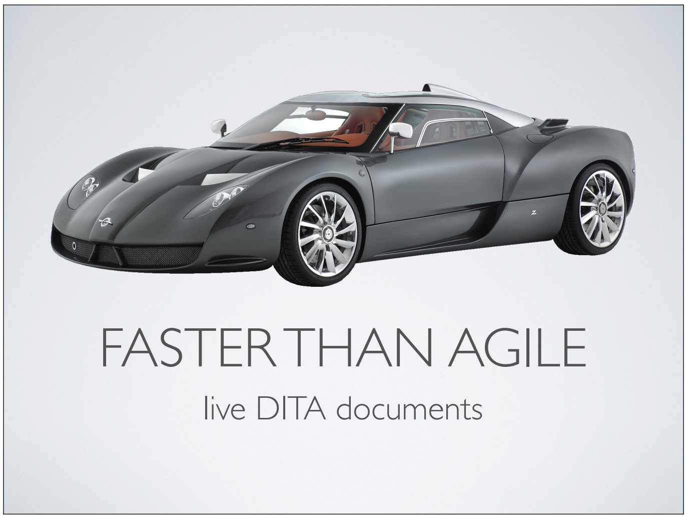


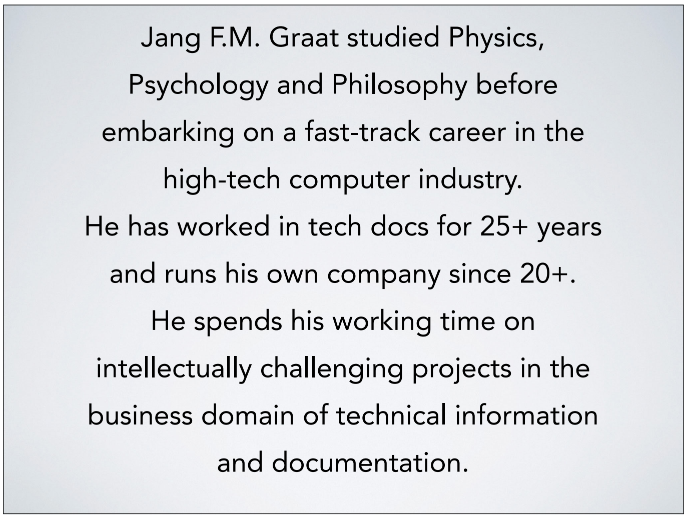
FASTER THAN AGILE

live DITA documents

Presentation for
DITA/CMS NA conference
in Chicago, April 20-22, 2015.

WHO'S TALKING ?

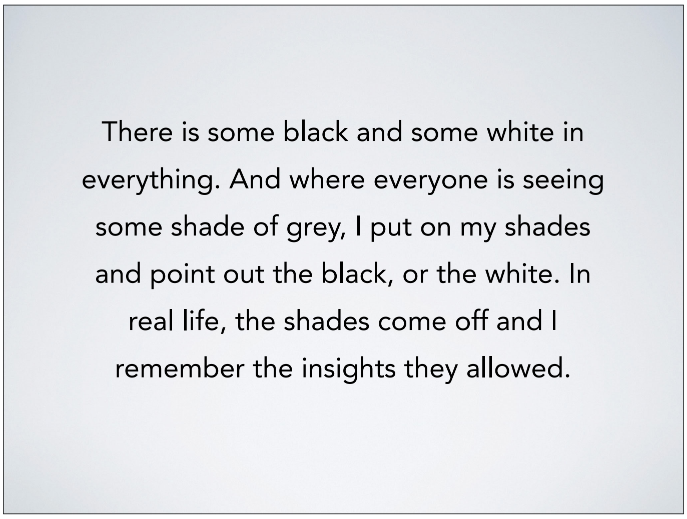
Jang F.M. Graat (geek philosopher)



Jang F.M. Graat studied Physics, Psychology and Philosophy before embarking on a fast-track career in the high-tech computer industry.
He has worked in tech docs for 25+ years and runs his own company since 20+.
He spends his working time on intellectually challenging projects in the business domain of technical information and documentation.

