Delivering quality when you have thousands of Olga runs to make

17 September 2019

Why checking is important

- Last week we delivered a complex fluid characterisation project

- Six different fields
- Six different oils or condensates, six different gases, six different waters
- Dozens of laboratory analysis reports and sampling reports
- And big gaps in the data

- We delivered a common characterisation for all the fluids, in any combination
- For subsea/topsides commingling, for flow assurance & process modelling

- 20+ hypotheticals using a bespoke equation of state
- 200+ binary interaction parameters

- All prepared by a strong lead engineer under the guidance and QA of our technical authority

- *Maybe* I should check the final report before we send it…

Why checking is important

- …the client's name was spelled wrong.

How to check

There is only one document to follow to assure the quality of any technical work: ISO 9001

Here is what it says:
- The *product* must meet client requirements.
- Design work must be *reviewed*, *verified,* and *validated*.
- Keep records.

But:
- Simulation input and output files are not products.

Not all ISO 9001 quality systems are created equal.

A good quality system

A good quality system should:

- Be readable
- Be short
- Be useful

and

- Be appropriate for the work being done

So what if you are delivering a relatively big flow assurance FEED?

Say, 10,000+ Olga runs?

How to check Olga

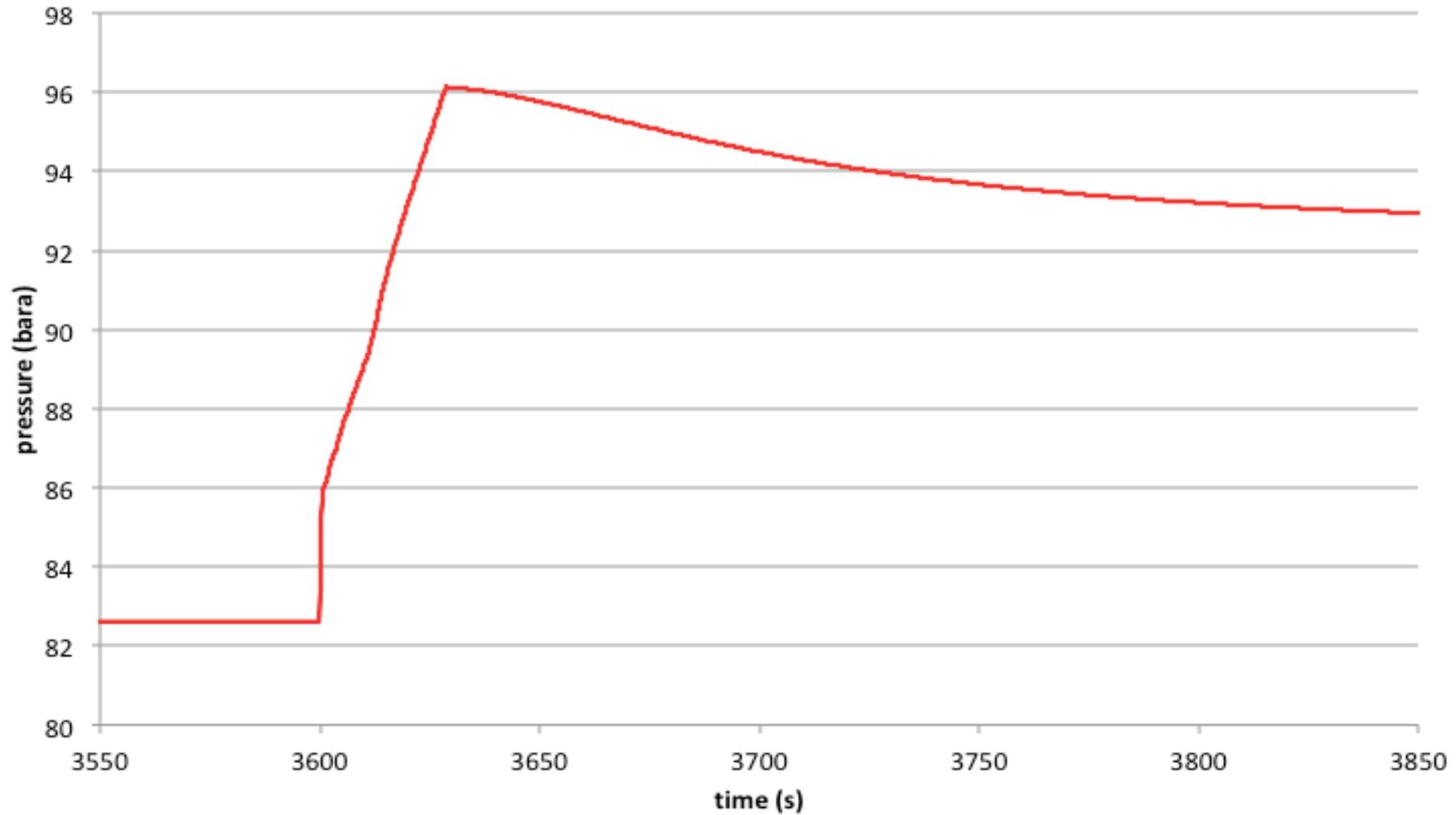- The problem: You can't manually check every model, but every result must be checked.

