Implementing Quality Metrics and Goals at the Corporate Level

Pete Rotella
Cisco Systems, Inc.
7200 Kit Creek Road
Research Triangle Park, NC, USA 27709
+1-919-392-3854
protella@cisco.com

Sunita Chulani
Cisco Systems, Inc.
821 Alder Drive
Milpitas, CA, USA 95035
+1-408-424-6802
schulani@cisco.com

ABSTRACT

Over the past eight years, Cisco Systems, Inc., has implemented software quality goals for most groups engaged in software development, including development for both customer and internal use. This corporate implementation has proven to be a long and difficult process for many reasons, including opposition from many groups, uncertainties as to how to proceed with key aspects of the goaling, and the many unanticipated modifications needed to adapt the program to a large and diverse development and test environment. This paper describes what has worked, what has not work so well, and what levels of improvement the Engineering organization has experienced in part as a result of these efforts. Key customer experience metrics have improved 30% to 70% over the past six years, partly as a result of metrics and process standardization, dashboarding, and goaling. As one would expect with such a large endeavor, some of the results shown are not statistically provable, but are nevertheless generally accepted within the corporation as valid. Other important results do have strong statistical substantiation, and we will also describe these. But whether or not the results are statistically provable, Cisco has in fact improved its software quality substantially over the past eight years, and the corporate goaling mechanism is generally recognized as a necessary (but of course not sufficient) part of this improvement effort.

Categories and Subject Descriptors

D.2.8 [**Software Engineering**]: Metrics – *process metrics, product metrics.* D.2.9 [**Software Engineering**]: Management – *software quality assurance, productivity.* K.6.3 [**Management of Computing and Information Systems**]: Software Management – *software development, software maintenance, software process.*

General Terms

Algorithms, Management, Measurement, Reliability, Standardization.

MSR'11, May 21–22, 2011, Waikiki, Honolulu, HI, USA

Keywords

Goaling, BU (business unit), BIC (best-in-class), CSAT (customer satisfaction), MTTR (mean time to repair), CFD (customer-found defect), IFD (internally-found defect), SWDPMH (software defects per million hours (per month)), CFR (customer-found regressions), IFR (internally-found regressions), RNE (release note enclosure), DPAI (development phase average index), NSSTG (Network Software and Systems Technology Group), SWQD (Software Quality Dashboard), SR (service request), IOS (Internetworking Operating System), IOX (IOS variant for service providers), CED (Customer Experience Dashboard), CAP (Customer Assurance Program).

1. INTRODUCTION

Prior to 2003, the approximately 35 business units (BUs) comprising the Cisco Engineering organization each decided which quality and productivity metrics to collect, track, and sometimes goal. The same situation applied to the internal software practices (code review, static analysis, requirements management, etc.) that many of the metrics were associated with. The resulting hodge-podge of metrics and practices made it difficult to track quality changes across the entire Engineering organization, an ever increasingly important consideration since customers typically construct networks that include many types of Cisco platforms and software releases. Since counting rules varied widely even for similar metrics across the BUs, another difficulty was that individual BU leaders sometimes would pick metrics that were in good shape to report up to the corporate level, and not report on troubled areas. Keeping up with all the resultant variability was too big a job for corporate leadership – so tracking at the corporate level tended not to get done in enough detail and with enough accuracy to enable the leadership to make many important decisions with sufficient confidence.

In 2003, one of the largest BUs (NSSTG – Network Software and Systems Technology Group) initiated a project to expand the NSSTG software quality metrics dashboard to include other interested BUs. This organization was uniquely positioned to undertake this effort since it had grown an innovative software engineering group that, in addition to developing and implementing metrics and dashboard capability, was also active in developing and implementing NSSTG-standardized development and test practices, and in managing the release operations function for most IOS (Internetworking Operating System – Cisco's largest software product) releases and releases for many other BUs.

