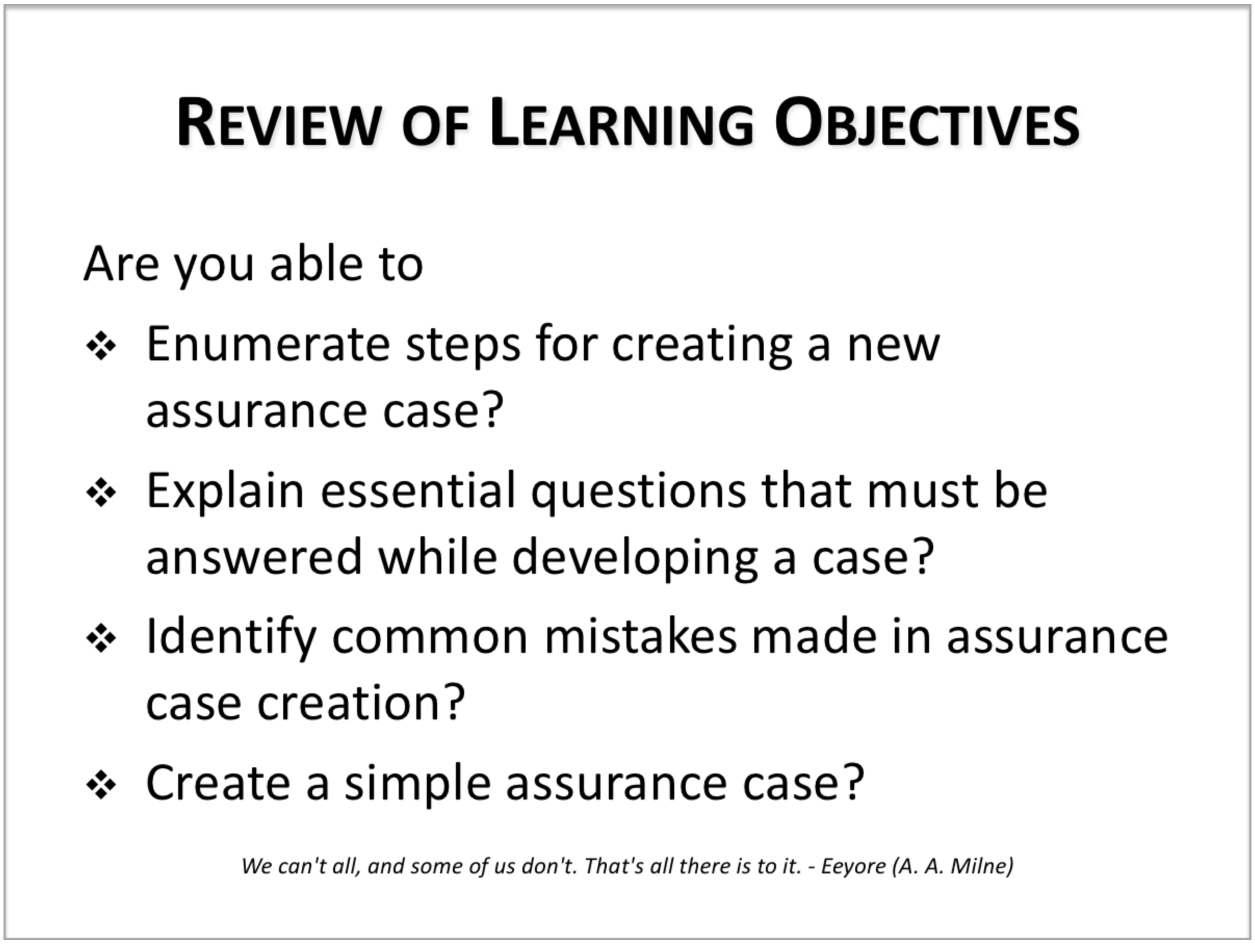
Fallacy of composition refers to inferring that a property that is true of a part is also true of the whole, without any other reasoning to establish the truth. This fallacy occurs in a safety case, for example, when the safety of individual subsystems is inferred to imply the safety of a whole system without also establishing the safety of interactions.

Arguing from ignorance is a name given to claims that something is true simply because it has not been proven false. “We ran lots of test cases and found no bugs; therefore, the software is necessarily bug-free” is a prototypical example.

A final mistake that may occur is to mistake ‘correctness’ for ‘safety’ when the requirements do not encompass ‘safety’. This mistake may be most likely to happen with software systems. If safety analysis is done in such a way that requirements are imposed on software to ensure safety (as is a fundamental assumption of DO-178C and its predecessors), then showing correctness does encompass ‘safety’. But in most other circumstances, ‘correctness’ and ‘safety’ are two different things. Conflating them is not a good thing.

That’s it for common mistakes. [Question to participants: Are there any questions or comments?]. [At this point in the original presentation I presented slide versions of a homework assignment. For this written version of Module 4, I will present the assignment at the end in straight text instead.]

At the beginning, I listed four things that I hoped you’d be able to do by the end of this module. Here are those four things recast in the form of questions. Think to yourself how you’d answer these questions.



Review of Learning Objectives

Are you able to

- ❖ Enumerate steps for creating a new assurance case?
- ❖ Explain essential questions that must be answered while developing a case?
- ❖ Identify common mistakes made in assurance case creation?
- ❖ Create a simple assurance case?

We can't all, and some of us don't. That's all there is to it. - Eeyore (A. A. Milne)

After you’ve thought about the questions for a little bit, please ask me any questions that you still have.

For those of you who want to conduct a case study about how well you have learned the material in Module 4, here is an assignment developed by Mallory Graydon.

Jill Smyth wishes to operate her ultralight aircraft from a backyard aerodrome. Refueling this aircraft has hazards, including the potential for fire. Construct an operational safety argument illustrating why it is adequately safe for Jill to refuel her aircraft as planned.

You may either assume that an assessment of the hazards of the refueling operation has been completed, or do one yourself using whatever technique(s) you like. In either case, you will need to posit plausible assumptions about the following:

- The scope of the analysis (e.g., whether to include fuel storage)
- The environment where refueling will be done
- Persons who might be present, including bystanders
- Containers and equipment used to store, move, and dispense fuel
- The type of fuel used
- The design of the aircraft, including the placement of its fuel tank, engine, and
- other components

Construct an argument to support the claim that it is adequately safe to refuel the aircraft as planned.

- Use any argument notation you prefer (for example, prose, structured text, tables, Goal Structuring Notation).
- You may use any residual risk acceptance test you prefer. But it might suffice in this case to allow readers to judge mitigations without appealing to an explicit risk acceptance test.
- Make reasonable assumptions about the kind of grounded premises (evidence) that Jill might provide.
- Focus on how operational risks are mitigated. You may assume that a separate, complete safety case report will discuss remaining issues such as responsible parties and incident reporting.
- Elaborate the arguments regarding one or two hazards down to grounded premises. It is not necessary to elaborate the arguments for all hazards.

Here are answers to some questions that you may have about the assignment.

Q. How long should I spend on the exercise?

No more than 2-3 hours. You need not read about fire hazards or create a perfect argument to complete this exercise.

Q. How do I get started?

You might begin by defining an overall safety conclusion, elaborating what it means in the first argument step, and then arguing over hazard mitigations.

Q. What is the overall safety conclusion?

Specific overall conclusions are prescribed in some domains. For this exercise you might take a broad, intuitive claim such as this following for your conclusion: The refueling operation is adequately safe. Context for this conclusion might be written as, “Procedures for refueling are defined in the airstrip policies and procedures document.”

Q. How do I define ‘adequately safe’?

As you probably know, no uniformly accepted definition of adequate safety exists. In some domains (such as commercial aviation), developers access potential risk then follow a design and development process with commensurate rigor. In other domains, developers are operators perform a risk analysis to determine residual risk than apply a risk acceptance test such as As Low as Reasonably Practical (ALARP). But for the purposes of this exercise you might define ‘adequately safe’ and ‘adequately mitigated’ implicitly through the premises you supply.

Here is an example of using this implicit definition approach for the top-level conclusion, “My word burning stove is adequately safe to use.”

```
Conclusion: My word burning stove is adequately safe to use.
Premises:   The risk of carbon monoxide poisoning is adequately mitigated.
            The risk of a chimney fire is adequately mitigated.
            … …

Reasoning:    Establishing adequate mitigation of identified hazards is
            sufficient to show adequate safety.

Conclusion: The risk of carbon monoxide poisoning is adequately mitigated.
Premise:    My living room is fitted with a functioning carbon monoxide
            detector.
… … …
```

Q. How much detail do I need to include?

As much as you think appropriate to include while abiding to the 2-3 hour time limit. As noted in the module, level of detail is a subject of debate. It is usually possible to add more detail to any argument. But added detail might either illuminate important issues or clutter the argument. Case writers must balance explicitness and brevity. For this exercise, try to develop your argument (for at least one hazard) to a level that seems appropriate for both the matter at hand and the likely readers of the safety argument.