Start on a strong foundation
- A good basis of design, fully referenced, carefully checked line-by-line.
- A base Olga model, carefully checked against the BOD, line-by-line
    - Maybe print it out and check it with a highlighter
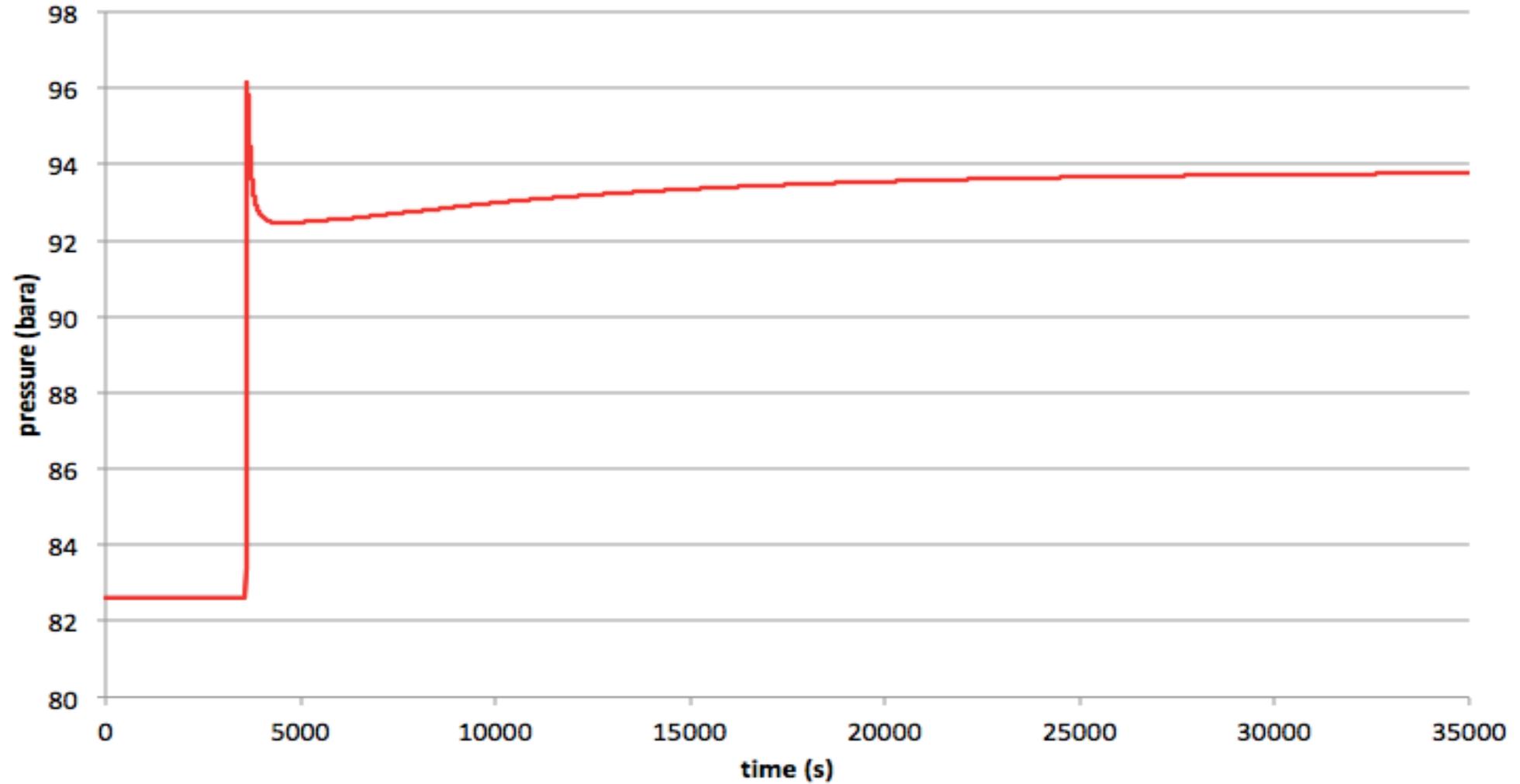   - Maybe write the input file from scratch, in a text editor

And then focus on the *product*.