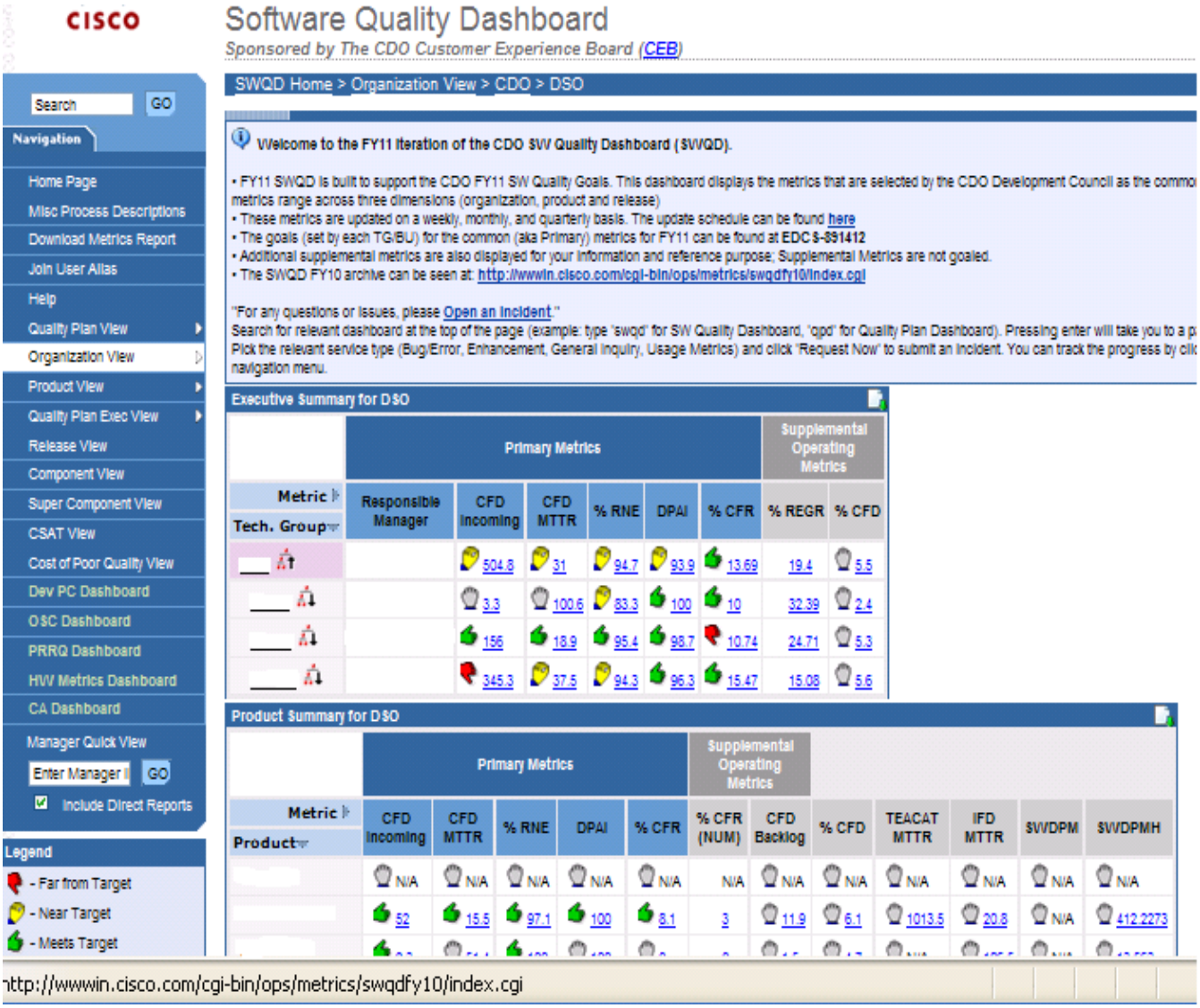

Figure 1: Example page of the Software Quality Dashboard

The NSSTG software metrics dashboard (SWQD – Software Quality Dashboard) was initiated in 2001 by the resident software engineering team, and under their leadership had become an instrumental part of the quality framework for NSSTG by 2003. Several other BUs, that had not already developed their own robust capabilities in metrics and dashboards, decided to adopt all the NSSTG metrics (including their counting rules), and contract with NSSTG to post their data on the SWQD. Most BUs, however, continued to use their own metrics and, in many cases, dashboards, since much time and effort had been invested in tailoring them to best suit their purposes.

When several groups 'joined' the SWQD, it became increasingly apparent to Engineering leadership that accurate, standardized metrics data is not only useful at the corporate level, but essential:

- to compare the progress levels of the various BUs, and to reap the benefits of the Hawthorne effect (viz., the positive influence of merely displaying the data to all BUs, presumably stemming from peer pressure)
- to be sure that metrics do in fact have the same counting rules across the organization, to reduce the temptation to cherry-pick flattering data to report to management, and
- to be able to rely on trends over periods of several years, the duration needed to assess the efficacy of major process changes or the implementation of new processes.

Even though Engineering management saw the benefit of metrics and dashboard standardization, it was reluctant to pressure the non-participating BUs to accept the SWQD as the primary

source of truth for their operations. Besides, most BUs initially resisted inclusion in the SWQD – here are typical reasons given:

- A BU's business is usually in some respects unique, necessitating the development of specifically applicable metrics, or at least some tailoring of SWQD metric counting rules, to best serve the business needs
- Some of the metrics included in the SWQD are not useful or not accurate
- Some individual BUs have extensive dashboards that include sufficient numbers of useful and extensively vetted metrics
- Some individual BUs have deployed more dashboard visualization and querying capabilities than the corporate dashboards, to include sophisticated and user-friendly visualization and reporting capabilities.

As in the Federalist/Anti-federalist debates in the U.S. in the 1780s, the benefits of 'federalism' in metrics, dashboarding, goaling, and, in many cases, the related software engineering practices, has at Cisco been seen to be more beneficial, in the aggregate and in the long run, than the 'anti-federalism' (i.e., siloed) approach in these aspects. We have not conducted a formal survey to confirm this contention, but we do now observe that nearly all of the hitherto harshest critics of the corporate approach are now openly supportive of it.

This paper addresses several aspects of the corporate implementation of standardized metrics and quality goals:

- The goaling approaches, including the best-in-class (BIC) method used to substantiate goal levels, and tying goals to customer satisfaction (CSAT) expectations
- The dashboarding approaches, including data drilldowns, trending, and retrieval of individual bug and service request data
- The 'linkages' analytic approach, including identifying in-process development and test 'levers' (i.e., adjustments to software engineering practices) that can be 'pulled' or 'pushed' to influence downstream quality
- The quantitative results of the Cisco quality improvement program over the past several years, including improvements in customer experience metrics and customer satisfaction survey results.

2. QUALITY IMPROVEMENT PROGRAM

2.1 Software Quality Dashboard (SWQD)

Over the course of the past eight years, the Cisco Software Quality Dashboard (SWQD) has evolved in many ways, and has been expanded substantially. Currently, there are 16 distinct dashboard 'views,' including software views at the organization, product, release, and component levels. In addition, there are several other related views, including: BU quality plans, 'cost of poor quality,' hardware quality, software engineering practices adoption and effectiveness, and customer satisfaction. Figure 1 shows an example page of the SWQD, with actual data (but with the organization and product information anonymized). Much of the metrics data on the SWQD is updated weekly (e.g., the software engineering practices view (SWPD)), or in some cases monthly (e.g., 'cost of poor quality' view), but several of the dashboard views (e.g., the organization view) that are automatically updated weekly can also, when desired, be updated by the users on-demand to include current data.

2.2 Common Metrics

'Common metrics' are SWQD metrics that are goaled for nearly all of Engineering for each fiscal year. In 2011, we have 13 common metrics, but this number is expected to grow to between 16 and 18 in 2012. The current common metrics roster includes:

- Static analysis, code review, and unit test adoption rates, and the average of these three, called the Development Practices Adoption Index (DPAI)
- Escape detection process adoption rate
- Customer-found defect (CFD) incoming level
- CFD MTTR (mean time to repair)
- Percentage of customer-found regressions (%CFR) – the percentage of all customer-found defects that are determined to be the result of collateral damage of fixes
- Percent release-note enclosures (%RNE) adoption
- Number of service requests (SRs) associated with software defects, per million usage hours per month (SWDPMH)
- Number of SRs per million usage hours per month
- Customer satisfaction with the Cisco software (SW CSAT) and with the Cisco hardware (HW CSAT).