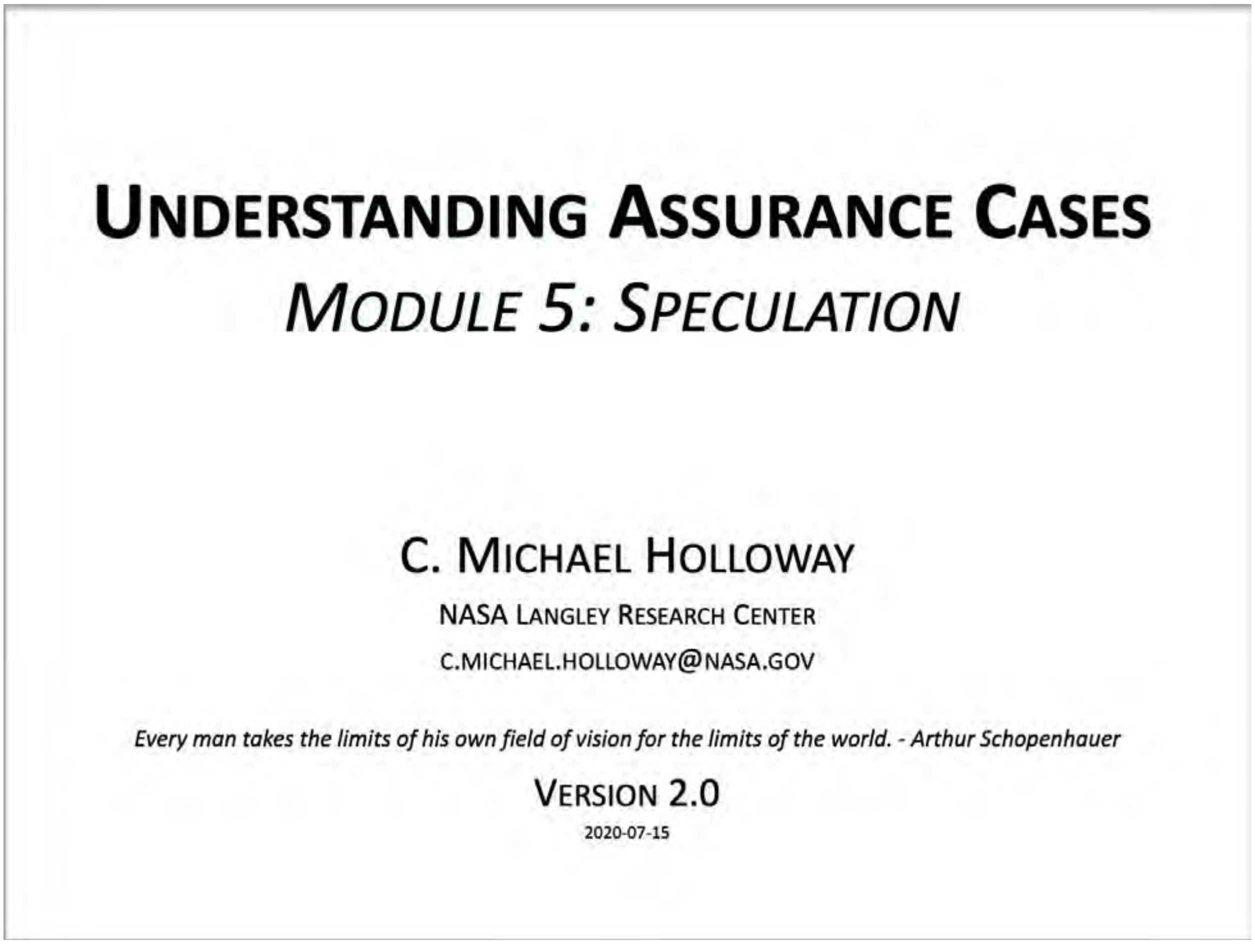
If you have questions or comments about this module, including the homework, contact its author at `c.michael.holloway@nasa.gov`.

Understanding Assurance Cases: An Educational Presentation in Five Parts

Module 5: Speculation

C. Michael Holloway
`c.michael.holloway@nasa.gov`

Senior Research Computer Engineer
Safety-Critical Avionics Systems Branch
NASA Langley Research Center, Hampton, Virginia, U.S.A.



UNDERSTANDING ASSURANCE CASES

MODULE 5: SPECULATION

C. MICHAEL HOLLOWAY

NASA LANGLEY RESEARCH CENTER

C.MICHAEL.HOLLOWAY@NASA.GOV

Every man takes the limits of his own field of vision for the limits of the world. - Arthur Schopenhauer

VERSION 2.0

2020-07-15

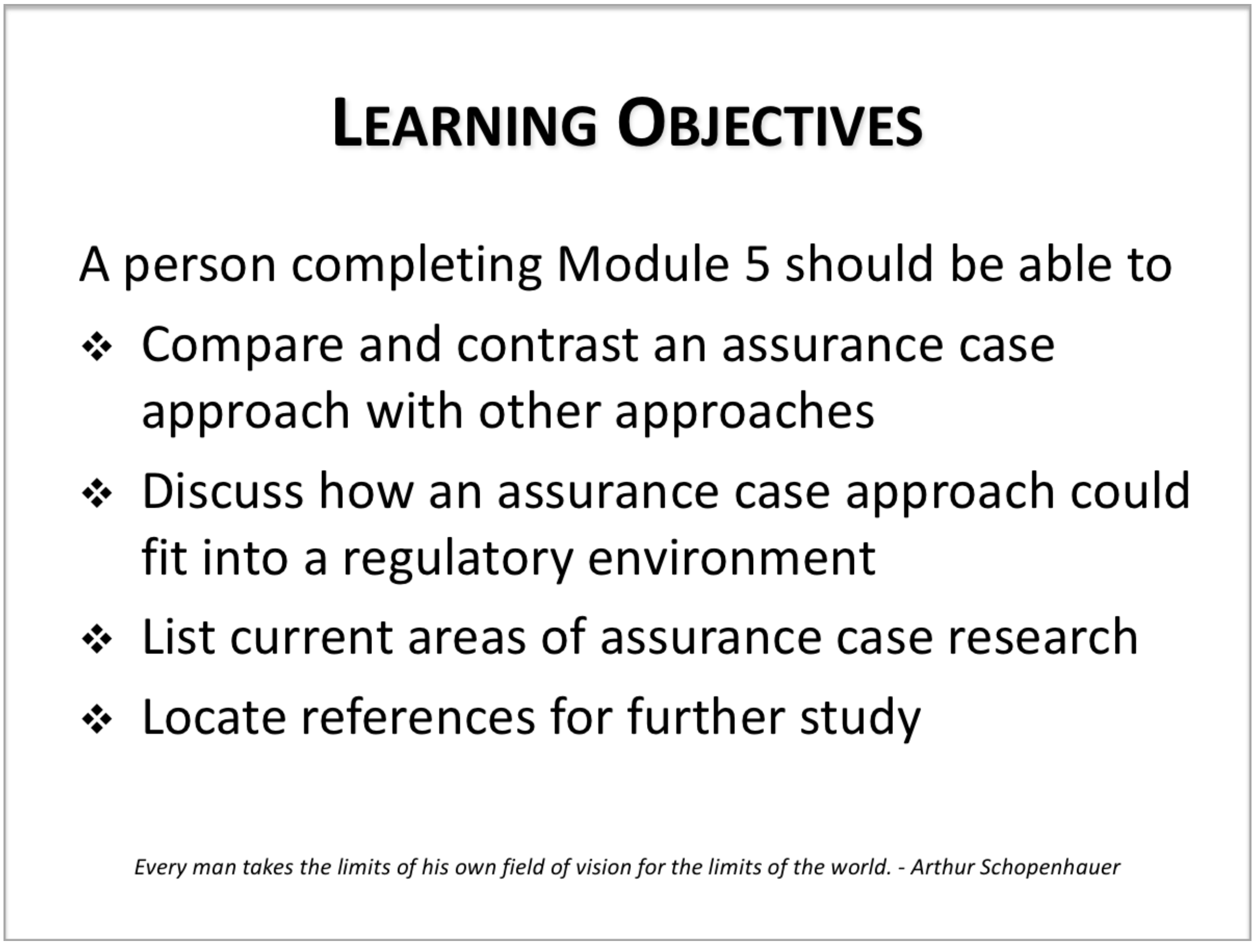
This material was created in 2015-16, as part of the Explicate '78 project. The project was supported in substantial part by the Assurance Case Applicability to Digital Systems task under the reimbursable interagency agreement with the Federal Aviation Administration for Design, Verification, and Validation of Advanced Digital Airborne Systems Technology (IAI-1073 Annex 2 for NASA; DTFACT-10-X0008, Modification 0004 for the FAA). The original presentations were delivered to a selected group of FAA civil servants and NASA Langley personnel. The audio was recorded and partial transcripts (containing only the words spoken by the presenter, Mr. Holloway) produced. The intent from the beginning was to collect the material into a form that could be made available publicly. The text adheres closely to the original transcript, with the exception of an occasional insertion of new information that arose since the original presentation dates. The full collection consists of six documents (including this one), which are available electronically through `https://shemesh.larc.nasa.gov/arg/uac.html`.

Hello everybody.

We've now come to the fifth, and final module in our educational series about Understanding Assurance Cases, which I've titled ***Speculation***. The topics we will discuss are a bit less concrete and a lot more speculative than the topics of the previous 4 modules.

Perhaps in talking about these topics we can provide a counter-example to Arthur Schopenhauer's assertion that "Every man takes the limits of his ***own*** field of vision for the limits of the world." [Schopenhauer, Arthur. 1951. *Studies in Pessimism: Essays from the Parerga and Paralipomena*. Translated by T. Bailey Saunders. London: Allen and Unwin.]

As always interrupt me at ***any*** point if you have a burning question. I'll either answer it or defer an answer to a more appropriate time. Also, as with the other modules, there will be a few times when I'll ask you questions, too.



Learning Objectives

A person completing Module 5 should be able to

- ❖ Compare and contrast an assurance case approach with other approaches
- ❖ Discuss how an assurance case approach could fit into a regulatory environment
- ❖ List current areas of assurance case research
- ❖ Locate references for further study