WARNING

this talk is conceptual and may exaggerate (a lot)



There is some black and some white in everything. And where everyone is seeing some shade of grey, I put on my shades and point out the black, or the white. In real life, the shades come off and I remember the insights they allowed.

GOOD NEWS ?

the development department has gone agile

The agile style of managing complex products is changing the way development teams work.

This has repercussions on the way the documentation is being maintained.

Not always easy.

HOW DO WE MANAGE ?

change only small parts of the full document set

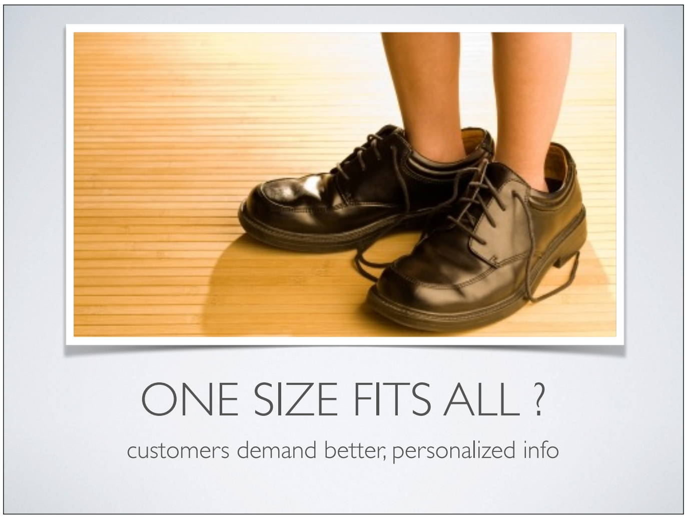
Tech docs departments are forced to follow their agile counterparts in development. Changing bits here and bits there, appointing a scrum master and adopting weekly or biweekly sprints.

Agile is cool, even if we don’t really like it.

COPY-CAT BEHAVIOR

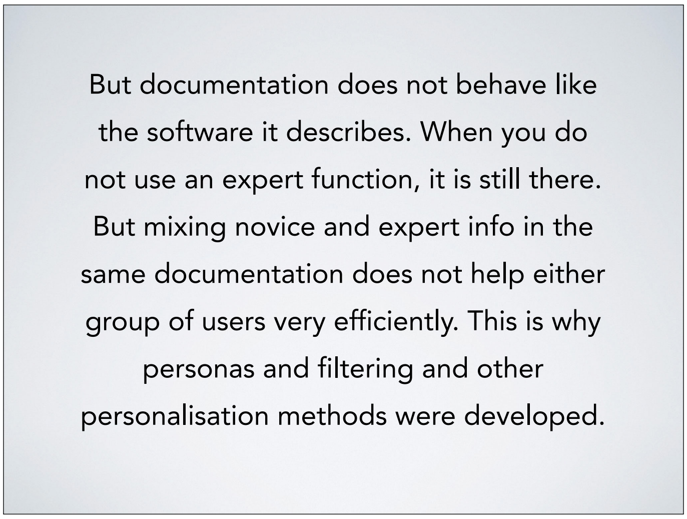
source control systems, automated builds

Both the production and the publication side of technical documentation are modelled after the strategies used in software development. Version control, automated builds, we simply do as they do, because we don’t know any better.

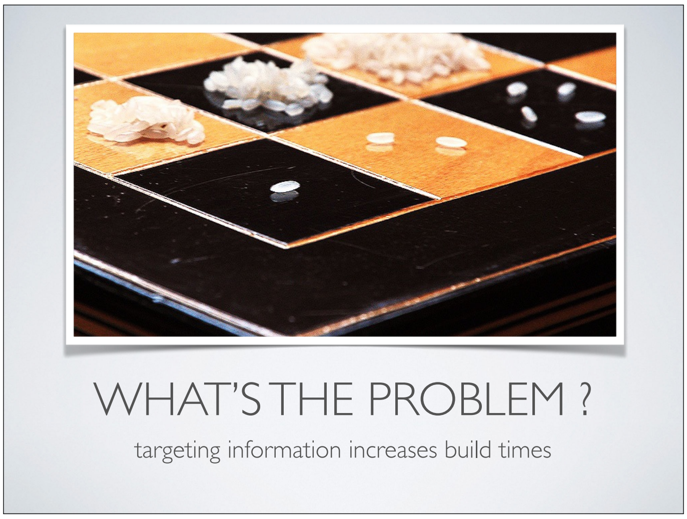


ONE SIZE FITS ALL ?

