

Vetting documents for quality – the case for automation

THE STATE OF PLAY TODAY

Most phases of the IT lifecycle are measurable. Systems architecture and design metrics are reported in a wide range of modeling and profiling tools, code test coverage levels are reported by automated test tools, build metrics are core to Configuration Management solutions. Measuring quality at all stages of the lifecycle lowers risk and results in more robust and reliable systems, yielding value to the business. In fact, quality metrics are a critical success factor for any organization that relies on software development as a strategic driver of competitive advantage, and sustained profitability, and risk management.

However, quality metrics remain conspicuously absent during the early phases of the IT lifecycle where initial business and system requirements are captured. Indeed, there is little or no systematic measurement applied to content stored in MS Office (MS Word and Excel). Applying measurable benchmarks at this stage of the IT process has a disproportionately positive effect on the likelihood of successful outcomes. Put simply, catch an issue earlier and you avoid downstream problems and rework costs. Yet the fact that we have at best ad-hoc manual reviews, points to a largely untapped area of potential efficiencies in the IT lifecycle.

Over 90% of business users leverage MS Word as their primary working environment (Forrester 2008)¹ and the vast majority of information for medium to large sized 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.

THE COST TO THE BUSINESS

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)². 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). In some cases it can cost a company's reputation (Toyota Prius, brake software).

AUTOMATION AND METRICS

In many organizations, Centers of Excellence, PMOs (Program/Project Management Office) and PQA (Process & Quality Assurance) teams use a variety of manual techniques to vet documentation that are time consuming and manual; leaving room for unintentional mistakes, missed steps and delays in catching errors with regards to governance and best practices. In the spirit of delivering the project on time and under budget, many times these quality review processes are rushed and reduced to cursory checks. Like ensuring documents exist with the right naming convention, rather than reviewing the internal contents of documents and ensuring the contents are of high quality.

¹ The New Business Analyst, Forrester Research, April 2008.

² Hooks, Ivy F., and Kristin A. Farry. 2001. *Customer-Centered Products: Creating Successful Products through Requirements Management*. Amacom

A real-time, automated solution is required that:

- validates IT specifications immediately and gives corrective guidance to document authors
- uses objective metrics to highlight areas of risk to project teams
- improves the quality of IT documentation by pinpointing linguistic 'defects', that if left unaddressed lead to rework
- can support 'center of excellence' initiatives by providing central control of SDLC templates and checking for adherence by project teams in real time.
- can be applied leveraging existing processes and tools
- analyses text content of documents for quality metrics as they are being saved thus providing early insight to authors
- supports impact analysis by identifying 'key concept' dependencies across documents

Consistent and impartial metrics cannot be created for all project documents without an automated solution. Many factors come into play which can affect the creation of metrics manually; the size of the project, the size of the document, the mood of the reader, the time pressures to deliver, the writing style of the author, etc. An automated solution is required that will validate IT specifications in real time using metrics along these dimensions:

SPECIFIC METRICS REQUIRED

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.) the solution will provide 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.

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. An automated solution must highlight sizing using both a specific number and/or color visualization within a thumbnail representation 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.

An automated solution must provide project level thumbnails showing a visualization of all documents and how they relate size-wise to each other. Expected variances should 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.

An automated solution must track all changes to a document along with stakeholders. This includes; document edits, document reviews, modification of associated Best Practice models etc. It should provide a full change history log.

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

5. **Discovery - Domain Concepts frequency Distribution & 'cross-cutting' across documents**

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 in BRDs (Business Requirements Documents) and related documents.

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 a low distribution of the term 'trade' would be considered highly suspect and likely to have a high defect rate.

An automated solution should calculate the frequency of all major concepts in the system and should offer an accompanying graphical view of the thumbnails for each document highlighting the exact distribution of the term under review or appropriate visualization. Additionally, we want an ability to visually identify 'cross-cutting' of these domain concepts for the purposes of impact analysis and what-if scenario discussions (what if I change the business rules for a specific concept?).

CONCLUSION

The automation of the capture of data and subsequent analysis is vital to the continuing operations of a highly efficient organization. Metrics demonstrate where to increase or reduce resources early in the lifecycle. Further, metrics help to identify the impact of change caused by regulation or otherwise across projects; does it make documentation or even a project longer to produce? With metrics available in real-time, senior management will be able to make decisions faster and early in the application lifecycle, averting cost of late stage defects.

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.