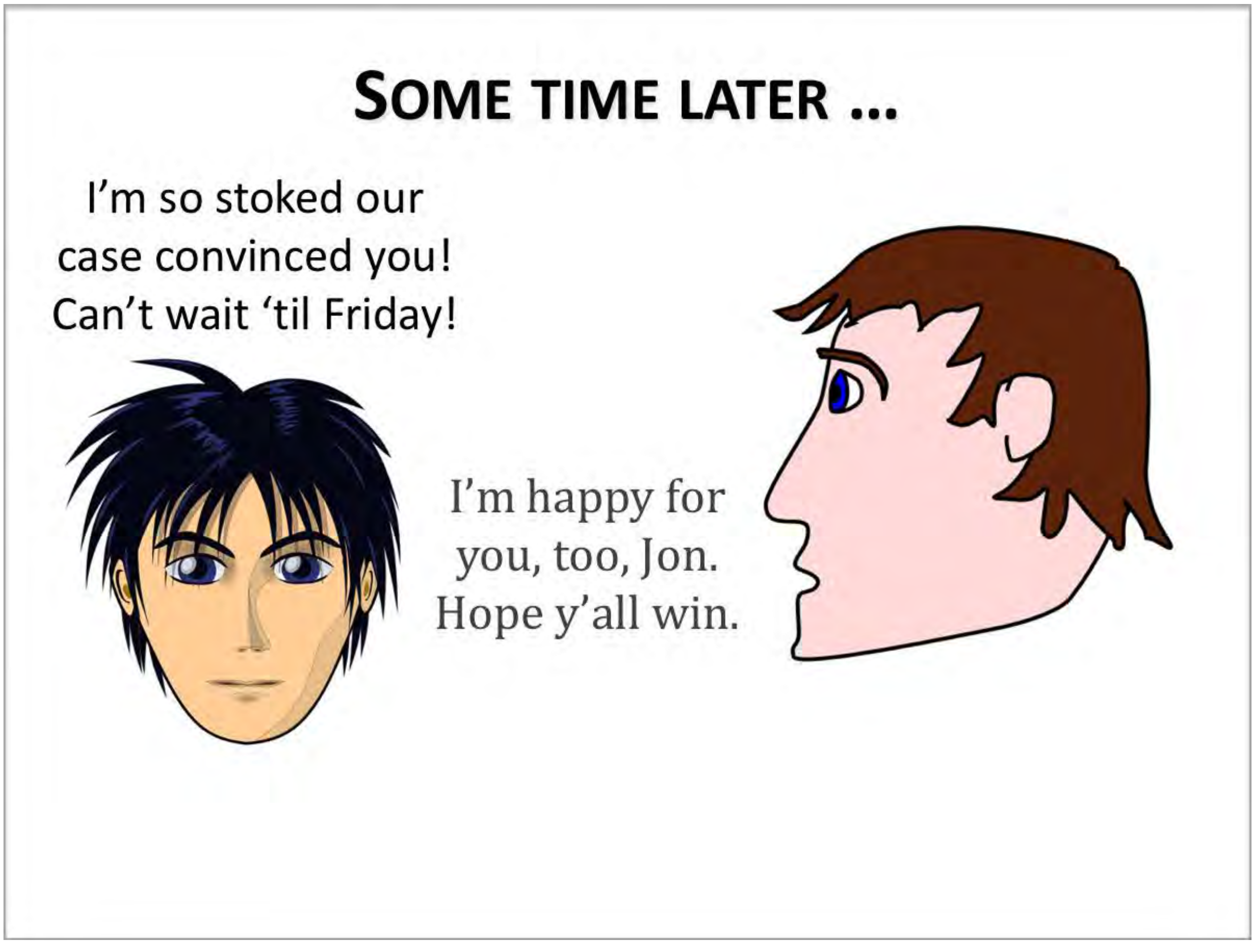
Every man takes the limits of his own field of vision for the limits of the world. - Arthur Schopenhauer

Here are our learning objectives for this module:

- Compare and contrast an assurance case approach with other approaches
- Discuss how an assurance case approach could fit into a regulatory environment
- List current areas of assurance case research
- Locate references for further study

In Module 4, we left Jon and Mike with Jon thanking his Dad for agreeing to show him how to create an assurance case concerning Tim driving Jon to the game.

We still don't know exactly who is playing in the game, or even what sport it is, which I find a bit disconcerting. But today we learn that Jon's case convinced his dad to let him go to the game.


SOME TIME LATER …
I'm so stoked our case convinced you! Can't wait 'til Friday!
I'm happy for you, too, Jon. Hope y'all win.

"I'm so stoked our case convinced you! Can't wait 'til Friday!"

(So we now know when the game is taking place.)

"I'm happy for you, too, Jon. Hope y'all win."

Mike's response suggests that perhaps Jon's school is one of the teams in the game.

After a brief pause, Jon says, "One more question … then I've gotta do homework."

"OK, Let's hear it," replies Mike.

"This case stuff really made me think. Tim said so, too. It seems like such a great idea. Why don't more people use it?" Jon asks.

His dad replies, "Wow. … that's a hard question with lots of different parts to the answer."

“Like what, for instance? ” inquires Jon.

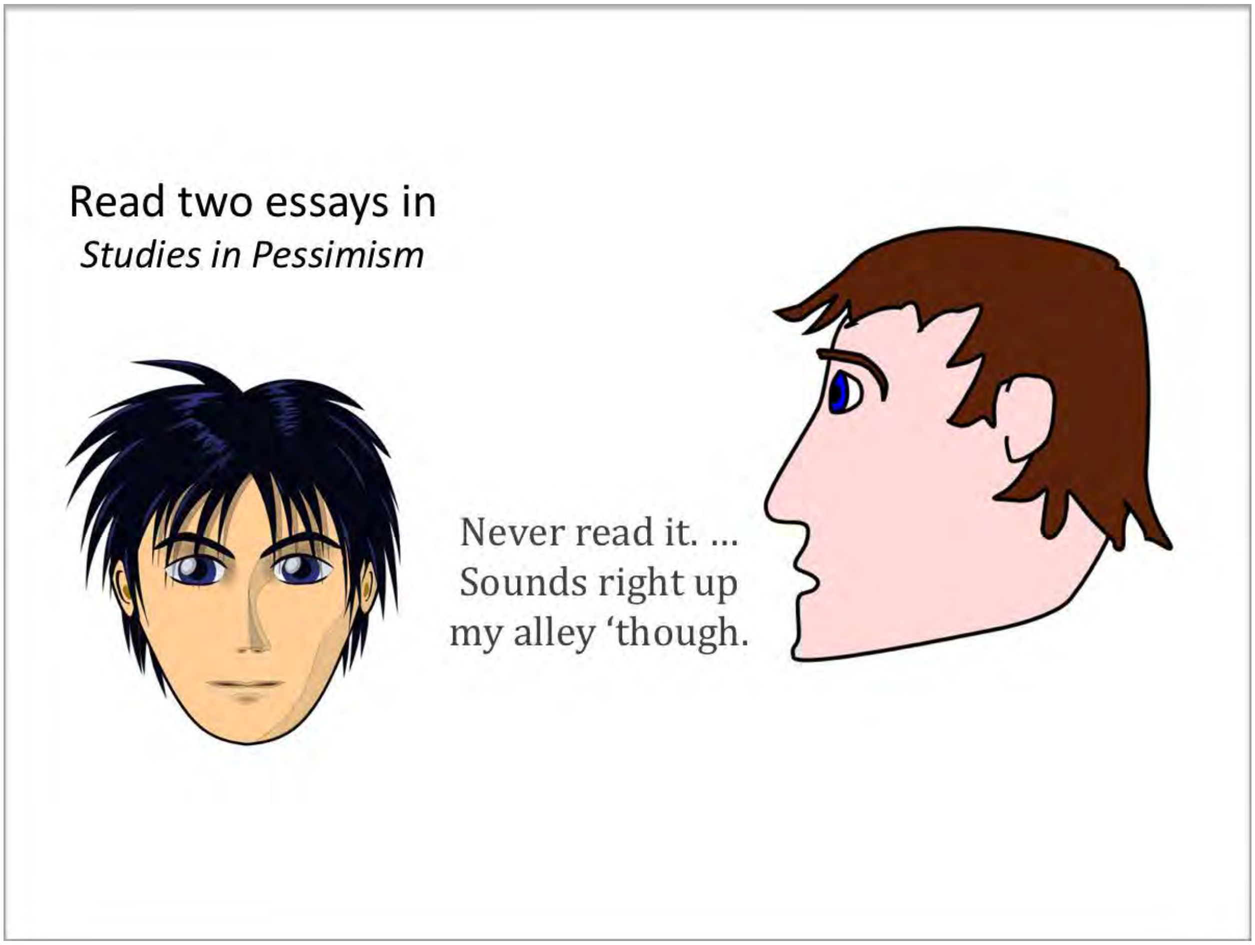
“Lack of understanding … thinking it is harder than it is … or thinking it adds nothing to what’s usually done … or even just believing rants from some experts.”

“Sounds confusing. When’s it gonna end?”

“Don’t know if it ever will,” says Mike frowning, “unless someone figures out a good way to compare how well different methods work.”

Jon says nothing for a second or two, then exclaims, “Better go do my homework now.”

“Watcha gotta do,” asks Mike.



“Read two essays in *Studies in Pessimism.”*

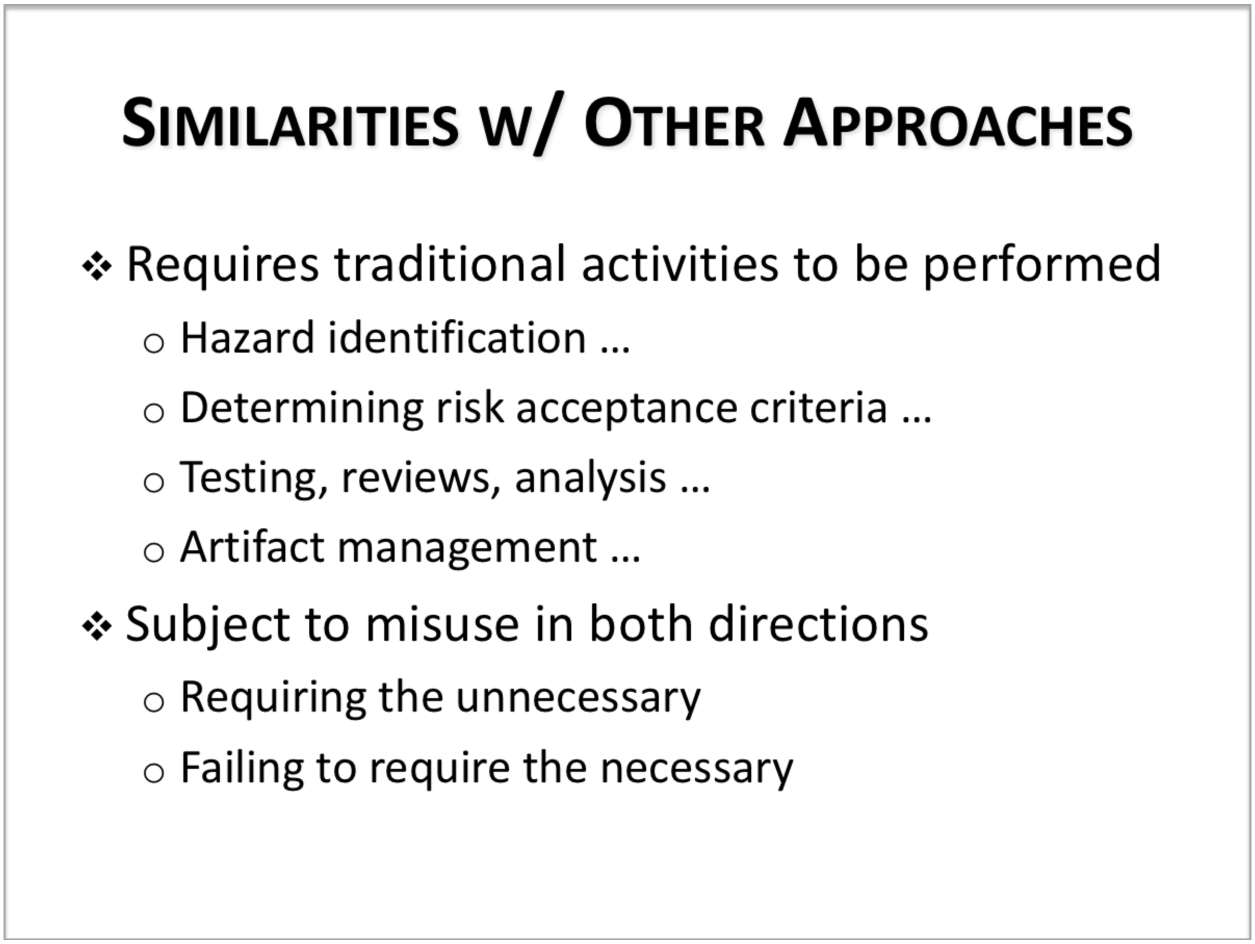
“Never read it,” says Mike, “Sounds right up my alley ‘though.”