Typically, we will publish new metrics on the dashboard for a year or more before we consider implementing goals, but on occasion we will implement goals after six months of vetting, although this risky approach is not often used. This is risky because many small and some not so small issues typically surface during the first year of publication, and often even the metrics developers do not adequately understand the dynamics of metric behavior in the full production environment.

Cisco employs more than 25,000 software development and test engineers whose activities are strongly influenced by the goaling and process standardization requirements instituted at the corporate level. With so many groups, producing a wide range of new products and many new major feature releases for existing products, some teams will likely encounter difficulties in using not fully understood or proven metrics in mission-critical situations. The metrics/process management team has implemented a review process to evaluate and approve situations that require short-term exemptions from the goaling and process standardization expectations, and this has worked well in practice.

2.3 Supplemental Metrics

We publish hundreds of additional metrics, besides the 'common metrics,' but these are not goaled by corporate management. However, many of these supplemental metrics are in fact goaled at the individual BU level, often with assistance from the central metrics/process team with metric modifications, goaling calculations, and dashboarding.

Many of the supplemental metrics are in-process metrics, which often are engineering-level metrics that measure the

effectiveness of individual development and test practices. Most teams are not keen on corporate effectiveness goaling of even the standard development and test practices – they typically believe that balancing the relative strengths of practices is something best done by those close to the actual development and test activities.

In addition to the in-process engineering metrics, we also publish many customer satisfaction metrics, with drilldowns to the product, release, customer, market segment, geography, and (Cisco) internal organization levels.

We also have built tools to effectively mine the corporate service request databases, but have not yet published many of the resultant metrics on the dashboards. In the second half of 2011, we expect to start publishing additional SR metrics, which we hope to then goal in 2012:

- Mean time to restore service (two metrics)
- Forced software upgrade incidents
- Ease of use and configuration (two metrics)
- Customer satisfaction with ease of upgrade.

2.4 'Quality Pyramid' – Internal Metrics

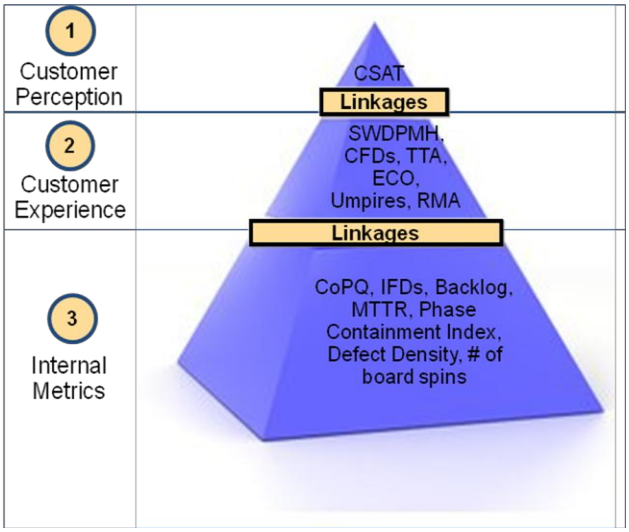

Figure 2: Quality Pyramid – three metric levels

The Cisco quality program consists of three levels (i.e., classes) of metrics and their associated practices and processes (see Figure 2). The first/bottom level consists mostly of in-process engineering metrics, the data used to measure the adoption and effectiveness of individual engineering practices such as static analysis, code review, requirements management, unit test, etc. In addition to metrics pertaining to individual practices, we include in this category other engineering metrics, such as code coverage, defect density, and code complexity, which gauge aspects of development and test that are functions of several, or even all, engineering practices.

The second/middle level of the pyramid includes a variety of customer experience metrics, including customer-found defects (CFDs), customer-found regressions (CFRs), CFD mean time to repair (CFD MTTR), service requests per install base, and others. These metrics quantify the 'pain' that the customer (individually or in the aggregate) experiences in using the software and hardware that our company develops.

The third/top level of the pyramid consists of customer perception metrics, which include software and hardware CSAT results, and other forms of customer feedback ratings and comments.

We are in the process of including two additional levels above customer perception – the customer loyalty level and the revenue bookings level. We have many metrics available at these levels, and are in the process of selecting the ones to dashboard and goal.

The primary purpose of the pyramid approach is to identify 'linkages' between individual metrics (and therefore engineering practices, if starting at the first/bottom level) at lower levels in the pyramid and the metrics at the next higher level, and so on. The overall goal is to identify metrics and practices at the first/bottom level that will, when 'improved,' likewise 'improve' metrics at the next higher level, and so on to the top level. See section 4.1.1 for additional discussion of this 'linkages' concept.

2.5 Quality Pyramid – Customer Experience

At the second/middle level of the quality pyramid is the collection of metrics that gauge the customer's experience with our products. Several of the common metrics fall into this category:

- Customer-found defect (CFD) incoming level – the number of high- and medium-severity software bugs reported by customers
- Software defects per million hours usage per month (SWDPMH) – the number of service requests associated with bugs of all severities that are filed per month, divided by the number of usage hours for the month
- CFD MTTR – the number of high- and medium-severity bugs in backlog, divided by the average weekly fix rate
- %CFR – the number of high- and medium-severity CFDs that are judged to be regressions (i.e., the result of collateral damage, that is, bad bug fixes), divided by all CFDs, times 100
- High- and medium-severity service requests (SRs) per million usage hours per month.

Other metrics are considered to be part of this middle level of customer experience, such as:

- SR time in high- or medium-severity (percent of cases resolved in under n hours)
- Time to initial solution for SRs (percent of initial solutions achieved in under n hours)
- Percent of SRs categorized as usability, configuration, or documentation problems
- Forced upgrade incidents reported by customers.

The common theme among these metrics is that they attempt to measure customer pain – the difficulties the customers have with the full range of products and services we provide, such as our software, hardware, documentation, and technical support.

2.6 Quality Pyramid – Customer Perception

The third/top level in the pyramid is the customer perception level – how the customer feels about the experience with our products and services. At any given time, a customer using a

wide range of our products and services may have several troublesome CFDs and SRs outstanding, but may feel positive about the full experience. We have found that there exist strong correlations between metrics at the experience and perception levels in the aggregate, but drilling down to individual experience attributes is important to identify and rectify problem areas that may be chronic, despite overall good perception, or areas that may grow in difficulties if not addressed.