Example: gas project HIPPS

- Typical HIPPS application.

- HIPPS (on the platform) to protect process from full wellhead pressure.

- Wellhead shut-in pressure > 300 bara.
- Maximum pressure in the process 96 bara.

- Flow assurance goal: what does the HIPPS closure time need to be?

- A small element of a big project.

Overall Field Development

Here is what we expected to see

But zoom out, and…

Let's do some checking

It is usually user error.

So we did some checking:
- The junior engineer self-checked the input file -> no issues
- The senior engineer checked the file -> no issues
- The lead engineer (me) checked the file -> no issues

- We did some hand calculations to make sure we were seeing the pressure & temperature rise at the rate we expected -> yes it does.

- So: the lead engineer came in on a Saturday, wrote a new Olga file from scratch, without the GUI, using the BOD only as reference – and had fun doing it – and got the same result

Our solution

We never worked it out*.

But that's okay. ISO 9001 says:
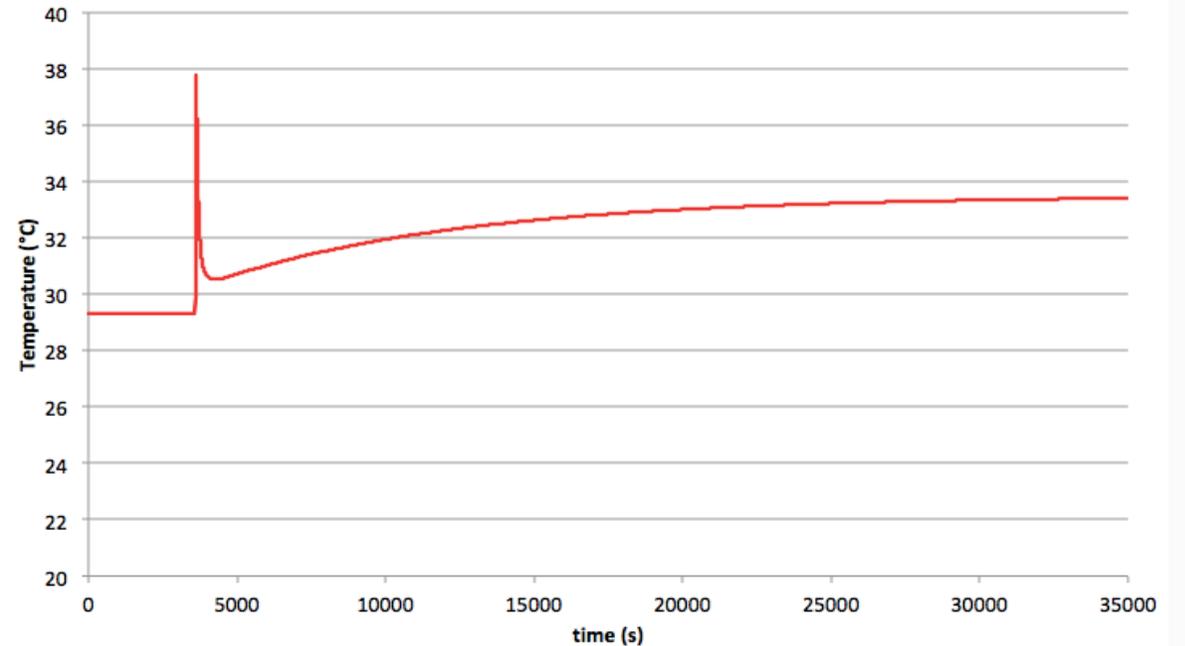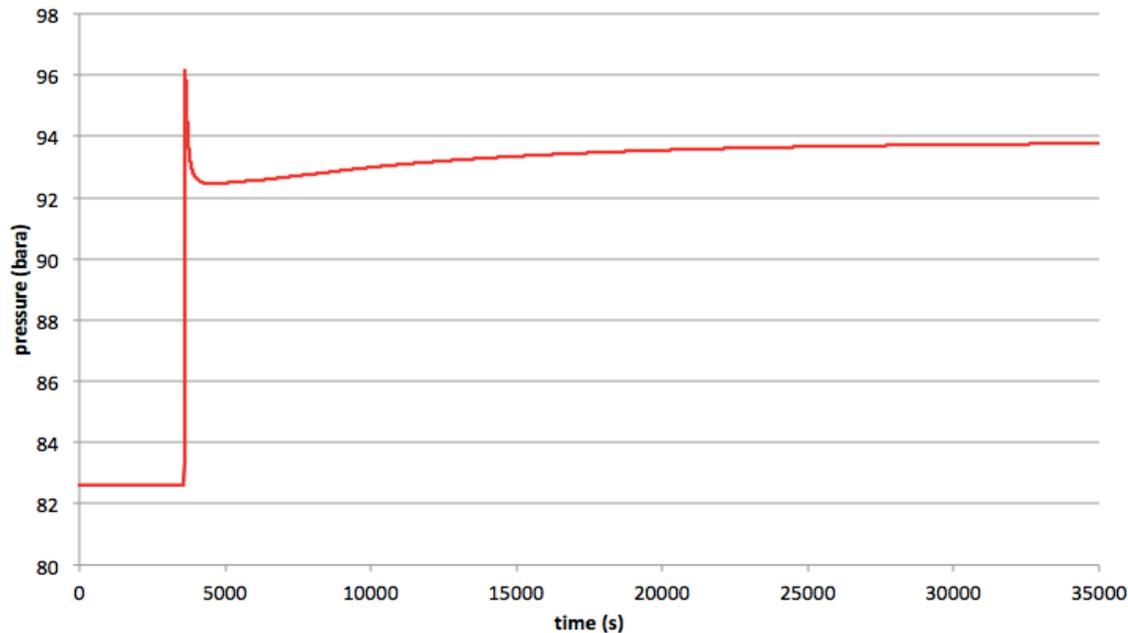- The product must meet client requirements.