Studies in Pessimism is by Schopenhauer by the way, and is the source of the quote for Module 5[1].

[1] Well, to be more precise, *Studies in Pessimism* is the English title of a translation of a compilation of some of Schopenhauer’s writings

To continue along the lines Mike mentioned, let’s now spend a little bit of time comparing and contrasting an assurance case approach to other approaches.

We’ll begin with similarities.



Similarities w/ Other Approaches

- ❖ Requires traditional activities to be performed
  - o Hazard identification …
  - o Determining risk acceptance criteria …
  - o Testing, reviews, analysis …
  - o Artifact management …
- ❖ Subject to misuse in both directions
  - o Requiring the unnecessary
  - o Failing to require the necessary

One of the most important similarities between an assurance case approach and traditional approaches to safety assurance is that writing an assurance case still requires traditional activities to be performed. We’ve talked about this several times in previous modules, but I want to emphasis it again here. Assurance cases are not a way to get out of doing necessary technical work. They may provide a way to get out of doing unnecessary “administrative”-type work, but *they’re not a shortcut to safety*.

In particular, writing an assurance case doesn't absolve you of the need to identify hazards or to determine risk acceptance criteria or to do testing, reviews, and analysis or to manage all your artifacts well. All of those things still need to be done, perhaps in slightly different ways, using different notations or techniques, but they still have to be done.

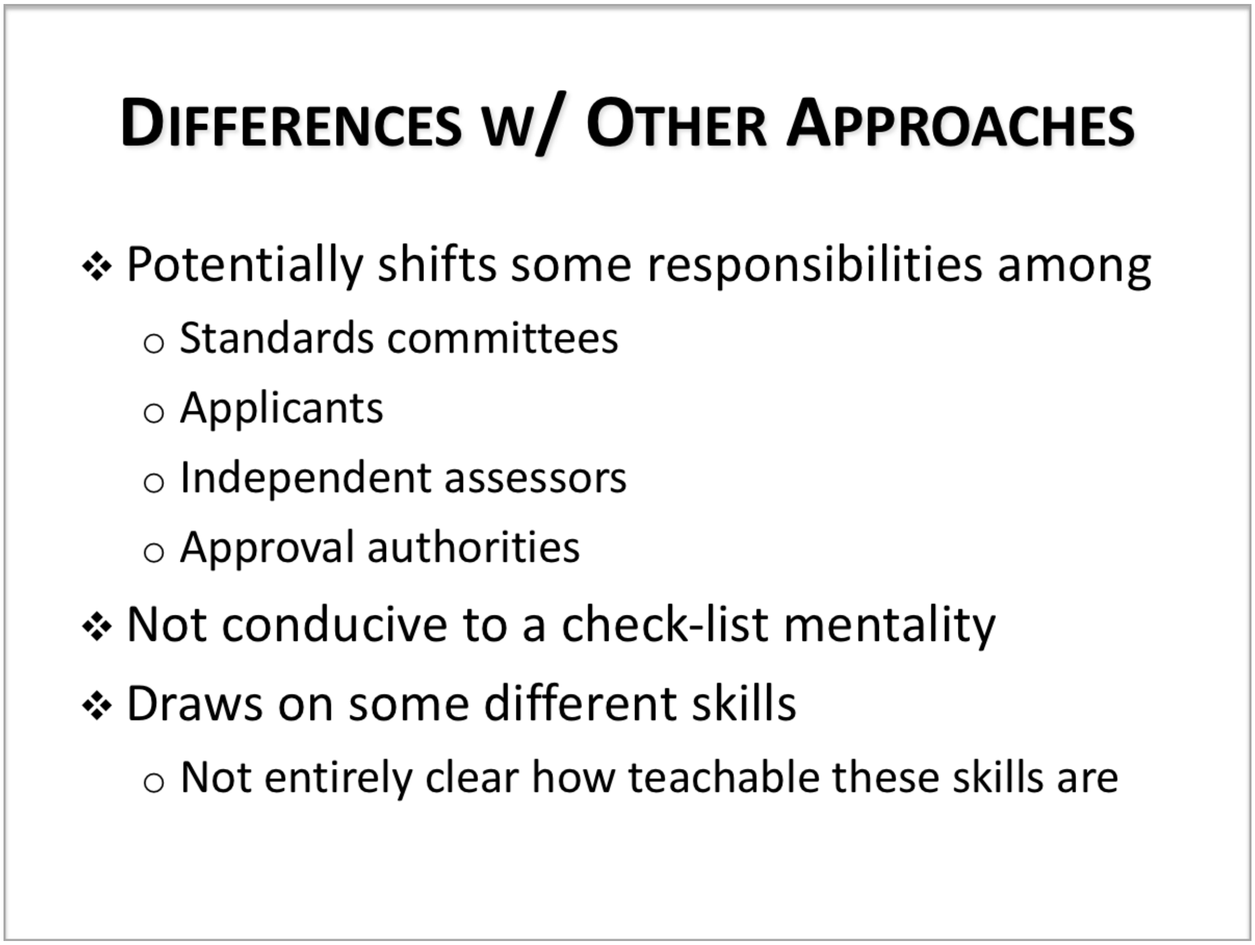
Another similarity between assurance case techniques and other techniques is that they do not provide a guarantee against misuse. Someone may adopt an insurance case approach and still require things that are unnecessary. Or fail to require things that are necessary. *Assurances cases are not a silver bullet*.

To emphasize again, they're neither a shortcut nor a silver bullet. Unsafe, incorrect, bad systems can be mistakenly assured as safe, correct, good under an assurance case regime just as they can be under some other regime. Perhaps you recall from Module 2 the Nimod accident, before which a bad system was proclaimed safe through an abysmal safety case.[2]

[Question to participants: Who has questions about similarities? Or perhaps suggestions for other similarities?]

Assurance cases are not identical to other approaches, ‘though, so let’s now talk now about some differences.

I’ll show two slides about differences. Here is the first.



Differences w/ Other Approaches

- Potentially shifts some responsibilities among
  - Standards committees
  - Applicants
  - Independent assessors
  - Approval authorities
- Not conducive to a check-list mentality
- Draws on some different skills
  - Not entirely clear how teachable these skills are

One of the most important differences concerns the potential for shifting responsibilities among various entities such as standards committees, applicants (in FAA terminology), independent assessors, and approval authorities. Exactly how these responsibilities may shift depends on the particular approaches that are taken to employing assurance cases.

[2] Critics of assurance/safety case approaches sometimes cite examples of poorly constructed and inadequately evaluated cases as conclusive evidence of inherent flaws in the approach. Supporters, on the other hand, point out that examples of improper use do not mean that proper use is impossible.

If, for example, a wide-open, un-fettered use of assurance cases is permitted, then it could well be the case that standards committees would be irrelevant, except perhaps if there is a standard for the particular notation used. In such an environment, the role of independent assessors, who would perhaps evaluate the assurance case arguments, may be greatly expanded.

As another example, in a somewhat more structured assurance case environment, perhaps the approval authorities would have a catalog (that's probably not the right word, but I think it may convey the general idea) of acceptable assurance case structures, and applicants would generally be expected to create their arguments using those structures.

Another difference between assurance case approaches and some traditional approaches is that assurance cases are generally not conducive to a checklist mentality. That is, one cannot easily create a checklist against which an argument can be evaluated unthinkingly. You wouldn't for example simply have a checklist that says look for at least three conclusions, six premises, and two reasons. As we saw in Module three evaluating assurance case arguments is not trivial.

Which brings us to the third difference, namely, that the use of assurance cases seems to draw on some different skills than perhaps are usually possessed by engineering organizations and regulators. It is not entirely certain that different skills are essential, but it seems intuitively to be so. At present the jury is still out on how teachable these skills may be. Looking into this issue seems like a fruitful, but difficult, area of research.

