

Managing Software Development Quality through KPIs

Whitepaper



Table of contents

Abstract	3
1. Introduction	4
2. Overview of software development KPIs	5
3. Determining what is important	6
4. Technical challenges	7
5. Communications and politics	8
5.1. Handling politics	8
5.2. Simplicity	8
5.3. Communication of results	8
6. Internal implementation at Verax – an overview	9
7. Results	11
8. Sources	11

Abstract

Managing software development quality and productivity is a challenging task. Organizations usually focus on source-code analysis only (lines of code, number of unit tests, coverage analysis) and neglect the not-so-obvious information such as, for instance, project quality as a function of test team efforts. One challenge here is the effort required to obtain such reports, another is to **integrate data from various sources**, such as:

- Source code management systems,
- Bug tracking,
- Document management,
- Time sheets,
- Test case management and others.

This paper presents a more holistic approach to software quality management. It advocates the systematic monitoring and ongoing analysis with an efficient business intelligence tool instead of occasional audits, which are time-consuming and selective by nature.

Analyst opinions

According to Gartner (results presented at the UKSMA conference in 2006), use of software metrics and KPIs can make a significant difference. On average, software companies spend 60% of their IT budget on 'running the business' and only the remaining 40% on 'growing and transforming' it. However, those using measurements can cut their IT expenditures on 'running the business' to less than 50%, thus leaving more than 50% for 'growth and transformation' which can make a significant difference to their business.

Intended audience

This paper is a publication created by Verax Systems' experts and specialists. Its purpose is to highlight the most important issues of software development project controlling to:

- Software development managers,
- Project managers,
- Quality managers,
- Executive management.

1. Introduction

In challenging economic times, the pressure on software development companies increases. Clients expect shorter delivery cycles within smaller budgets and without compromising the quality. As a result, software vendors have to become more streamlined and efficient.

At Verax Systems, we have managed to address these market challenges by implementing Key Performance Indicators (KPIs) embracing various aspects of **software production and the delivery cycle** (not only software design and coding) such as:

- Number of project issues and their resolution cost.
- Product development progress and the number of reported/possible enhancements.
- Testing progress, e.g. number of test cases developed.
- Product documentation, e.g. project knowledgebase growth.
- Relation of quality achieved as a function of efforts invested.

We looked for more complex correlations, for instance: how does having better product design documentation affect the number of issues in the newly implemented features? The details on how the software development KPIs were structured internally at Verax Systems are presented in section 6.

2. Overview of software development KPIs

The textbook KPIs can be divided into the following main categories:

- **Performance** – targeted at measuring performance of individual software developers, project or product development teams and technologies (including different platforms and programming languages, as well as development tools and application architectures). The main challenge lies in identifying proper metrics for the scope of work. Different approaches use the following values: lines of code, functional points, use case points, user interface screens, etc.
- **Quality** – Measuring quality aims at quantifying the defects detected in software. The number of defects is normally weighed with their gravity (or with the effort needed to resolve them) and related to the total development effort.
- **Project management** – The selection of KPIs depends strongly on the methodology applied. Most typically, Earned Value Management is recommended (e.g. by Project Management Institute). This method combines the measurements of scope, schedule and cost in one unified system. Within the system, the Planned Value (PV), the Earned Value (EV) and the Actual Cost (AC) of the completed work are compared. As a result, a number of indexes such as Schedule Variance ($SV = EV - PV$), Cost Variance ($CV = EV - AC$), Schedule Performance Index ($SPI = EV / PV$) and Cost Performance Index ($CPI = EV / AC$) are computed and more complex values are derived based on them: Cost Schedule Index ($CSI = CPI * SPI$), Estimate At Completion (EAC), To-Complete Performance Index (TCPI) and others.

A summary of the most popular KPIs is presented in the figure below:

Typical software development-related KPIs	
Performance	<ul style="list-style-type: none"> • Lines of code per day • Average screen programming effort • Average functional point programming effort • Average use case programming effort
Quality	<ul style="list-style-type: none"> • Number of defects per line of code • Number of defects per functional point / use case point • Percentage of bugs found in-house • Percentage of bugs reported by the client • Number of defects in production • Number of critical bugs compared to the total number of bugs • Number of defects created per man month of effort • Percentage of hours used for fixing bugs (in relation to the development effort)
Project Management	<ul style="list-style-type: none"> • Schedule Variance (SV) • Schedule Performance Index (SPI) • Cost Variance (CV) • Cost Performance Index (CPI) • Cost Schedule Index (CSI) • Estimate At Completion (EAC) • To-Complete Performance Index (TCPI)

Table 1: Typical software development-related KPIs.

3. Determining what is important

As your organization will try to meet the required KPI levels, it is essential to define them correctly. The number of bugs in the product is a good example. If developers are awarded for entering them, they will log anything as a bug. Introducing performance measuring into the organization requires a proper information campaign and the defined KPIs may need amendments as they are in operation.

