

# The single-source mindset

## Lessons from a favourite childhood game

Gary Calwell

As a child, did you ever play a game of Chinese Whispers? If not, let me set the scene.

Assemble a small group of children and ask them to form a line. Ask the first child to think up a thirty-second story and then whisper it to the second child. The second child whispers the story to the third child and so on until finally the last child writes down the story as they heard it.

What are the chances that the final story will be in any way similar to the original? That's right: almost zero.

If we explain that the objective is to keep the story as accurate as possible along the chain, you would expect that after several attempts the quality of their communication would improve. Now let's make the game more interesting. Offer to give each child a sweet for every minute that they play the game. The children now have two competing motives: a team goal to finish the game quickly by keeping the story accurate and individual goals to prolong the game as long as they can get more sweets.

What does this game have to do with technical writing and the single source mindset? More than you might think. When these children become adults, they might find themselves working on a large project. If so, chances are they will get to play a grown-ups' version of this game. The whispers might be replaced by spreadsheets, status meetings and reports—and the sweets might be replaced by daily rates and contract fees—but the competition between team goals and individual goals often remains.

In Chinese Whispers there are many versions of The Truth, each represented by each child's version of the story. This is also true of many large projects, where different teams work in silos to create their own "story" and try and keep all the stories aligned by "whispering" to the other teams. Let's look at a real-life example.

A company is attempting to develop and deploy a new customer management system. We'll count how many versions of The Truth there are in this multi-million dollar project. The business analysis team writes design documents that are a mixture of technical information and end-user procedures (1). Since there are no defined standards, the output predictably is very inconsistent. The technical writing team uses the design documents to create a set of user manuals (2). Since some of the information in these manuals is not suitable to be used in training, the training team creates a set of training materials (3). The testing team creates a set of test scripts (4). Finally, the operations team creates a set of work instructions (5).

We now have five sets of information and five versions of The Truth. Each set of information is designed for a different purpose but there is considerable duplication of content. If you need any proof of this, just look at the most common authoring process: copy from one document set, paste into another and edit as required. In theory, a process like this can work. Information can flow from one silo to the next through well-defined entry and exit gates. In practice however, this process usually fails, for it is based on two myths:

- Myth 1 is that information flow is one-way. Authoring systems are designed so that information flows from one document set to another. In practice, there are many feedback loops as teams make design changes and fix errors. Having feedback and rework loops in a single set of documentation is hard enough to manage, but try doing this with five sets.
- Myth 2 is that documentation will always be kept up-to-date. However, it is usually difficult to get others interested and motivated enough to keep one set of documentation up-to-date. If a project team is struggling to achieve its milestones, the task of updating documentation is often pushed down the priority list, especially if there are multiple sets of documentation to keep track of and maintain.

The concept of single-source documentation is simple enough. The content in one set of documentation can be filtered, sorted and formatted in different ways so that it can be used for multiple purposes: for example, a user guide, technical manual and training guide could be produced from a single document.

Many people think of single-source as a software feature rather than as a process of eliminating content duplication regardless of the technology used. If you visit the web sites of help authoring tools, for example, you will see that a “feature” of their software is its ability to produce multiple output documents from a single set of source documents.

However, you can achieve a single-source solution using the least sophisticated of technologies. Imagine a group of product managers who need to give presentations about a new product to different audiences. Using a blank sheet of paper and a pen, they could discuss, clarify and write down the key points of the presentation. Using different coloured highlighters they could then mark which points were most relevant to each audience. There you have it. A single-source solution for under \$10!

Single-sourcing is a mindset, not a software feature. It’s about allowing a group of people to collaboratively work on a single document set that can later be used to meet all the project’s downstream information needs.

So how could we apply these principles to the earlier real-life example? The first step is to clarify the purpose and audience of each document, eliminate duplication and use hyperlinks to cross-reference rather than copy and paste. We could also encourage greater collaboration between the teams. For example, there is often similarity between the scenarios used for testing and training exercises. Why not use the same scenarios for both purposes?

If you think in these ways you might come up against the limitations of your authoring tool (Microsoft Word 2007 in our real-life example). If so, you may be able to use the automation features of your authoring tool to make it easier to do repetitive tasks. If you want people to change their behaviour, it’s much easier if the new way is easier for them than the old way. In our real-life case we used Visual Basic for Applications (VBA) macros to add new functions, such as:

- A new hyperlink command that allows users to link to any topic in a user manual without the time-consuming need to manually insert bookmarks in the destination document.
- Locking all documents—other than tracked changes—so that users don’t corrupt the document’s styles and layouts. When an authorised user (from the technical writing team) opens the document a macro automatically unlocks the document and locks it again when the document is saved. This provides a reasonable level of security but, more importantly, allows everyone to collaboratively work on the same document set.
- Adding text tags to the Microsoft Word documents so that they can produce both facilitator guides (containing facilitator notes) and participant guides (excluding participant notes).

Customisations like these take some time (and therefore money) to build and maintain and so it is important to do some form of cost–benefit analysis. For small projects with a small team it is probably more cost effective to continue to do things manually. But for a large and complex project, and a large team, automation can result in substantial benefits including reduced development time, better quality and greater consistency—not to mention dramatically reducing everyone’s stress levels.

So, the next time you walk out onto the project floor, think about to what extent the different teams are really embracing single-source. Listen carefully for the whispers and look for duplication in your documentation sets. Once you become aware of these problems, it’s then possible to act on them. In doing so, we can add real value to the project and help others see that technical communication really can be an asset to the project rather than a liability.

---

### **Gary Calwell**

Gary Calwell is the director of Claricom Consulting, a Melbourne-based consultancy specialising in process improvement, documentation, training and Microsoft Office customisation. He can be contacted at [gary.calwell@claricom.com.au](mailto:gary.calwell@claricom.com.au).