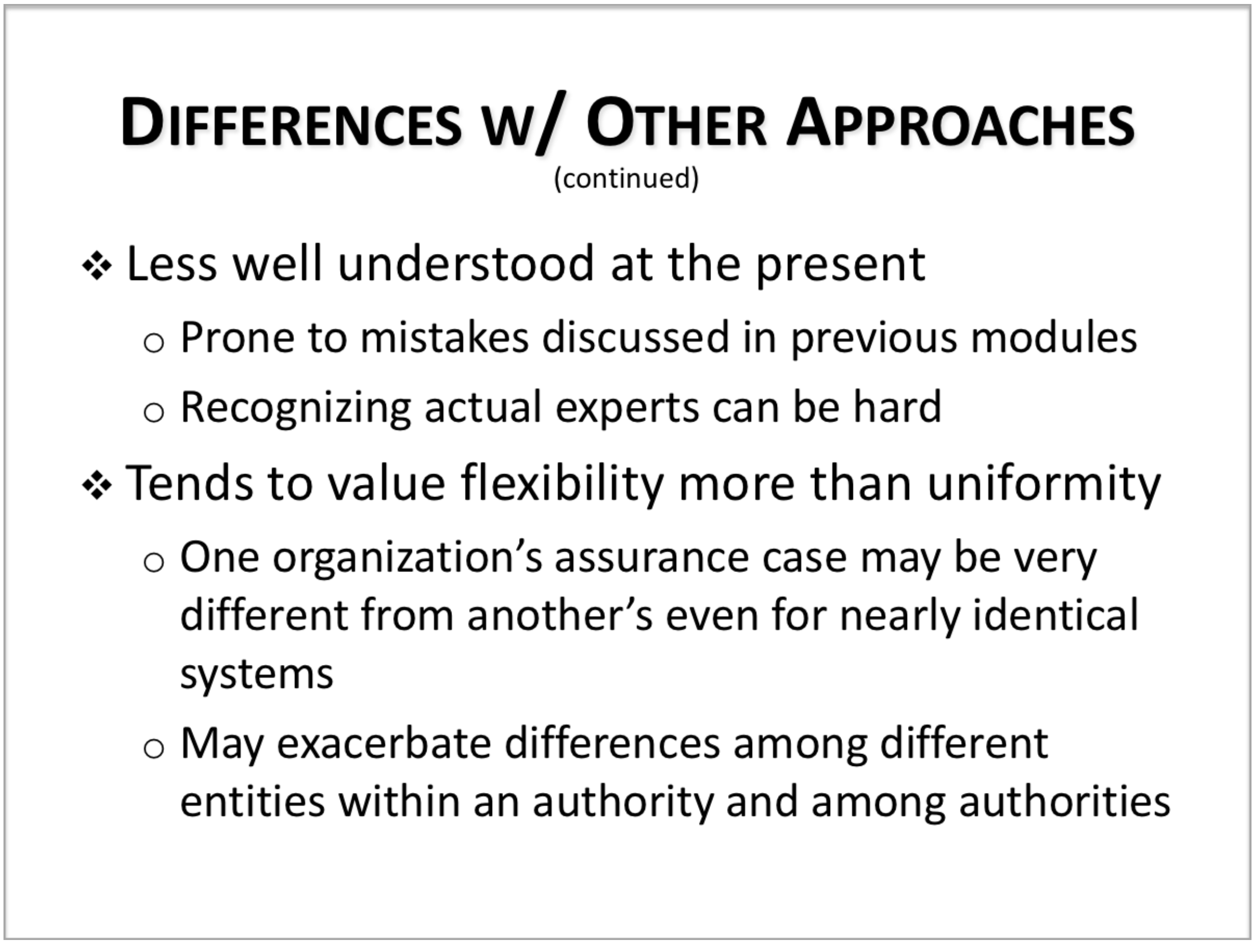
We'll talk about research in just a few minutes, but let's continue with the differences.

[Question to participants: Any questions on this slide before I go to the next one?]

Another difference between assurance cases and traditional techniques is that, within the US at least, assurance case methods are less well understood.

We've talked in previous modules about the sorts of mistakes that novices can make; I won't reiterate those here, unless someone wants me to do so.

Perhaps more dangerously, the general lack of understanding means that recognizing actual experts can be hard. Because assurance cases have become a somewhat trendy topic within academic circles people have jumped on the bandwagon without necessarily having the knowledge to contribute anything useful or to even recognize they are unable to do so. (The temptation to go into a rant at this point is great, but I shall resist it.)



DIFFERENCES W/ OTHER APPROACHES

(continued)

- ❖ Less well understood at the present
  - o Prone to mistakes discussed in previous modules
  - o Recognizing actual experts can be hard
- ❖ Tends to value flexibility more than uniformity
  - o One organization’s assurance case may be very different from another’s even for nearly identical systems
  - o May exacerbate differences among different entities within an authority and among authorities

Another difference, which in some ways may be the biggest one, particularly as far as the use in regulatory environments may be concerned, is that the assurance case approach tends to value flexibility more than uniformity.

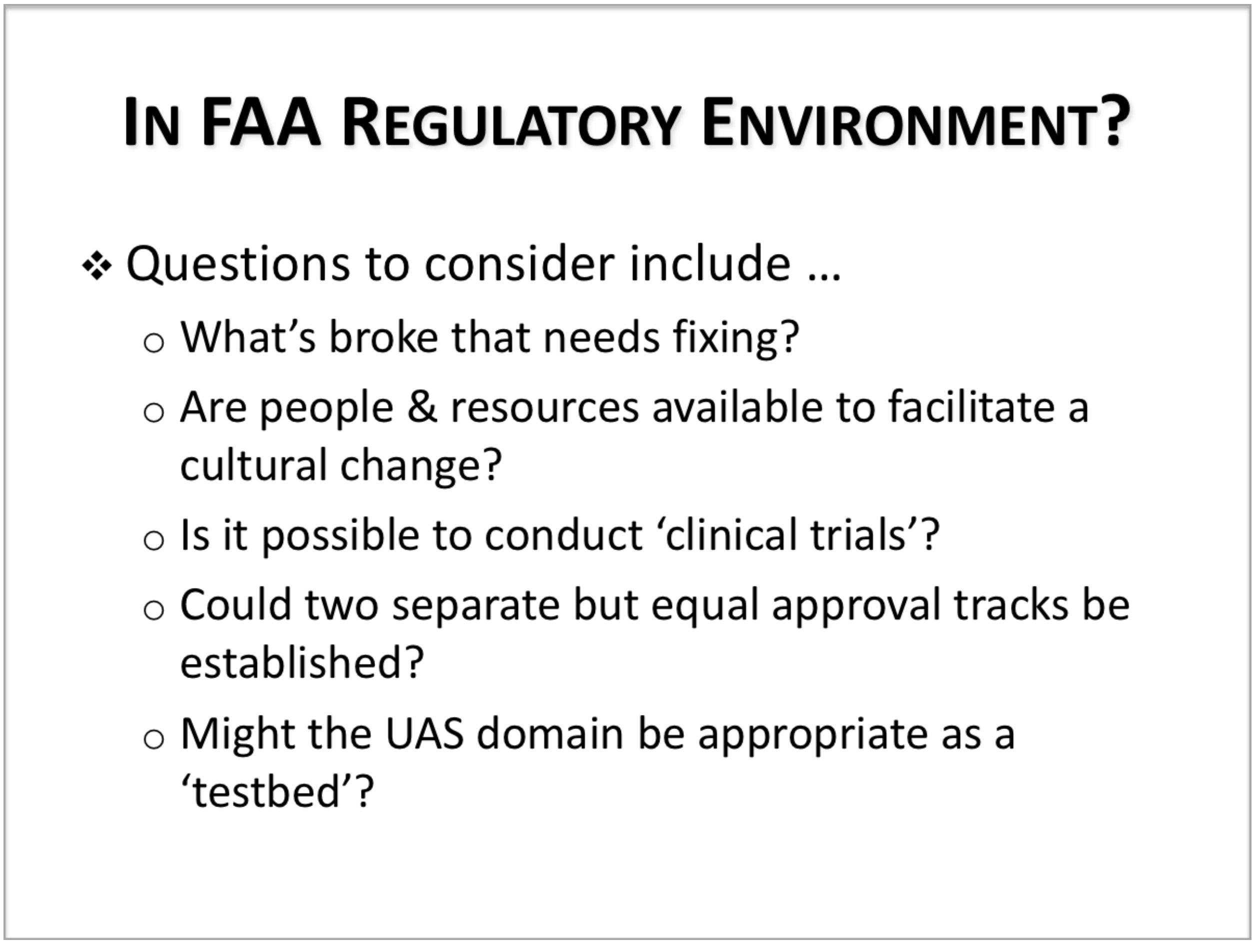
In a general assurance case regime, one organization’s assurance case may look very different from another's even for nearly identical products or systems or subsystems. The case might be structured differently, it might use different notations, it might take it different approach to specifying reasoning, it might take a different approach to addressing defeaters, and so on.

These differences could very well serve to exacerbate already existing differences among entities within a single approval authority and among different approval authorities. A case that is accepted in one region may be rejected in another, for example.

This difference leads directly into our next subject, which is talking a bit about questions that need to be considered concerning using some form of assurance case approach within an FAA regulatory environment[3].

[Question to participants: But before we do that, are there any questions?]

[3] The discussion here is in the context of the FAA environment, but the general questions should be similar, or have analogs, in just about any regulatory situation.



In FAA Regulatory Environment?

- Questions to consider include …
  - What's broke that needs fixing?
  - Are people & resources available to facilitate a cultural change?
  - Is it possible to conduct 'clinical trials'?
  - Could two separate but equal approval tracks be established?
  - Might the UAS domain be appropriate as a 'testbed'?

In thinking about assurance cases and the FAA environment, I'm going to suggest five general questions that I think need to be carefully considered. Let's read them all together, then discuss each one a bit more.

One very important question is "What's broke that needs fixing?"

Question two is "Are people and resources available to facilitate a cultural change?"

The third question is, "Is it possible to conduct 'clinical trials'?"

Question four: "Could two separate but equal approval tracks be established?"

And the final question that I propose is, "Might the UAS domain be appropriate as a 'testbed'?

The "what's broke" question is critically important, because its answer may go a long ways towards helping to decide whether some form of assurance case approach is likely to help fix the perceived problems.

As we just discussed, in general assurance case approaches tend to promote flexibility at the expense of uniformity.

If the biggest problems that are currently facing the FAA in terms of the regulatory environment is that the environment is too rigid, that it tends to discourage or even prevent useful innovation, then moving towards an assurance case regime may well help address those sorts of problems. If, on the other hand, the biggest problems involve inconsistency among different approvers within the FAA or between the FAA and EASA, then moving towards an assurance case regime may not help at all. (I'm not saying it wouldn't be possible to create a specialized assurance case regime that could help with such a situation, just that it may be difficult).

Concerning resources being available to facilitate a cultural change, I think what we've discussed in these series of lessons have made clear that some cultural changes would be necessary. As we just discussed a few minutes ago, there may need to be some skill-set changes, too. Unless resources will be available to make these things happen, moving towards an assurance case approach is not likely to succeed.

The last three questions on the slide all are about the same general theme, "How can you go about establishing that an assurance case approach 'works' well for the FAA?"