At Verax Systems, we decided to **focus on the areas that required most improvement first**. In most software development companies these areas are **not in the coding** (the manager knows who is a good, productive developer and who is not doing well) – the shortcomings come in different fields, for instance:

- **Documentation** – how well is the software documented? Typically there are HLDs and LLDs but are they really produced and updated? Who are the best documenters?
- **Testing** – how good are we at testing? Typically developers would immediately think about unit and system tests. However it is a much broader subject:: how many test cases have been created for the software, is the test plan improving? At what rate? Is the test density (number of cases per lines of code) and coverage increasing or decreasing? What’s the high level view?
- **Product improvement** – it is important to build a culture of continual improvement (e.g. if form X was arranged better, the user experience would be improved), however it has to be a controlled process. Product or software enhancements should be collected and accounted for. Who are the greatest contributors? Is there room for product improvement? Are there many suggestions reported by clients?
- **Enforcing up-to-date information in supporting systems** – a classic example is bug tracking. Typically, a bug lifecycle is similar to “assigned” (to developer to resolve) » “fixed” (now testing has to verify) » “verified” (QA team makes sure that it works). The problem is that “fixed” bugs tend to be forgotten about: either not even changed to “fixed”, or not “verified” (this was the case of 25% of bugs at Verax Systems). Deploying KPIs helped to enforce the process and close all the “stalled” issues in about six weeks time. Another good example is time tracking.
- **Time tracking.** Timesheets (especially in R&D/product development organizations) are the most underestimated monitoring tools. In an advanced tracking system, it is possible to drill down on each activity. This gives a wealth of information (far better than per-cost centre accounting reports). For instance, there are only a few organizations that can answer the following question: “how does investment into quality assurance activities correlate with number of issues, and customer reported complaints?”. The answer to this and similar questions is invaluable for any management team.

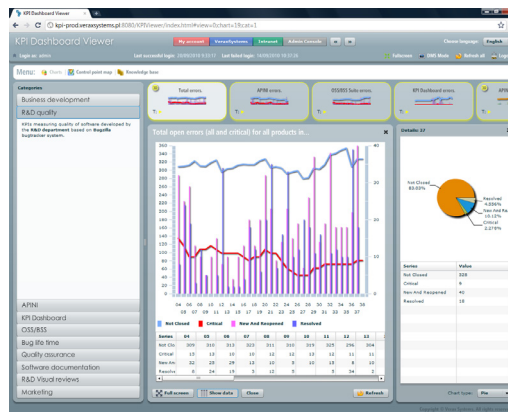


Figure 1: Total number of open issues in all Verax Systems product in the KPI Dashboard.

4. Technical challenges

Obtaining good KPIs sounds easy, however there are a few technical challenges associated:

- **Lack of historical data.** Most KPIs require visibility not only into current data, but also (if not most importantly) about the past. Current reports can be obtained from most systems very quickly, however only a few can show past status and trends. For instance, Bugzilla can report on current bugs, but information how many bugs were open in any prior week of the year is not readily available.
- **Cross-system integration.** The real value of business intelligence comes from combining information from various systems, e.g. the number of bugs in the project vs. efforts invested in QA from the timesheet system. This is not possible without some form of a central data repository.
- **Various types of data sources** – some data is easy enough to extract from databases, however some of it is only available from Excel spreadsheets or text files. Automating the data retrieval process may turn out to be a time consuming task.

In essence, some form of business intelligence is required. Since at Verax Systems we apply the “eat our own dog food” policy, the obvious choice was to use our KPI Dashboard product (<http://www.veraxsystems.com/en/products/kpidashboard>). KPI Dashboard had all the features required to do the job, such as:

- Collection of information from various data sources,
- Wide range of easy to understand diagrams and presentation modes (e.g. pie chart, bar chart, tables and more),
- Automatic calculation of trends and projections,
- Ability to interface with commonly used project management and software development tools such as Bugzilla, JIRA, APINI, Replicon and others.
- Interactive drill-down of visualized data.

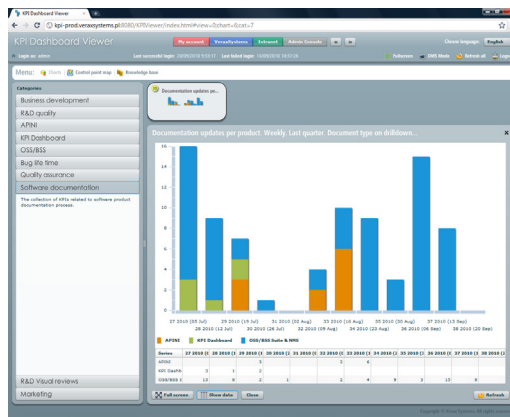


Figure 2: Software documentation progress across all Verax Systems projects – document library and knowledgebase article updates are counted for each product.

