

How many technical writers should we have in our organisation?

Thursday, 03 April 2003

Introduction

We were asked recently if we knew of any research on 'standard' ratios between developers and technical authors. We decided to carry out some research and this article covers our preliminary findings.

The problem with "per page" ratios

Our initial thought was that the number of writers really depends on how much needs to be described to the user. The well-known industry standard ratios approach it this way. They ask:

- Which screens and tasks need to be documented?
- How many pages need to be written to describe each screen or task?
- How many pages per day can be produced?
- How many author days do we need?

This may be fine for user documentation where a lot of development can go on behind the scenes. However, if you are describing everything, i.e. producing systems documentation, administrator and reference guides, then ratios based on the number of user screens and user tasks in an application may not help you.

Planning writing resource at the beginning of an IT project

The standard user-based writer ratios may not help you if you are at the beginning of a IT project, as the project may not be defined sufficiently to apply them. Software development methodologies can't help either. According to Assure Consulting <http://www.assureconsulting.com/workplace/writer.shtml>,

"The widely adopted software development processes - the spiral and the waterfall models - make no reference to documentation, underscoring low priority of technical writers in the development cycle. Proportionately fewer test case have been evolved for documentation than development. "

"Ideally documentation aimed at the end-user should receive 20 per cent of the product development cycle time but most writers admit that less than five per cent of time is budgeted for documentation and reviews. "

Our survey

We decided to carry out our own survey into the current trends in technical communication. Our PRELIMINARY findings (based on the first 100 responses) indicate:

- A ratio of 12 developers per writer for organisations with fewer than 500 developers and where developers do not produce user assistance documentation.
- A statistical correlation of 0.78 between the number of developers and the number of writers.
- Little correlation between the number of total employees in an organisation and the number of writers.

Past studies

There seems to have been very little research into any industry norms in recent years. In a discussion thread <http://www.raycomm.com/techwhirl/archives/0109/techwhirl-0109-00488.html> on the RayComm Web site it seems that a mainframe manufacturer carried out some research in the 1960s. Their survey showed:

- For large projects at large companies, one writer could support 3 to 6 programmers
- Their methodology enabled one writer to support 9 programmers.

All their documentation was online and appeared to cover system documentation only.

A more recent, informal study <http://www.raycomm.com/techwhirl/archives/9607/techwhirl-9607-00014.html> was carried out in 1996. This indicated

- A ratio of 10-12 developers per writer seemed to be the norm.

There were some variations - "beacon" IT companies had a ratio of 7 developers per writer; others had 20 and 50 programmers per writer.

There was also a study of 35 organisations from 1998

<http://www.hci.com.au/hcisite2/articles/mis98Srvy.htm> that showed

- An average ratio of 8 developers per writer.

Tekom, the society for German technical communicators, has carried out some more recent research in Germany. Their survey, carried out during 2002/3, showed

- An average ratio of 6 developers per writer in Germany.

So are our figures accurate?

There are a few points to note:

- Could it be that others, apart from the technical authors, are writing the documents? In most cases they are not. When asked "Does your organization use developers (e.g. engineers, programmers) to produce user assistance documentation?", only 25% of the respondents answered Yes. We have excluded these from our analysis.
- Because we doubted people would know exactly how many developers were in their organisation, we asked people to select from a range: 1-10, 11-20, 51-100, 101-200, 201-300, 301-500 and 501 or more. We used the mid-point in each range (5.5, 15.5 etc) to analyse the results, and this approach could have introduced some inaccuracy. We minimised this inaccuracy, in part, by omitting the figures for 501 or more developers from our analysis (because we couldn't determine an accurate mid-point).
- The survey is still running, so these figures may change as we get more responses.
- Whilst recognising its weaknesses, we believe our survey has produced some meaningful results.

The optimal ratio

These findings only tell us how many writers per developer there are, not how many there should be. In our opinion, the optimal ratio is almost certainly less than the actual ratios shown in these surveys. A number of the articles mentioned above indicate an optimal ratio of 5 - 7 developers per writer, although we couldn't find any statistical evidence to back up these claims.

Perhaps this optimal ratio was determined by organisations looking back at a project and calculating how much resource they should have allocated. Maybe this ratio has come from calculating a "per page" ratio for all documents at the start of a project.

Conclusions

We can make the following conclusions:

- Many articles quoted an optimal ratio of 5-7 developers per writer, though we couldn't find any statistical evidence to back this claim.
- The PRELIMINARY findings from our survey show an actual ratio of 12 developers per writer for organisations with fewer than 500 developers and where developers do not produce user assistance documentation.
- Based on anecdotal evidence of the optimal developer/writer ratio (i.e. that used by the most well-regarded IT companies), it would seem that many organisations with fewer than 500 developers could be allocating half the documentation resource they should to their IT projects.
- If we accept that a technical author provides benefits to an IT project, then it seems that the most common software development methodologies should take account of and plan for technical authoring.

Cherryleaf will be producing a report on the final results from our survey this summer. If you want to find out the main results from our survey you will need to subscribe to our free newsletter. Contact us if you are interested in finding out how Cherryleaf can help you address these problems.

What do you think?

We welcome your comments. Let us know what you think.

Survey

This summer, Cherryleaf will be producing a report on our survey into technical communication. We'll be looking at how many people are using these trends today. To participate, please complete this survey <http://www.surveymonkey.com/s.asp?u=55967177787>".