Listed below are some of the perception-level metrics that are currently gathered and published on the CSAT dashboard, called the Product Satisfaction Analyzer (PSA), and the corporate dashboard that concentrates on customer-centric satisfaction metrics and data, the Customer Experience Dashboard (CED):

- Customer satisfaction with software (displaying attributes and aggregates, with drilldowns to product, release, and internal organizations) – if the corporate software CSAT yearly aggregate goal is achieved, the yearly bonuses for all employees are increased
- Customer satisfaction with hardware (displaying attributes and aggregates, drilldowns to product and organization – similarly, the corporate hardware CSAT level influences bonuses for all employees
- Customer perception attributes are included on the Customer Experience Dashboard (CED), including loyalty factors and levels, technical support satisfaction, account relationship satisfaction, and others.

Our current goaling strategies, 'BIC' and 'CSAT Targeted' (see sections 2.7 and 2.8), are devised and implemented to improve the customer perception, as gauged by the CSAT Survey metrics. We put heavy reliance on the 130,000 or so customer responses we receive yearly. For the BIC approach, we also rely on the 'linkages' analyses (see section 4.1.1) to correlate best-in-class metrics levels with CSAT levels. For the CSAT Targeted approach, for now we rely solely on the CSAT survey results.

CSAT is the dimension we are ultimately focused on at this time, even more so than customer loyalty or bookings, but once we establish reliable linkages between CSAT and loyalty, and loyalty and bookings, we may decide to shift some focus, depending on the strength of the correlations, the granularity and accuracy of the correlation results, and other practical considerations [see ref. 2].

2.7 Goaling: Best-in-class Method

Several goaling approaches have been implemented in recent years. In August, 2009, Engineering switched away from a goaling mechanism that basically used historically achievable results as benchmarks. For example, if a certain metric achieved a 10% improvement last year partially, it is supposed, as a result of setting, say, a 20% improvement goal, perhaps if we set a 20% goal this year, we will again achieve similar results. The central metrics/process team and many engineering groups were not happy with this approach – there was no rational basis for the goaled percentages or absolute target levels, except history.

Engineering switched from a history-based goaling mechanism to one linked to benchmarks that identify the top 10% (i.e., 'best in class') companies for each goaled metric in the appropriate market segment. We have investigated the literature [1] and have received input from industry experts to ascertain values indicative of the top 10% of the applicable industry for each 'common' metric. (For example, for the IOS, IOX, and similar bridge/router software, we consider the top 10% of the 'telecommunications and operating systems' sector to be the appropriate comparison sector.) Once we ascertain the approximate value corresponding to 'best in class,' it is a straightforward calculation to determine the yearly goal that is needed to enable a group to achieve a 'best in class' value in a specified/desired number of years. A 'what-if'-type BIC calculation tool (Common Metrics Goals Calculator) is used to develop goal proposals that are used in yearly negotiations with the BUs, and to enable BUs to generate what-if scenarios and assess tradeoffs for goaling. In an iterative way, this approach enables us to present agreed-upon goaling proposals to the Cisco Development Council, the council that manages Engineering.

In other words, we ask ourselves how soon we want to reach best in class values and then estimate whether or not the required improvement rates are do-able, given the history of the group. History does continue to play a role, but in a different way. If the required improvement rate is estimated to be too aggressive, the corporate expectation is usually to not extend the number of years to BIC, but to ramp up in quality improvement initiatives.

This seems to be, on the surface, an inappropriate way to proceed – why not simply match historical rates with future expectations, then adjust the number of years to BIC accordingly? Fortunately, we have found that groups that 'produce' poor quality metrics are often open to accelerating their improvement efforts to include more proven quality initiatives and greater attention to existing initiatives and practices. But still, the adjusting of goal levels, years to BIC, and the types of initiatives and practices employed are ultimately a matter of negotiation with BU General Managers, the central metrics/process team, and the Development Council.

2.8 Goaling: CSAT Targeting Method

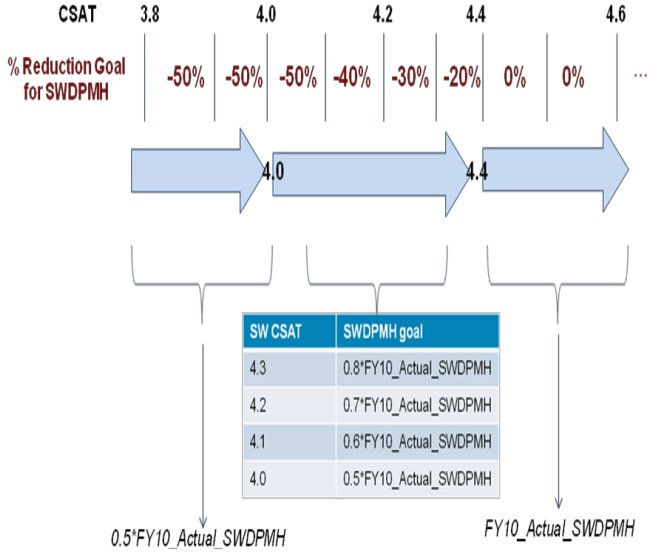

Figure 3: 'CSAT Targeting' goaling method

In addition to linking metric goals to BIC levels, we also link some goals directly to customer satisfaction (CSAT) survey results (see schematic in Figure 3, an example of this method for SWDPMH goaling, where CSAT is measured on a Likert scale of 1 to 5, with 5 the highest satisfaction score). This approach was developed and implemented to address the situations where we do

not have sufficient certainty in the BIC benchmark levels. Instead of calculating years to BIC, we calculate years to achieve a CSAT value that corresponds to deep market penetration. Efforts are underway to link BIC levels to CSAT itself, but this capability is not yet established. As with the benchmarked BIC approach, we calculate the number of years to the desired CSAT level. When we complete collection of CSAT BIC data for pertinent market segments and products, we will establish years to BIC for CSAT in the same way we currently do for the common metrics.