5. Communication and policies

5.1. Handling policies

Being measured on individual performance is not a pleasant thing. Therefore, while deploying KPIs it is important to communicate that the new solutions are there to improve things for everybody rather than to act as a tool to criticize employees. The correct **communication of why KPIs have been introduced is essential**:

- Communicate plans beforehand.
- Engage your staff in defining KPIs or identifying areas for improvement.
- Show management commitment and enthusiasm.
- Do not set high KPI targets – in fact it is better to start without them and see what are the reasonable ranges. If targets are impossible to meet, they will only lead to frustration.
- Last but not least: if KPI values are far from expectations – as a manager do not blame anyone and show a positive attitude “now at least we know where we stand – let’s improve from here”.

5.2. Simplicity

At Verax Systems, we strongly believe that simplicity is the key to success. In real life, a software development company will not analyze cubes or perform in-depth analyses using multi-dimensional query language (lack of time is one reason, lack of free resources to spare is another one). A high level overview is required at a glance. This is especially true, considering that the KPIs should be easily understandable for all staff.

5.3. Communication of results

It is important to ensure the resulting KPIs are presented to everyone in the organization. Online web access seems an obvious thing nowadays, but such a passive approach may be insufficient. Taking into account the importance of performance monitoring for the overall business productivity, it is worth considering displaying operational KPIs in employee areas to actively convey metrics at a glance. Similarly, KPIs displayed in executive offices will support decision making processes at the highest level.

6. Internal implementation at Verax – an overview

The deployment of KPI Dashboard internally at Verax Systems was a rapid process – it took only two weeks to implement the key KPIs and about fifteen man-days of effort over a quarter to provide the additional data, including enhancements and elaboration of very specific KPIs for some product teams. Apart from software development and QA, these efforts also covered Sales, Marketing and Business Development KPIs. The following were the key areas of interest:

Quality

- **Number of issues per project** – the ratio of new issues appearing vs. those being resolved. These were taken from Bugzilla used internally. The obviously important parameter was the trend, showing whether the number of problems was growing or being reduced. The applied 3D breakdown into the components gives an idea, where the team has the most problems or what is the technology piece causing most issues (in our case it was Hibernate and JavaScript/DHTML).
- **Number of blocker and critical issues known** – this gives a very good understanding of which resources are required to maintain a product/project. Critical and blocker issues usually require immediate action – knowing how many are in the pipeline allows for much more realistic planning.
- **Average time to resolve an issue and the effort required to fix all the issues** – this is a very good tool for budgeting and forecasting when a desired level of quality can be reached.
- **Number of test cases per project**. It is very important to encourage “writing tests as you go”, or for QA teams to generate test cases when bugs are found. Not only does it indicate the test coverage, but also the trend – coming up with a good test plan is a continuous task and new cases should appear en route. At Verax Systems, Bugzilla Test Runner is used which is where the data was collected from.
- **Documentation**. At Verax Systems, software is continuously documented via SCR (Software Change Request) documents. The SCR is a high-level to low-level document that in theory should be updated as functionality evolves. Typically what happens is that the first SCR version is written and accepted, but rarely updated afterwards and quickly becomes obsolete. The number of SCRs produced is not relevant to the R&D management as they have to be provided in order to kick off implementation anyway. It is the number of changes and updates in the development process that matters as it reflects if a team documents their work.

Innovation

- **Enhancements and improvements**. Building an innovative culture into a company and not forgetting good ideas is always a challenge. That is why at Verax we decided to make the enhancements reporting as simple and as least-time consuming as possible– this was achieved by just by adding a new category in Bugzilla. Enhancements are not rewarded in any way in order to prevent a flood of garbage (pseudo-enhancements), however they are an important aspect of project team assessments.
- **Improvements in specific product areas** (e.g. GUI/front end, performance). This KPI gives a good indication of which product areas require most work. Some general conclusions can also be drawn, such as general lack of resources to optimize UI response times. The risk here is that most internal users will report usability and GUI related issues, not back-end problems.
- **Top enhancers**. Who is the most active person in the project? This KPI is a good indicator in individual performance reviews.

Efforts

- **Man hours in the project.** It is good to see weekly and/or monthly deltas, especially if the effort can be compared against and correlated with other indicators, such as the number of issues in the project. At Verax Systems, we compare individual projects, which enables us to direct resources to the projects that have the most business requirements.
- **Per-activity breakdown.** While software development budgets are usually well controlled, the direction of the spending (e.g. new features, product hardening, support & fixes, etc.) is usually a hazy matter. This KPI immediately shows where the project effort goes to and the condition of each product.