customers demand better, personalized info

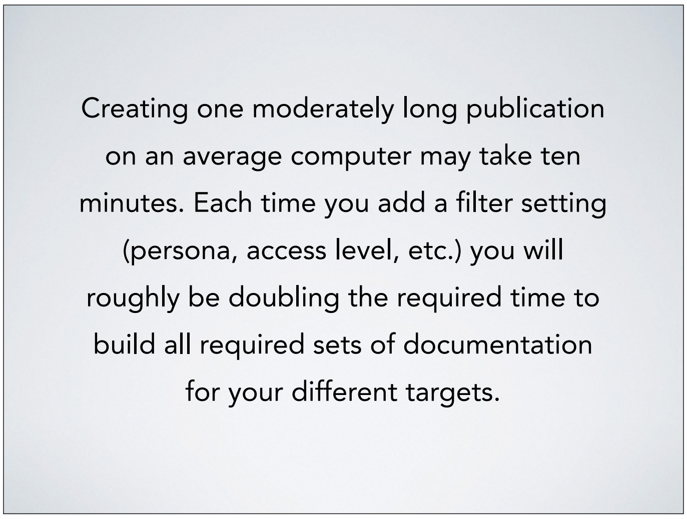


But documentation does not behave like the software it describes. When you do not use an expert function, it is still there. But mixing novice and expert info in the same documentation does not help either group of users very efficiently. This is why personas and filtering and other personalisation methods were developed.



WHAT'S THE PROBLEM ?

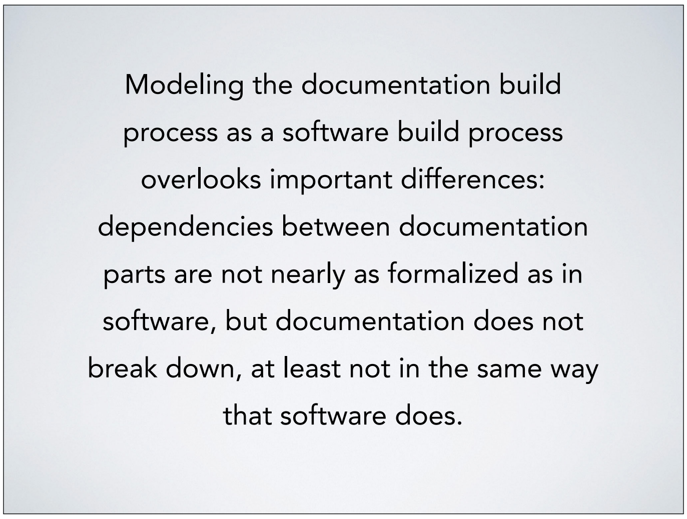
targeting information increases build times



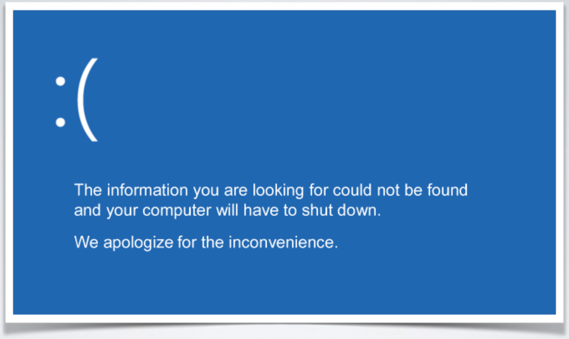
Creating one moderately long publication on an average computer may take ten minutes. Each time you add a filter setting (persona, access level, etc.) you will roughly be doubling the required time to build all required sets of documentation for your different targets.