2.9 Standardized Development Processes

Many of the in-process metrics, goaled or not goaled, are directly related to specific development or test practices that have been standardized across the company. By 'standardized,' we mean that there are mandatory requirements for static analysis, code review, requirements management, unit test, and other practices, but the individual groups can choose which tools to use and when in the lifecycle to do the practice. Specific tools are recommended for each practice, but conformity is not required. (It is easier in practice, however, to meet the corporate process, metrics, and documentation requirements if the recommended tools are used, therefore standard usage is the norm.)

Current goaling concentrates mostly on 'adoption' metrics associated with the standard practices. We want to make sure that engineering groups do the basic practices, but there has been a lot of resistance to setting corporate 'effectiveness' goals for the individual practices. A number of practice effectiveness metrics are published on the SWQD, but none is goaled across the company. Most engineering groups want to be able to decide for themselves how to allocate engineering time and other resources, and therefore standard goaling has not yet been possible across the entire organization. Many teams have set their own practice effectiveness goals, however, and we publish the metrics data, goals, trend charts, bug lists, and other ancillary data in the 'quality plan view' on the SWQD.

2.10 Service Request Metrics

iSRV is a tool used to query the service request (SR) database (called C3). iSRV also provides various visualizations, SR lists, drilldowns, and other important capabilities. In early 2011, only 'SRs per install base' is published and goaled for all Engineering, but plans are to publish and goal 3 to 5 additional SR-centric metrics later in 2011 and 2012 (see sections 2.3 and 2.5).

Analysis of, and concentration on, improving SR attribute metrics unfortunately is far behind the work done on understanding and improving CFD rates, CFR rates, and other bug-related aspects of quality. We know from linkages studies (see section 4.1.1) that approximately half of the correlation between customer experience and customer perception is explained by bug-related metrics. There is ample evidence indicating that much of the remaining 'linkage' exists in correlations between SR attributes (such as usability, maintainability, documentation, performance, and others) and software CSAT.

3. QUALITY IMPROVEMENT RESULTS

Over the past eight years, the company has grown substantially, with non-services revenue increasing by 110% and engineering headcount increasing by 65%. Over 200 new software products were introduced in 2010 alone, growing from ~100 yearly as recently as five years ago. Concurrent with this high level of growth (including the acquisition of 72 companies over the past eight years), a number of quality improvement strides have been made. This section describes what worked and did not work in achieving corporate quality goals during this high growth period.

3.1 Customer-found Defect Reductions

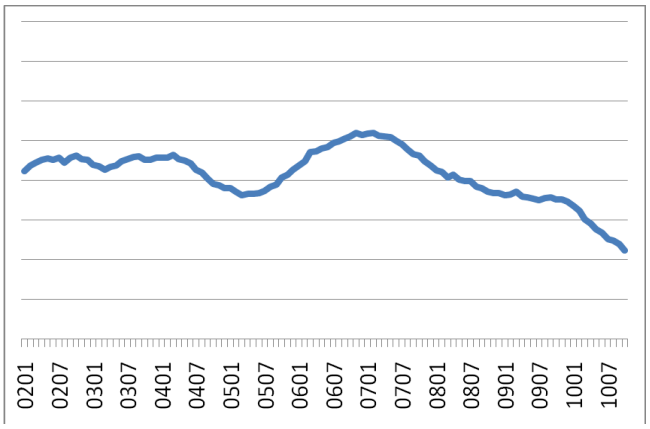

Figure 4: CFD trend over the past eight years

Up to 2002, CFDs increased substantially across the company, but then leveled off and dropped during 2004 and early-2005, only to increase rapidly starting in mid-2005 (see Figure 4 – many of the values of key metrics in this paper, including most y axis values, are not shown for reasons of company confidentiality; all y axes are linear; not all y axis scale values start at zero). Customer complaints about our software increased dramatically, and several wide-ranging quality improvement initiatives were spawned and ramped up in Engineering. One major set of initiatives, termed 'phase containment initiatives,' was a primary factor in the dramatic improvement in CFD rate over the next two years, 2007-2008, and beyond. (The phase containment initiatives consist of the mandatory and aggressively goaled standardized practices for static analysis, code review, unit test, and escape detection – see section 2.9.)

A second, and equally strong, improvement inflection occurred in mid-2009, largely as a result of the implementation of the best-in-class goals (described in section 2.7) that was rolled out in late-2008 (since the average CFD lag is seven to nine months after the start of the rampup of a typical new quality improvement initiative). This overall progress has continued up to the present time, with CFD volume 23% lower than that seen 8 years ago. CFD volume is 28% lower than the level seen at the start of the major quality initiatives in the 2005-2006 timeframe, and at the same level we experienced 9½ years ago, when the company was building roughly half the software we are building now. Therefore, the software we are currently building is introducing approximately one half the number of defects into the customer space, per KLOC (thousand lines of source code), as we did in the 2000-2002 timeframe.

Another way to look at the impact of the quality improvement initiatives over the past 8 years is to examine CFD data normalized by product revenue. Figure 5, below, shows the improvement in this metric over eight years (where the lag between revenue and CFD incoming is known to be ~12 months):

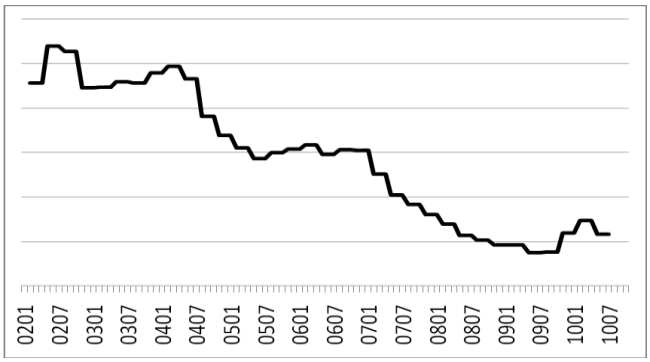

Figure 5: Customer-found defects normalized by revenue

We still have a ways to go to reach the BIC level with CFD incoming, however. We need to reduce the CFD incoming levels by an additional 21% below the 2002–2003 baseline, beyond the 24% already achieved. At the current improvement rate, we expect to reach BIC sometime in early 2013.

3.2 SWDPMH Reductions

Software DPMH (i.e., service requests associated with defects, per million usage hours per month) is a welcome addition to our spectrum of customer experience metrics, since it gives us a normalized view of customer pain, whereas CFD incoming levels, despite utility in gauging the total defect population encountered by customers, do not give us a feel for how often the CFDs are actually encountered. Most CFDs are encountered by only a handful of customers, but some are encountered by hundreds.