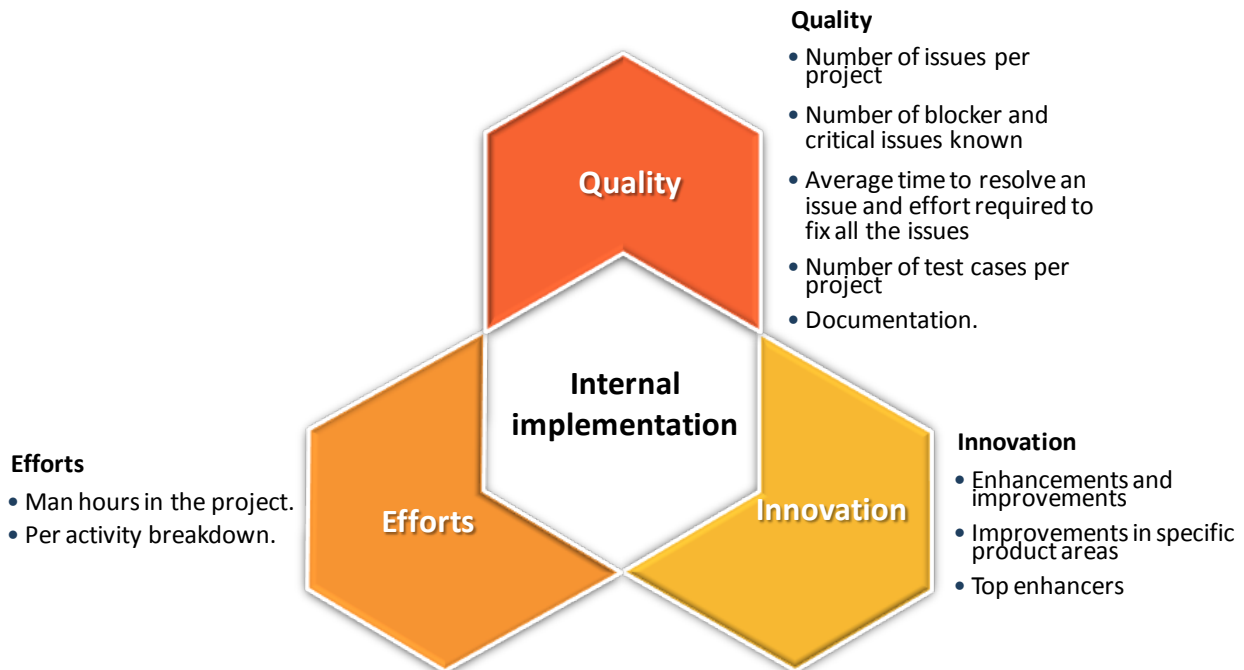


Figure 3: Verax internal software development KPIs – an overview.

7. Results

The KPIs described in this whitepaper worked well for Verax internally. For other organizations, different indicators may be relevant. Despite the relatively small investment, results could be observed in a short time frame (a few weeks):

- Data in various supporting systems (bug tracking, time sheets, test case management) became **accurate and up to date** very quickly (about two weeks time).
- The culture of **continual improvement was established**, as the staff understood that improvements in these areas were essential and it was easy for the management to detect a lack of progress. The most significant improvement was achieved in the areas of enhancement reporting, documentation updates and test case updates.
- Replacement of “a gut-feeling” approach in less visible aspects of software delivery (e.g. project effort breakdown) with **concrete evidence**.

Additional benefits

As a provider of services and solutions mainly to the banking and telecommunication sectors, Verax Systems frequently undergoes various quality audits. The audits are usually performed directly by our clients. The use of KPI Dashboard enabled us to easily demonstrate a good, measured process and build further credibility.

Return on investment

Any new IT systems or processes introduced into a company require an investment. For KPI Dashboard, the expenditure pay back is relatively fast even for small software development departments. The table below presents the calculation of payback period and 1st year ROI (cost oriented) for a team of 15 developers based on the following assumptions:

- the deployment of KPI Dashboard will require 30 man days of consulting,
- the efficiency improvement will amount to 3% (which is a very conservative assumption),
- no additional benefits are taken into consideration (e.g. better management decisions), only developer efficiency is accounted for.

KPI license	2 000
Deployment (30 man days, times consulting rate \$450)	13 500
Total cost	\$15 500,00
Size of engineering team	15
Annual salary (blend)	\$75 000,00
Total annual salaries	\$1 125 000,00
3% efficiency improvement (per year)	\$33 750,00
3% efficiency improvement (per month)	\$2 812,50
Payback period (months)	6
ROI (1st year)	118%

For more information, please visit our website www.veraxsystems.com/en/products/kpidashboard.

8. Sources

- General material from Gartner on KPIs in IT:
http://www.gartner.com/teleconferences/attributes/attr_161183_115.pdf