I mention the idea of a 'clinical trial' approach because it may provide a fairly inexpensive initial assessment of feasibility. The idea of 'separate but equal' approval tracks is meant to suggest the possibility of allowing organizations to continue what they're doing now if they like, or to try going down an assurance case based track instead[4]. If it turns out that the assurance case based track doesn't work, then all that would be necessary is to remove that track; no changes would otherwise be necessary. Finally, suggesting that the UAS domain might be appropriate as a 'testbed' stems simply from my perception that this area seems to be in a bit of turmoil right now, and trying out assurance case approaches to regulation there might (or might not) help resolve some of the turmoil.

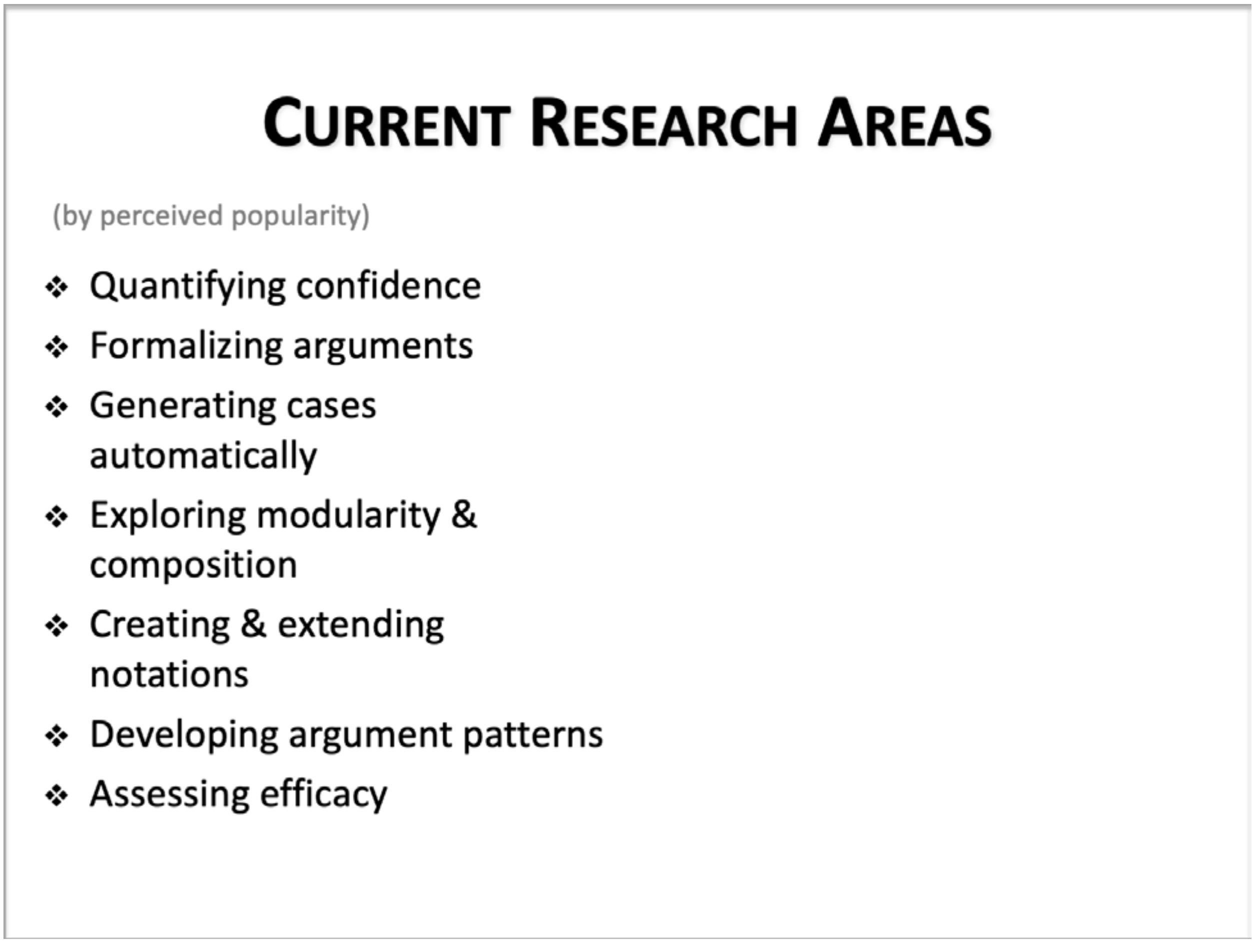
[Question to participants: That's all I've planned to say on this topic, does anyone have some questions?]

We'll move now to talking a little bit about the research that is currently going on in the assurance case / safety case arena. This discussion will be necessarily quite subjective, and you will easily be able to find people who have very different opinions from my own. Please remember that I giving you only my personal thoughts, none of which should be construed to represent an official NASA position.

Shortly I will show you two different lists of current research topics. First, you will see a list ordered by my subjective evaluation of current popularity. The ordering is entirely subjective, but I did ask some other people within the community for their opinions, and they generally agreed with my ordering, with only an occasional exception. Second, you will see a list ordered by my opinion of the priority that ought to be given to the various topics[5].

[4] The Overarching Properties work which arose after, and was partially motivated by, the Explicate '78 project is based on applying this principle.

[5] Although these orderings were developed in 2016, I do not think any significant changes (to either side) have happened in the last four years.



Current Research Areas

(by perceived popularity)

- Quantifying confidence
- Formalizing arguments
- Generating cases automatically
- Exploring modularity & composition
- Creating & extending notations
- Developing argument patterns
- Assessing efficacy

The first three areas that you see here — quantifying confidence, formalizing arguments, and generating cases automatically — are almost certainly the currently most popular research areas.

Each of these research areas has some first glance appeal.

If it is possible to place useful numbers on the degree of justified confidence that one should have in an assurance case argument …

Before continuing that sentence, let me explain what I mean by useful numbers.

I mean numbers that can be compared and manipulated, so that, for example, a confidence score of 995 would be known to always be better than a score of 850, and that if a minimum threshold of say 990 was required, we could be sure that a score of 993 indicated sufficient justified confidence.

So, repeating the sentence I started ….

If it is possible to place useful numbers on the degree of justified confidence that one should have in an assurance case argument, then having such numbers would seem to be clearly a good thing.

Similarly, if it is possible to formalize assurance arguments, particularly to make them purely deductive (if you don't remember from Modules 1 & 3 what purely deductive means then ask, or look it up if you're reading the material), then formalizing them seems like a good thing, too. Much of the evaluation of formal arguments could be done automatically. And, if it is possible to generate arguments automatically, based purely on things that engineers are already doing, then this, too, seems to be a great thing to do.

The next three items that you see on the slide are also being research fairly actively.

Exploring modularity & composition refers to efforts aimed at creating arguments that can be reused directly in other contexts and to developing ways to compose existing arguments into higher-level arguments without having to change or reevaluate the individual original arguments.

Creating & extending notations is pretty self-explanatory.

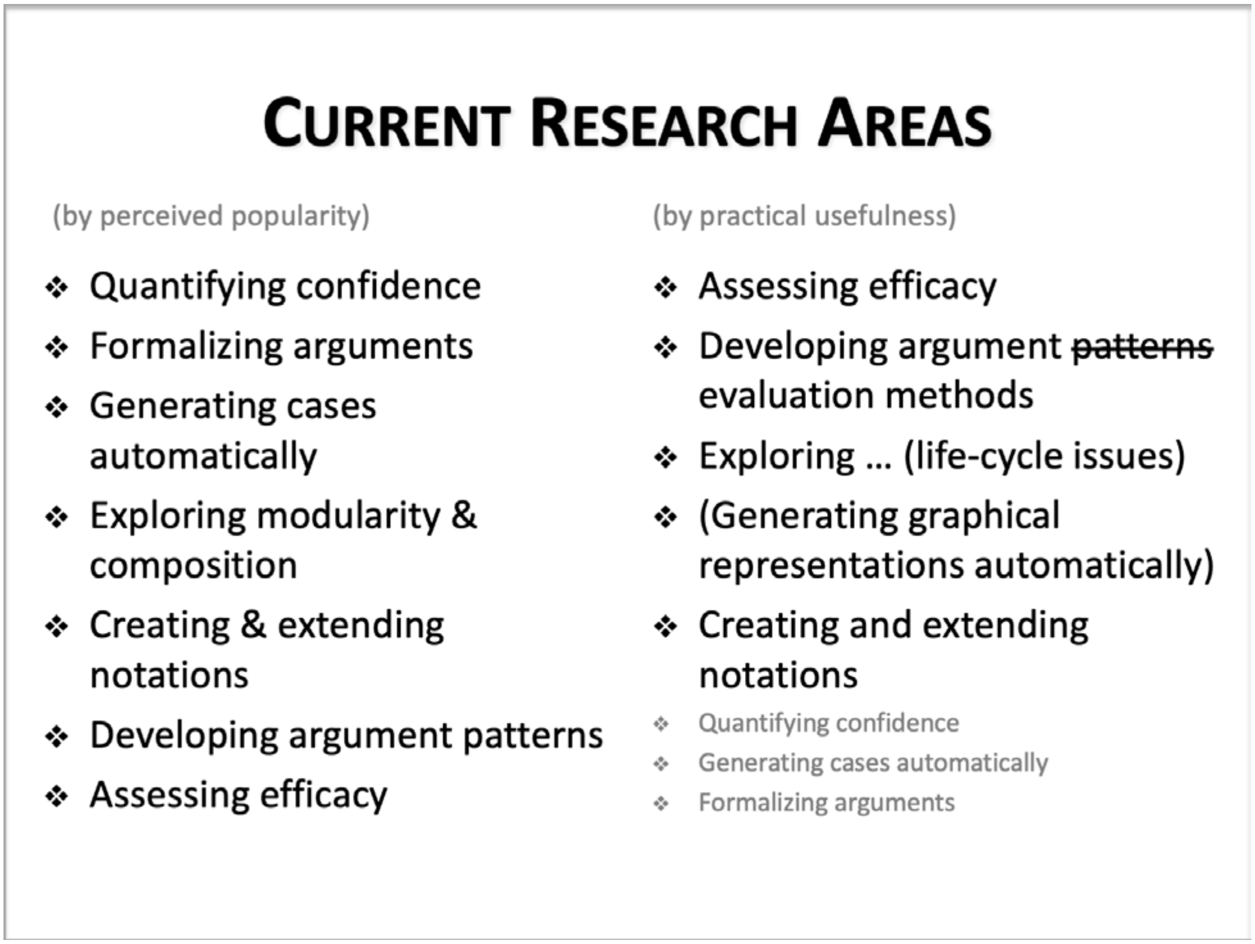
Developing argument patterns is a bit similar to the modularity and composition idea, but on a different scale. Rather than trying to create completely reusable arguments, pattern research seeks to create general frameworks for certain types of arguments, which then may be instantiated with system specifics as necessary.