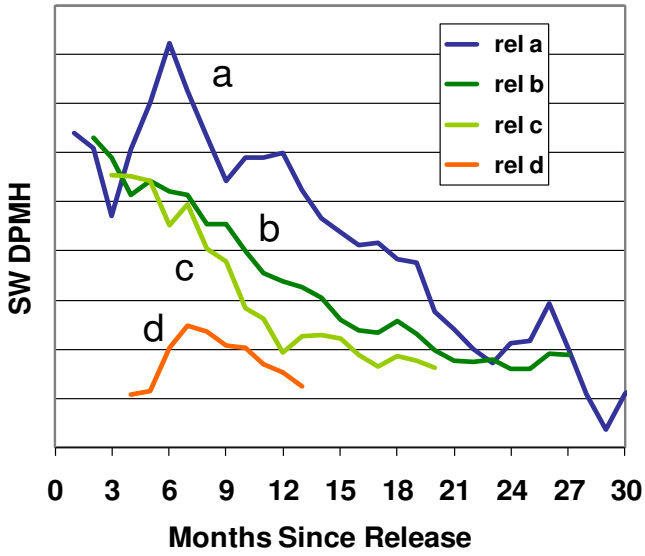

Figure 6: SWDPMH for a release sequence for one product

We have not yet implemented an aggregate view of SWDPMH for current values and historical trends, so we cannot show a reliable corporate improvement experience over the three years that SWDPMH has been published for most of Engineering. But we can show an example (see Figure 6) of a typical large product family for a sequence of major feature releases, showing substantial (and typical) improvement over time. Seventy percent of the product families are goaled for SWDPMH in early-2011, with >90% expected to be goaled in late-2011. Data for >75% of the products has been shown on the SWQD for 3 years.

In addition, this same 70% of the product families are also goaled for high- and medium-severity service requests (SRs) per million usage hours (install base) per month (i.e., SR/IB), with similar (to SWDPMH) expectations for goaling >90% in 2012. However, this is the first year of publication of SR/IB on the SWQD, so there is some doubt that it will be fully vetted by the start of fiscal year 2012 (beginning in August, 2011).

3.3 MTTR Reductions

We calculate and publish several variants of an MTTR metric, but only goal CFD MTTR at the corporate level. (MTTR here is the number of high- and medium-severity bugs in backlog, divided by the average weekly fix rate.) Over the past 8 years, we have seen a 47% improvement in CFD MTTR, using 28 days as the BIC level for the 'telecommunications and operating systems' software types. Figure 7 shows the long-term trend:

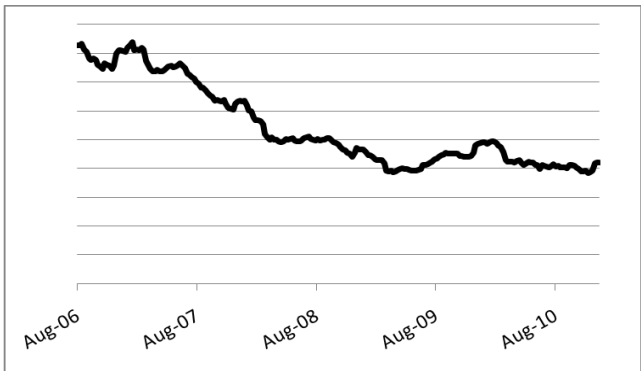

Figure 7: CFD MTTR corporate long-term trend

Progress in this metric has been slower than expected over the past two years, but Engineering has introduced several new MTTR initiatives, and we expect to reach BIC levels in 2013.

3.4 DPAI Increases

We started out in August, 2005, at about 25% adoption for the corporate-standard 'phase containment' practices, then ended up at 77% adoption of the corporate-standard static analysis suite of tools and processes, 80% adoption of standardized code review, and 75% adoption of standardized unit test. DPAI (development practices adoption index – the average of static analysis, code review, and unit test adoption rates) reached 77% in 2006, thereby exceeding the goal for FY2006, starting in August, 2006, of 75% for DPAI and each of the three individual practices. We also exceeded the escape detection process (EDP) adoption goal of 75%, with an 87% corporate level. In August, 2007, we reached 92% adoption for static analysis, 93% for code review, 92% for unit test, 92% for DPAI, and 93% for EDP, so we missed all of our 95% adoption goals. Effectiveness metrics (such as high- and medium- code review issues per KLOC) are published on the SWQD, but have not yet been goaled at the corporate level, although a number of BUs goal these.

3.5 %CFR Reductions

The NSSTG organization has goaled %CFR (percentage of all customer-found defects that are determined to be the result of collateral damage, that is bad bug fixes) for several years, and has

managed to improve ~9% per year using a variant of the BIC method for goaling. Several key processes designed to reduce the collateral damage of fixes have been implemented in NSSTG, and several other BUs are now piloting similar approaches.

All of Engineering has been goaled for %CFR since August, 2009, and the metric has been improving at the yearly rate of 11% since then (over the past 18 months).

3.6 %CFD Improvements

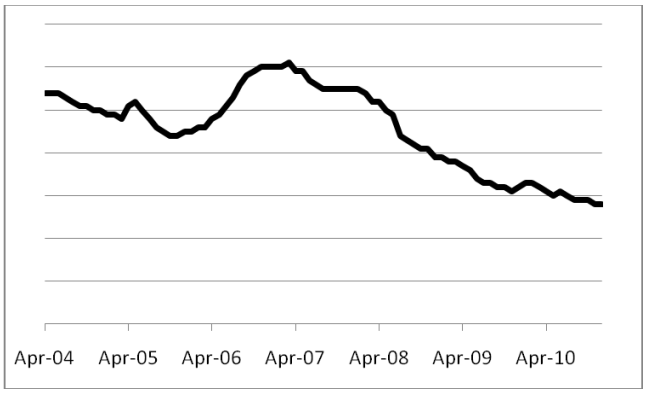

Figure 8: %CFD corporate long-term trend

Percent CFDs (our version of a defect removal efficiency metric) is the number of CFDs divided by the total defects (including (fully counted) issues uncovered in the development practices, such as code review) times 100. Our goal is 3.5 %CFD, the BIC level for telecommunications and operating systems software. Figure 8 shows the %CFD trend over the past eight years.

