

## Metrics Captured From Documents by Using VisibleThread

Fixing requirements errors eats up roughly one third of your software project budget, specifically requirements errors consume 28% to 42.5% of total software development costs (Hooks and Farry, 2001)<sup>1</sup>. If you're budgeting \$500,000, you're spending about \$150,000 fixing defects that originate in your requirements and IT specifications.

Further, finding and fixing defects outside their phase of injection can cost you 400 times as much. It might cost \$25 to \$100 to find and fix a requirements error during the requirements phase. But if you don't find that error until the product is transitioned to the customer, repairing it can cost you \$8,000 to \$10,000 (Reifer, 2007)

Over 90% of business users leverage MS Word as their primary working environment (Forrester 2008)<sup>2</sup> and the vast majority of information for medium-large programs / projects is contained in MS Office artifacts, typically MS Word documents. Validating and inspecting these documents for quality is both time consuming and often an ad-hoc process.

VisibleThread automates the validation of IT specifications in real time using metrics along these dimensions:

### COST SAVINGS

These savings are realized immediately when the document is uploaded to the VT server vs. manual checks.

1. **Structural Completeness/Alignment (%)** - how well a document rates from the perspective of the categories of content expected for that type of document.

For example, an IT specification document may require categories of content such as 'Data Performance', 'Security', 'Uptime' etc.

The % metric reflects the degree to which these categories of content are found to be present. The lower the score the higher the risk. The fallout of a low % rating in this metric typically results in downstream rework, especially from a design/build perspective.

Using reference templates, customizable by project style (Waterfall, Iterative etc.) and type (Web Interface, Data Intensive etc.) VisibleThread provides a percentage score based on how well the document adheres to the templates appropriate for that type of project.

2. **Text Ambiguity (Quality %)** - how well a document measures from the perspective of linguistic ambiguity, that is; defects occurring in the document(s) due to the presence of 'non-actionable', vague or imprecise language. Examples of vague language include; 'appropriate', 'high', 'low'. VisibleThread highlights these words and scores accordingly.

Low % levels in this score indicate loose and open-ended language and a high defect rate per page. The consequence of open language if left undetected after sign off is a large contributor to rework costs and scope creep.

---

<sup>1</sup> Hooks, Ivy F., and Kristin A. Farry. 2001. *Customer-Centered Products: Creating Successful Products through Requirements Management*. Amacom

<sup>2</sup> The New Business Analyst, Forrester Research, April 2008.

## COST AVOIDANCE

These potential savings are realized during the lifecycle of the projects.

3. **Section Quantity/Size** – How large are particular structural areas within a document? What size are documents relative to each other?

Certain structural sections of IT specification documents are critical to effective build and deployment. For example, a low size evident in the Performance requirements section (under non-functional) will indicate considerable risk of a system that will not scale and will contribute to post deployment problems, often the worst type of problems that may undermine the business.

Size becomes critical as a project comes close to an analysis cycle sign off phase gate. VisibleThread highlights sizing using both a specific number and a color visualization within a thumbnail of the document.

The second element of size is overall size of certain document types relative to each other. The classic example is the relativity between a typical functional requirements document such as a BRD (Business Requirements Document) vs. a Test Plan. We would expect that to be between 1:5 and 1:7 in terms of overall document size. More specifically the same ratios would be expected between a use case document and a test case document.

VisibleThread provides project level thumbnails showing a visualization of all documents and how they relate size-wise to each other. Expected variances can be easily spotted.

4. **Change over time** – how much or little change is occurring in a document and who is making those changes?

Documents adhere to a lifecycle. During early stages of a normal, non-agile project we expect a high amount of 'churn', post signoff phases (in classic SDLC scenarios) are accompanied by a suitable change control and impact analysis conducted for any change along with re-estimation.

VisibleThread tracks all changes to a document along with stakeholders. This includes; document edits, document reviews, modification of associated Best Practice etc. It provides a full change history log.

In short, reviewing change to documents helps access whether the project set of documents is being modified as expected and whether the right parties are actively involved.

5. **Frequency Distribution across documents (Discovery)** – For any project, there exist domain specific concepts. For example an automated trading system in a banking context, will likely be dealing with concepts such as; 'trade', 'account', 'volume', 'dealer'. Business rules are expressed both in BRDs (Business Requirements Documents) and we expect a 'traceability' to more detailed technical specifications.

A distribution of certain key terms across relevant document types would be expected. For instance, a test document that either does not refer to a 'trade' or has low distribution of the term 'trade' would be considered highly suspect and likely has a high defect rate.

VisibleThread automatically calculates the frequency of all major concepts in the system and offers an accompanying graphical view of the thumbnails for each document highlighting the exact distribution of the term under review.

### **ABOUT VISIBLETHREAD**

VisibleThread helps corporate IT departments create superior program / project documentation leading to successful project delivery. Our document structure and quality analysis tools, combined with the ability to create tailor-made best practices documents, provide customers with the insight and metrics they need to make better decisions throughout the IT project lifecycle. VisibleThread ensures a uniform approach to IT documentation resulting in consistency across documents, higher quality outputs and lowered cost.