The final item, you see here, assessing efficacy, refers to efforts to determine whether, and if so, how, assurance cases truly provide the benefits that proponents claim. Think back to Mike's comment to Jon. To date, all of the efforts in this area have tended to involve case studies, retrospective evaluations, or non-public proprietary studies.

[Question to participants: Any questions about what I mean by any of these areas?]

I will now show you a different ordering and slightly different set of research areas that corresponds to what I personally think ought to be going on.

Once again, please remember that I am showing you *only* my opinion. Plenty of smart people within the assurance case research community disagree with me.



Current Research Areas

(by perceived popularity)

- Quantifying confidence
- Formalizing arguments
- Generating cases automatically
- Exploring modularity & composition
- Creating & extending notations
- Developing argument patterns
- Assessing efficacy

(by practical usefulness)

- Assessing efficacy
- Developing argument ~~patterns~~ evaluation methods
- Exploring … (life-cycle issues)
- (Generating graphical representations automatically)
- Creating and extending notations
- Quantifying confidence
- Generating cases automatically
- Formalizing arguments

Perhaps the first thing you'll notice is that the order is almost directly inverted from the current popularity order. Or perhaps the first thing you'll notice is that I have written the three areas that are currently the most popular in grey and a small font. I did this because, despite the first-glance intuitive appeal of these areas, in practice, given the current state of the art and state of our knowledge, the "if" clauses for all three are practically false.

It is *not* possible to generate useful numbers. Well, to be more precise, none of the proposals thus far to do so can withstand scrutiny[6].

It is *not* possible to formalize important parts of assurance case arguments. The concepts with which these arguments are concerned are often not formal concepts themselves, but rather emergent, non-deductive properties that can't be described precisely in any existing logical formalism.

And finally, it is *not* feasible to generate very many useful assurance case arguments automatically, partially because automatic generation assumes some sort of formalization.

[6] See the journal article [Graydon, P.J., Holloway, C.M. 2017. An Investigation of Proposed Techniques for Quantifying Confidence in Assurance Arguments. *Safety Science,* vol. 92, pp. 53-65.] and the significantly longer and more detailed technical report [Graydon, P.J., Holloway C.M. 2016. An Investigation of Proposed Techniques for Quantifying Confidence in Assurance Arguments. NASA/TM-2016-219195.] for the results of applying scrutiny to existing quantification techniques.

I think that the current top three most popular areas of research should be mostly abandoned, or left entirely in the hands of academic departments who have no interest in practicality.

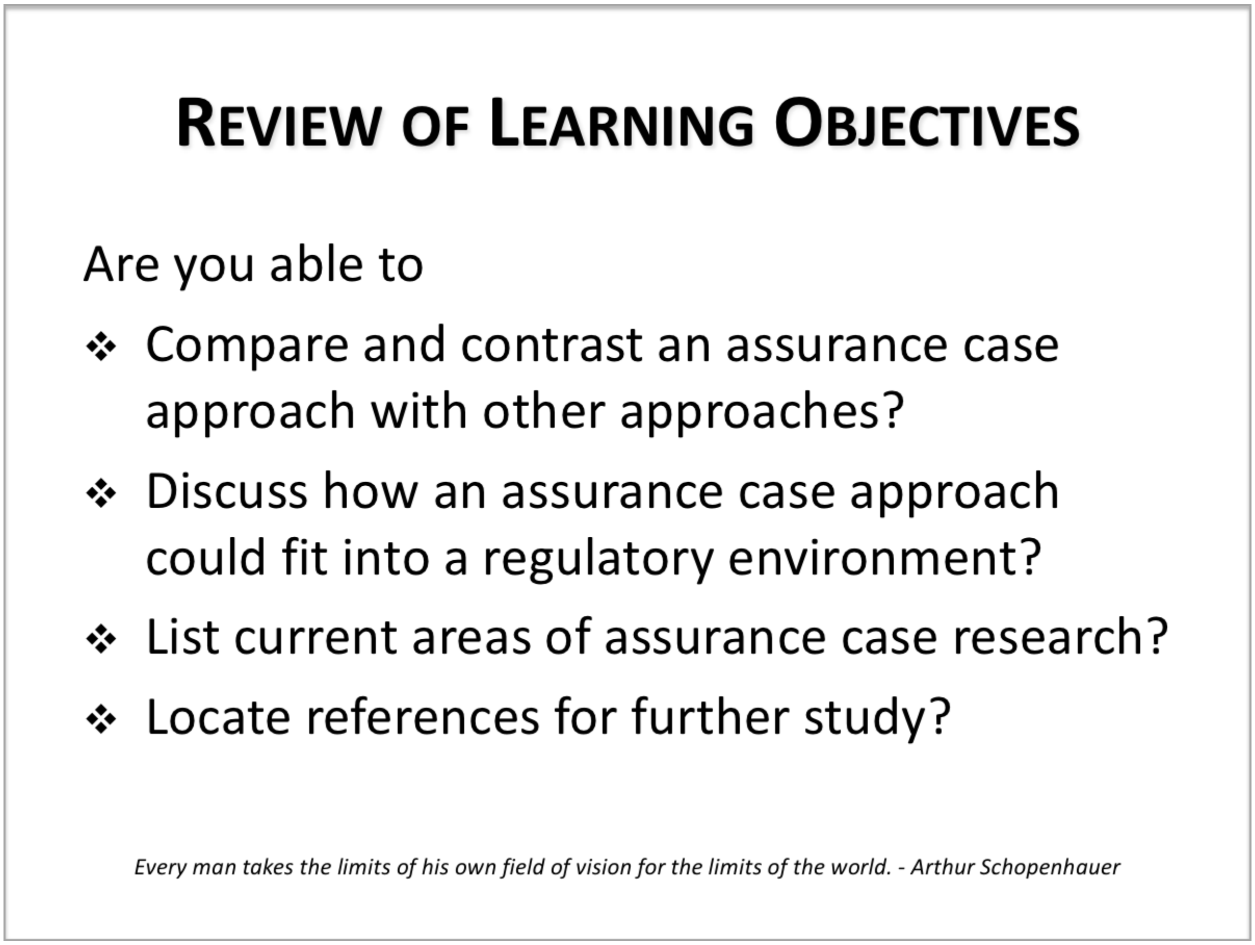
The area that is currently the least studied, should be, in my opinion, the most studied, namely assessing efficacy: determining whether assurance cases truly provide the benefits proponents claim they do. Such research is not easy, nor is it cheap, nor is the sort of thing that seems to be currently in vogue with funding agencies at the moment, but I think it is critical. We at NASA have started a bit of work in this area, and are hoping to be able to expand the work further. Perhaps you will be reading about the results of the work one day.

I won't go into anything more about these other areas unless someone has a question.

Let's now talk a bit about what you can do to further your study of assurance cases.

(At this point in the original presentation, I showed three slides. On each slide was a list of five references for further study. Since the original presentation in 2016, I have revised my recommendations slightly, and created three different priority orderings. Rather than replicate the original slides, I will show the new material at the end of this document.)

Following the practice we established in Module 1, we will review the learning objectives, formulated as questions.



REVIEW OF LEARNING OBJECTIVES

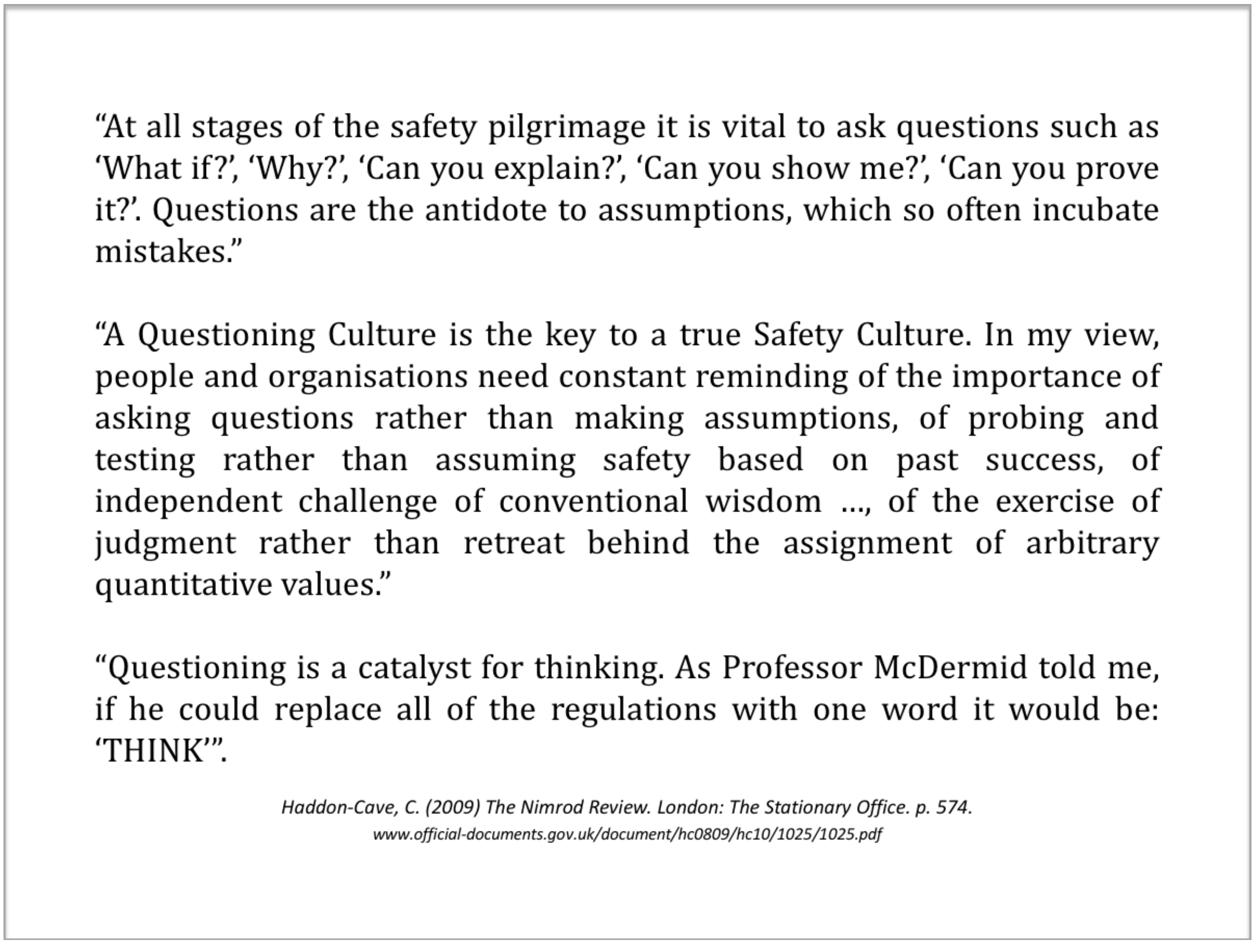
Are you able to

- ❖ Compare and contrast an assurance case approach with other approaches?
- ❖ Discuss how an assurance case approach could fit into a regulatory environment?
- ❖ List current areas of assurance case research?
- ❖ Locate references for further study?