3.7 CSAT Improvements

The aggregate values of software CSAT and hardware CSAT are goaled metrics for all of Engineering, and, in addition, are substantial factors in the equation used to calculate yearly bonuses for all Cisco employees. Software and hardware CSAT aggregate values are at present the key metrics in the third/top level of the quality pyramid described in section 2.4, and as a result are a major focus of 'linkages' correlation work (see section 4.1.1), the attribute analyses to identify important problem areas (as well as to identify potential best practices), and many other analyses. The major contribution of the BIC and CSAT-targeted goaling methods, and the initiatives motivated by and to some extent driven by these approaches, has been the contribution to improvement in the CSAT aggregates and attributes over the past five years. We have, however, seen a flattening of all CSAT aggregates and most attributes in 2009-2010, mostly as a result of macroeconomic factors associated with the deep recession in the technology and other sectors – this macroeconomic influence is substantiated by the experience of our survey vendor (who is also supporting other companies in our sector).

Figure 9 shows software and hardware CSAT values over the past six years. The flattening of CSAT results across the board in FY09 (starting August, 2008) and FY10 (starting August, 2009) is being examined in detail to identify additional (besides macroeconomic) possible causes and possible remediation steps.

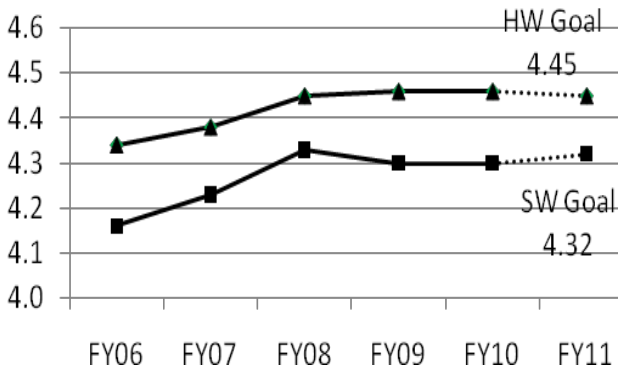

Figure 9: Software and hardware CSAT long-term trends

3.8 Customer Escalation Reductions

The Customer Assurance Program (CAP) deals with customer escalations (i.e., high customer dissatisfaction with software, or hardware, or technical support, etc.). The number of CAP cases has declined ~7% per year over the past eight years, and we have evidence that much of the decline (50–60% of the drop) is attributable to software issues. See Figure 10:

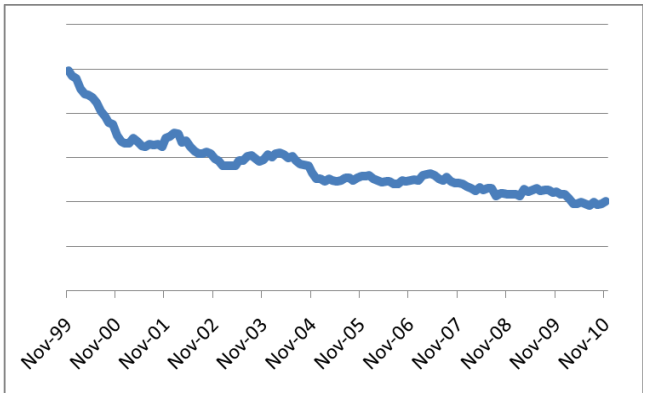

Figure 10: Customer escalations (CAP cases) long-term trend

4. ANALYSIS AND MODELING

Several types of analysis and modeling are used to guide and substantiate the creation of viable goaling approaches, and also the adoption or modification of development and test practices.

4.1 Linkages Modeling

One important type of modeling we do aims to find correlations between in-process metrics in the first/bottom level of the quality pyramid described in section 2.4 and specific customer experience metrics in the second/middle level. We call this approach 'linkages' modeling, since we are trying to find 'levers' in development and test that can positively influence customer experience. Univariate linear regression models are initially examined, then multivariate models are built. Figure 11, below, shows an example of an initial univariate examination [3] of the top five Pearson correlations between 35 in-process metrics and one major product's SWDPMH, an important (and goaled) customer experience metric:

Logical Lever	Correlation with SWDPMH	t-value	P-value
Development Defect Density	0.91	6.50	< 0.01
Unit Test Escapes (for CFDs)	0.74	3.35	0.03
IFD MTTR	0.57	-2.29	0.08
% IFR	0.55	2.22	0.09
Code Review Escapes (for CFDs)	0.52	2.08	0.10

Figure 11: In-process metrics correlations with SWDPMH

Next (Figure 12) is an example of 'heat map' visualization that displays several correlations found in multivariate linear regression models (with P values < 0.05) for three major products:

Lever Metric	Product A	Product B	Product C
Unit Test Escapes	VS	VS	M
Development Defect Density	M	VS	M
Release-gating Bugs (OIB,S1,SS)	VS	M	W
MTTR	VS	W	S
Code Review Escapes	W	S	VS
Open Bugs at FCS	S	W	S
% IFR	W	S	W

vs = Very strong lever (r^2 = 70-93%)
s = Strong lever (50-70%)
m = Moderate lever (30-50%)
w = Weak lever (0-30%)

Figure 12: Heat map of in-process v. SWDPMH correlations

Next (Figure 13) is a visualization summarizing a collection of multivariate regression models showing linkages between in-process metrics and customer experience (SWDPMH) and linkages between SWDPMH and customer perception (software CSAT):

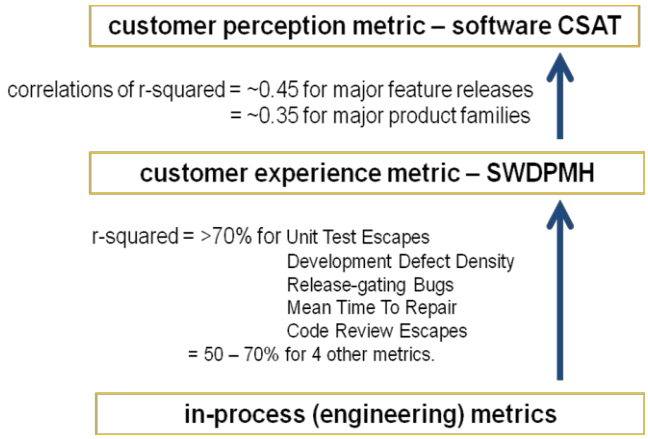

Figure 13: Linkages results cascading to the top pyramid level

If we find linkages, we recommend a number of process changes that have been found to be effective in other groups and BUs. Figure 14 is a sample set of recommendations:

Figure 14: Possible practice changes tied to modeling results

In-proces Metrics	Suggested Actions (incomplete list)
Unit Test	• More focus on unit test during development • RCA of unit test escapes to discover weak areas • Training / adoption of best practices around unit test • Unit (white box) testing removes internal-structure/all-paths bugs
Code Review	• More focus on code review during development • RCA of code review escapes to discover weak areas • Training / adoption of best practices around code review • Formal inspection identifies (removes) many common bug types
Development Defect Density (DDD)	• Begin tracking DDD in your org if not already • Set DDD thresholds and do PPAs on projects that exceed that value • Employ dev phase containment practices (resolve all static analysis warnings, run unit tests / code reviews, fix all Sev1/2 and SS/TS bugs before handoff to devtest) • Under consistent testing conditions, DDD measures bug injection rates
OIB's and other release-gating bugs (i.e., S1, SS)	• Engage Advanced Services to identify OIBs early in the process • Address all release gating bugs prior to FCS • Identification and removal of the most toxic bugs
Open S1-3@FCS	• Tighten commit and release/throttle pull criteria • Fix defects earlier in the process • Identification and removal (or deferral) of serious and critical bugs
MTTR	• Achieve / maintain corporate standard 28-day MTTR • Steady-state bug removal indicator; efficacy of sustaining effort
Regressions	• Set aggressive regression reduction goals • RCA regression defects to discover sources of collateral damage • Implement architecture / process changes to reduce collateral damage in future releases • Injection reduction while bug fixes are made, inspected, and tested.

4.2 SR Analysis

Figure 15 displays a sample view from the Interactive Service Request View tool, iSRV. In addition to querying capabilities against the C3 and SRH service request databases, this tool also provides several data views that we are considering for use for goaling purposes in fiscal year 2012, starting in August, 2011:

Figure 15: Service request data mining and visualization tool

5. QUALITY PROGRAM OBJECTIVES

In order to substantiate a potential goaling method or promote a new development or test practice, it is essential to first build a model that estimates sufficient impact to the business. Then, following a successful pilot of the model, it is necessary to 'sell' a full-scale production implementation. We say 'impact' and 'sell,' because, in addition to discovering numeric correlations and hopefully causality, the implementation of a model's findings in a large corporate setting must also be expected to be impactful. Companies typically do not implement small-impact changes across all of Engineering or even across a large BU, since there is only a limited number of process implementations or changes that a large organization is willing to support in a given year.

Therefore, in the analysis and modeling efforts, we are more interested in models that have potentially high impact, rather than those that simply have high precision, accuracy, and recall, but have less of a predicted overall effect – it is more important to take a bigger risk and try to implement a bigger change in reliability, availability, productivity, ease of use, or whatever type of product improvement we are after, than to implement a collection of less risky, but smaller improvements.

In order to achieve any impact at all, however, it is essential that the data mining and reporting be highly accurate at all times, and particularly in the early stages of dashboarding, reporting, and goaling. Early on, many groups (particularly the poor-performing groups, and ironically, the highest-performing groups) will dispute the data and raise numerous (often minor) issues that have the cumulative effect of weakening the program. From time to time we hear this type of comment, even now: 'If this data point is wrong, how can we trust any of the data?' Therefore, we spend at least a week testing and validating any changes made to the dashboards, or their underlying queries or algorithms, prior to inviting the user community to examine data on the test server for at least a week for testing and validation prior to the production rollout. After changes go into production, responsive technical support is also essential to keep user confidence high. It is easy for a corporate-level quality program to lose credibility if the data accuracy drops too low or if bugs are not corrected quickly.

6. NEXT STEPS

In 2011, we have transitioned from only organization-level goals to both organization and product goals. Customers are usually focused on how specific Cisco products and solutions are doing, so we are now concentrating on both the product and organization dimensions, and attempting to understand the dynamics of the product dimension more fully.

We know that less than half of customer dissatisfaction with our software is a function of software bugs – the remainder is a function of ease of use, time to adoption, interoperability, technical support, and several other 'ilities.' We are concentrating on these non-reliability areas more heavily in 2011, attempting to identify which areas contribute the most to CSAT and which 'levers' in development and test are most influential in improving customer experience attributes relating to these areas.

A key objective is to reach best-in-class levels for all our common metrics in 2012, if possible, or at least in early 2013. This objective is putting a lot of pressure on Engineering to not only concentrate on a common set of goals, but to also change business practices that in many cases are long-established. This objective also necessitates that we continue to improve our data mining and data reporting capabilities. Without thorough, accurate, and timely metrics data, our corporate goaling/process improvement program would be largely ineffective.

7. ACKNOWLEDGMENTS

The work described in this paper was done, of course, by many hard-working and dedicated Cisco employees in addition to the authors, including Wenje Lai, J.P. Chen, Bob Mullen, John Intintolo, Rick Follenweider, Bob Karski, Jim Lambert, Luis Morales, Ken Patton, Katie Foster, Steve Arnold, Saritha Kola, Aditi Chopra, Jun Chen, Steve Ricossa, Rozita Mirzamohammadi, Mark Manzanares, Mark Telford, Venil Tirouvingadame, and many others.

8. REFERENCES

[1] We relied heavily on the work of Capers Jones, John Musa, and others, and including teams at Carnegie Mellon U., IBM, Siemens, NASA, Nortel, and other organizations.

[2] Chulani, S., Santharam, P., Hodges, B., Anders, K. 2007. Metrics-based Management of Software Product Portfolios. *IEEE Software* (Mar. 2007), pp. 66-72.

[3] Rotella, P. and Chulani, S. 2010. Linkages Between Metrics Types: In-process, Customer Experience, Customer Perception. *Proc. of ISSRE* (Nov. 2010), pp. 42-81.