WHAT'S NOT A PROBLEM ?

documentation ≠ software



Modeling the documentation build process as a software build process overlooks important differences: dependencies between documentation parts are not nearly as formalized as in software, but documentation does not break down, at least not in the same way that software does.



:(

The information you are looking for could not be found and your computer will have to shut down.

We apologize for the inconvenience.

INFO DOES NOT CRASH

users can handle missing or outdated info

Even though a broken link or outdated information may look dramatic, it is not the end of the computer's or user's world. Most users will be able to identify the error and find information using one of the other available channels (e.g. social media or another query).

INFO IS NOT AN ISLAND

always fragmented and connected

Information is always fragmentary - you cannot describe everything in the same topic. And because of this, it is also connected in many different ways. One piece of info leads to many others, inside and outside of your documentation set.

AGILE OPPORTUNITIES

rethink the publication process

The special nature of technical information allows methods that would not work well for software development. Let's see what happens when the main premise - you have to publish all topics to create a consistent new release - is discarded.

DITA AS INFO STORAGE

well-organized, semantically rich content

DITA - as well as other XML standards - not only offers a basis for authoring but can also be seen as information storage. Indeed, there are several open source systems that can use any XML file as database: semantically rich info storage.

PUBLISH ON DEMAND

transform to whatever format is requested

Publication is transforming XML-based info to the required output format. Much of this is done by XSLT, which runs on personal computers as well as all modern web servers. Whether you do one topic or a whole bunch, that does not make a real conceptual difference. So why not do it at the moment when it is needed.

SERVER-SIDE FILTERING

ultimate single-sourcing

This allows optimising the material that is passed to the user, by running the XSLT that gives the best result for each individual user on each particular device. After all, that information is available in the request that is sent to the web server.

LEAN PUBLISHING

only produce what is needed

The pre-processing phases (resolving keyrefs, conrefs) can be performed when saving the topics to the server. The final step in the publishing process can be done when the topic is requested - when all filtering info is known. Only what is needed right now right here will be passed to this individual user.

ROUND-TRIPPING ?

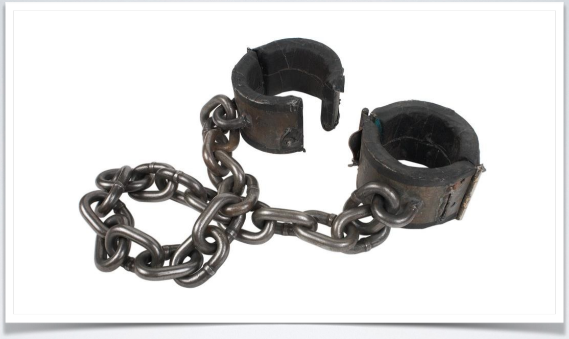
modern browsers allow in-situ editing

Another issue in documentation that does not resemble software development: a reader may be an author, too. With HTML5, most content can be made editable in standard browsers. This allows reviewing via browsers, saving changes to the same sources that are authored in specialist software by the tech docs staff.

OLD SCHOOL

editing in and publishing from CCMS database

The old method of creating documentation from a CCMS, with or without agile sprints and scrums and automated builds, means that everything is published again and again, even when changes are very small. A heavy-weight process that is not scalable by definition.



DEPENCENCIES

built-in editors, CCMS-specific publishing chains

Apart from the bulky publishing process, the average CCMS means you are locked into that system and the cost of moving to another publishing chain or storage system can be so high that this will never be a viable choice anymore.

NEW SCHOOL

live DITA docs, publish on demand, edit anywhere

Using DITA documents as XML storage changes the nature of your CCMS. With the XML documents getting transformed on demand and editing capabilities in browsers for reviewing or crowd-sourcing, scalability and flexibility are obvious.

QUESTIONS ?

Check out my (live) website

www.ditadocs.com

or send mail to jang@jang.nl