Every man takes the limits of his own field of vision for the limits of the world. - Arthur Schopenhauer

Think to yourself how you'd answer these questions.

After you've thought about the questions for a little bit, I'll end with the superb quotation from the Nimrod report that I used back in Module 2.



"At all stages of the safety pilgrimage it is vital to ask questions such as 'What if?', 'Why?', 'Can you explain?', 'Can you show me?', 'Can you prove it?'. Questions are the antidote to assumptions, which so often incubate mistakes."

"A Questioning Culture is the key to a true Safety Culture. In my view, people and organisations need constant reminding of the importance of asking questions rather than making assumptions, of probing and testing rather than assuming safety based on past success, of independent challenge of conventional wisdom …, of the exercise of judgment rather than retreat behind the assignment of arbitrary quantitative values."

"Questioning is a catalyst for thinking. As Professor McDermid told me, if he could replace all of the regulations with one word it would be: 'THINK'".

Haddon-Cave, C. (2009) The Nimrod Review. London: The Stationary Office. p. 574.
www.official-documents.gov.uk/document/hc0809/hc10/1025/1025.pdf

If you remember nothing else from these five modules about Understanding Assurance Cases, please remember those words. If you cannot remember all of these words, then at least remember Professor McDermid's single word: **think**.

Thank you for your attention, and I'll be happy to field any remaining questions or comments about this module in particular, or the whole series in particular.

Thus ended the educational presentations.

If you have questions or comments about this material, contact its author at `c.michael.holloway@nasa.gov.`

Recommendations for additional reading.

These suggested references are intended to provide a broad overview of philosophy, principles, and practices associated with the assurance / safety case approach to obtaining confidence in the safety and efficacy of systems and services. Reading all of the suggested references will not tell you everything you need to know, but it should provide you with the knowledge that is needed to understand most everything else that you will encounter. The length of the material various considerably, from a low of 6 pages to a high of nearly 600 pages.

No single one of the references is complete in itself. Also, some of the references take points of view that are different from others. Inclusion on the list does not imply endorsement of the content.

All of the listed references except for the Toulmin book are available for free in electronic form. The lists below include URLs that worked as of 14 July 2020.

Three different suggested reading orders are provided: one for students, researchers, and the simply curious; one for practicing engineers and approval authorities; and one for managers, which contains only five suggestions.

Recommended order for students, researchers, and the curious

1. The Uses of Argument (Updated edition) Toulmin, S. E. (2003, 1958). This book must be purchased. One place to get it is www.amazon.com/Uses-Argument-Stephen-E-Toulmin/dp/0521534836/

2. The Safety Argumentation Schools of Thought. Graydon, P. J. (2017). hdl.handle.net/2060/20180000378

3. A Taxonomy of Fallacies in System Safety Arguments. Greenwell, W. S., et al (2006). hdl.handle.net/2060/20060027794

4. Current Practices in Constructing and Evaluating Assurance Cases With Applications to Aviation. Rinehart, D. J., Knight, J. C., & Rowanhill, J. (2015). hdl.handle.net/2060/20150002819

5. The Purpose, Scope, and Content of Safety Cases. Office for Nuclear Regulation (2013). www.onr.org.uk/operational/tech_asst_guides/ns-tast-gd-051.pdf

6. Arguing Safety - A Systematic Approach to Managing Safety Cases. Kelly, T. P. (1998). www-users.cs.york.ac.uk/tpk/tpkthesis.pdf

7. Reviewing Assurance Arguments: A Step-By-Step Approach. Kelly, T. P. (2007). www-users.cs.york.ac.uk/~tpk/dsnworkshop07.pdf

8. A New Approach to Creating Clear Safety Arguments. Hawkins, R., Kelly, T., Knight, J., & Graydon, P. (2011). www.cs.virginia.edu/~jck/publications/SSS.2011.safety.cases.pdf

9. The Nimrod Review: An independent review into the broader issues surrounding the loss of the RAF Nimrod MR2 Aircraft XV230 in Afghanistan in 2006. Haddon-Cave, C. (2009). www.official-documents.gov.uk/document/hc0809/hc10/1025/1025.pdf

10. The Friendly Argument Notation (FAN). Holloway, C. Michael. (2020). shemesh.larc.nasa.gov/arg/fantm.pdf

11. Regulatory Report: Chevron Richmond Refinery Pipe Rupture and Fire. U. S. Chemical Safety and Hazard Investigation Board (2014). www.csb.gov/assets/1/20/chevron_regulatory_report_06272014.pdf

12. Certification and Safety Cases. Graydon, P., Knight, J., & Green, M. (2010). www.cs.virginia.edu/~jck/publications/ISSC.2010.pdf

13. Assurance cases and prescriptive software safety certification: A comparative study. Hawkins, R., Habli, I., Kelly, T. P., & McDermid, J. (2013). www.sciencedirect.com/science/article/pii/S0925753513001021

14. Explicate '78: Uncovering the Implicit Assurance Case in DO-178C. Holloway, C. M. (2015). hdl.handle.net/2060/20150009473

15. An Investigation of Proposed Techniques for Quantifying Confidence in Assurance Arguments. Graydon, P. J., Holloway, C. M. (2016). hdl.handle.net/2060/20160006526

Recommended order for practicing engineers & approval authorities

1. The Purpose, Scope, and Content of Safety Cases. Office for Nuclear Regulation (2013). `www.onr.org.uk/operational/tech_asst_guides/ns-tast-gd-051.pdf`

2. Regulatory Report: Chevron Richmond Refinery Pipe Rupture and Fire. U. S. Chemical Safety and Hazard Investigation Board (2014). `www.csb.gov/assets/1/20/chevron_regulatory_report_06272014.pdf`

3. Current Practices in Constructing and Evaluating Assurance Cases With Applications to Aviation. Rinehart, D. J., Knight, J. C., & Rowanhill, J. (2015). `hdl.handle.net/2060/20150002819`

4. The Safety Argumentation Schools of Thought. Graydon, P. J. (2017). `hdl.handle.net/2060/20180000378`

5. Arguing Safety - A Systematic Approach to Managing Safety Cases. Kelly, T. P. (1998). `www-users.cs.york.ac.uk/tpk/tpkthesis.pdf`

6. Reviewing Assurance Arguments: A Step-By-Step Approach. Kelly, T. P. (2007). `www-users.cs.york.ac.uk/~tpk/dsnworkshop07.pdf`

7. The Nimrod Review: An independent review into the broader issues surrounding the loss of the RAF Nimrod MR2 Aircraft XV230 in Afghanistan in 2006. Haddon-Cave, C. (2009). `www.official-documents.gov.uk/document/hc0809/hc10/1025/1025.pdf`

8. Assurance cases and prescriptive software safety certification: A comparative study. Hawkins, R., Habli, I., Kelly, T. P., & McDermid, J. (2013). `www.sciencedirect.com/science/article/pii/S0925753513001021`

9. Certification and Safety Cases. Graydon, P., Knight, J., & Green, M. (2010). `www.cs.virginia.edu/~jck/publications/ISSC.2010.pdf`

10. Explicate '78: Uncovering the Implicit Assurance Case in DO-178C. Holloway, C. M. (2015). `hdl.handle.net/2060/20150009473`

11. A Taxonomy of Fallacies in System Safety Arguments. Greenwell, W. S., et al (2006). `hdl.handle.net/2060/20060027794`

12. A New Approach to Creating Clear Safety Arguments. Hawkins, R., Kelly, T., Knight, J., & Graydon, P. (2011). `www.cs.virginia.edu/~jck/publications/SSS.2011.safety.cases.pdf`

13. The Friendly Argument Notation (FAN). Holloway, C. Michael. (2020). `shemesh.larc.nasa.gov/arg/fantm.pdf`

14. An Investigation of Proposed Techniques for Quantifying Confidence in Assurance Arguments. Graydon, P. J., Holloway, C. M. (2016). `hdl.handle.net/2060/20160006526`

15. The Uses of Argument (Updated edition) Toulmin, S. E. (2003, 1958). This book must be purchased. One place to get it is `www.amazon.com/Uses-Argument-Stephen-E-Toulmin/dp/0521534836/`

Recommended order for managers

1. The Safety Argumentation Schools of Thought. Graydon, P. J. (2017). hdl.handle.net/2060/20170007188

2. The Purpose, Scope, and Content of Safety Cases. Office for Nuclear Regulation (2013). www.onr.org.uk/operational/tech_asst_guides/ns-tast-gd-051.pdf

3. Regulatory Report: Chevron Richmond Refinery Pipe Rupture and Fire. U. S. Chemical Safety and Hazard Investigation Board (2014). www.csb.gov/assets/1/20/chevron_regulatory_report_06272014.pdf

4. Current Practices in Constructing and Evaluating Assurance Cases With Applications to Aviation. Rinehart, D. J., Knight, J. C., & Rowanhill, J. (2015). hdl.handle.net/2060/20150002819

5. An Investigation of Proposed Techniques for Quantifying Confidence in Assurance Arguments. Graydon, P. J., Holloway, C. M. (2016). hdl.handle.net/2060/20160006526