ISO 9001 doesn't say:
- You're not allowed to fail.

We failed here. So we talked to the client about it, and agreed to change their requirements.

(* We did solve it, later. And maybe you can solve it, right here and now. Also: we're hiring.)

Any theories?

- We start with a steady-state model at maximum flowrate, arrival pressure 82.5 bara
    - then created a blockage 4.32 m^3 downstream of the HIPPS valves
    - initiated valve closure once the HIPPS sensed pressure reach 93 bara
    - tested closure times until we topped out at 96 bara (9 seconds)
    - then allow to settle to ambient conditions

Our guideline for quality on a big project

Fail Fast

(but don't fail often)

Why
- being wrong now saves rework later

How
- be brave
- make an assertion
  - keep it simple, and make it right/wrong
  - avoid grey areas and opinion
  - "I think that I is correct"
- Now *test* the assertion

Testing

Olga and other calculations:
- Start with the primary sources
  - *can I prove where this number comes from?*
- Create tests for Olga results
  - *if I run this case, what result do I expect?*

In documents:
- Highlight areas of uncertainty & weakness
  - help reviewers find where to test and focus
- Create tests for risky areas
  - anywhere you're not confident or there are possible contradictions
  - *numbers*: tables & charts

Communicating
- Test your beliefs with experts
- Listen, and look for areas where there are differences in factual understanding

A note on psychology

There are cultural factors that you need to consider.

- Communicating across cultures takes trust and patience. Different cultures have different ideas about what constitutes a fact.
- It's not always as straightforward as you might think.

For example: the English

- You can't tell an English person that they are wrong. The "fact" will be ignored; they will hear "I don't trust you.
- But English people love it when you tell them that you are wrong. They will think that you are honest and trustworthy.
- To test a fact with an English person: ask for help, and don't make the fact personal.

- (And don't try this approach with a Japanese client.)

An example: hydrate management

This was a major scope in a big project. The main goal is to review and recommend options for hydrate management.

We started with a set of options for hydrate management during normal operation:
- No inhibition.
- KHI only.
- MEG only.
- KHI plus MEG.

But does this set of options meet the client requirements?

We tested it.
- We informally asked the client in a meeting: "here are the options we intend on reviewing".
- We were wrong: the existing MEG system is in place but not working. We hadn't picked this up in the kick-off documentation.

An example: hydrate management

We failed, but we failed fast.

We saved a lot of time and work.

We saved being corrected later in the project, probably while delivering work that does not meet client requirements.

We built the relationship with the client by allowed them to help us do our job.

Failure is okay. And also inevitable.

Back to ISO 9001

- The *product* must meet client requirements.
- Design work must be *reviewed*, *verified,* and *validated*.
- Keep records.

Review:
- Will this work meet client requirements?
- Test it.

Verify:
- Is this result correct?
- Test it.

Validate:
- Is this product useful?
- Test it.